In-depth Analysis of Financial Product Price Fluctuations Based on SVJJ Model Improved Reinforcement Learning

Rui Zhang[a], Zixu Li[b], Qier Mu[b], Chenxue Yang[c,*]

[a]*School of Management, Beijing Institute of Technology, , Beijing, 100190, China1120220789@bit.edu.cn*
[b]*School of Computer Science & Technology, Beijing Institute of Technology, , Beijing, 100190, China1120211166@bit.edu.cn, 1120220659@bit.edu.cn*
[c]*Agricultural Information Institute, CAAS, , Beijing, 100190, Chinayangchenxue@caas.cn*

Abstract

Financial product prices are often affected by a variety of complex factors, such as economic data, political events, market sentiment, capital flows and market structure. The uncertainty and variability of these factors lead to a high degree of uncertainty in the price fluctuations of financial products. The nature and high volatility make it particularly difficult to accurately predict and analyze financial product prices.

Therefore, a new in-depth analysis method of financial product price fluctuations based on **SVJJ model improved reinforcement learning** is proposed. The **Hurst index centralized fractional Brownian motion** is used to measure the volatility of financial product prices, and the **rough Heston model with jumps** is introduced to construct the SVJJ model that simulates the price fluctuations of financial products.

The **SVJJ model** is used as the agent of the **inverse reinforcement learning** algorithm to achieve improvements in reinforcement learning. The reward function of the inverse reinforcement learning algorithm is updated through the **gradient ascent method**, and the algorithm is trained using the global gradient calculation process to achieve in-depth analysis of price fluctuations of financial products.

The **CSI 300 ETF option product** was used as the analysis object for experimental testing. The experimental results show that this method can deeply analyze the price fluctuations of financial products. Based on the historical volatility and expected volatility analysis results, it is expected to bring **higher returns** and **lower risk** and benefits to investors and financial institutions.

Keywords:
SVJJ model; Improved reinforcement learning; Financial products; Price fluctuations; In-depth analysis; Reward function

1. Introduction

In today's highly integrated globalization and information technology, the financial market is increasingly becoming a complex and volatile system, in which the price fluctuations of financial products, as a direct reflection of market information, have depth and breadth that not only concern the direct interests of investors, but also profoundly affect the stability and development of the entire economic system.

The **depth of price fluctuations**, in short, refers to the amplitude and frequency of up and down fluctuations in the price of financial products in a specific time period, as well as the information content of these fluctuations and the trading activity of market participants. It not only reflects the dynamic changes in market supply and demand, but also implies the influence of the macroeconomic environment, policy guidance, market sentiment, investor expectations and other factors.

In the financial market, accurately predicting the price fluctuations of financial assets has always been the focus of investors and policy makers [1]. Price fluctuations not only reflect the changes in market supply and demand, but

*Corresponding author

Figure 1: Volatility comparison across major financial assets (2020-2025), showing the varied patterns of price fluctuations among stocks, bonds, commodities, and cryptocurrencies.

also make clear the impact of macroeconomic, policy environment, market sentiment and other factors on the price of financial assets [2].

Therefore, in-depth study of the depth of price fluctuations of financial products not only helps to reveal the internal logic of the market operation, but also provides investors with more accurate risk assessment and return prediction tools, and promotes the effective allocation of resources. At the same time, for regulatory agencies, mastering the law of the depth of price fluctuations can help to detect abnormal market fluctuations in a timely manner and take necessary intervention measures to maintain the fairness, justice and stability of the market.

Currently, **deep learning** and **reinforcement learning models** are increasingly used in the financial field [3]. In recent years, many researchers have conducted research on financial analysis. **Soleymani et al.** studied a deep graph convolution reinforcement learning method for financial portfolio management [4].

This method uses **graph convolutional networks** to capture the dynamic inter-relationships between financial instruments. These relationships are represented in the form of graphs, where nodes correspond to financial instruments, and edges correspond to pairwise correlation functions between assets, allowing the model to more accurately understand complex relationships in the market and optimize investment portfolios accordingly.

Through the **actor-critic reinforcement learning structure**, **DeepPocket** can automatically learn and execute investment policies, while using critics to evaluate investment policies to determine the best course of action. This automated learning process enables the model to continuously adapt to market changes and optimize portfolio returns. DeepPocket can be evaluated and optimized in different investment periods and different market environments, showing strong adaptability and generalization capabilities.

Deep neural networks can handle a large number of features in complex environments and have excellent per-

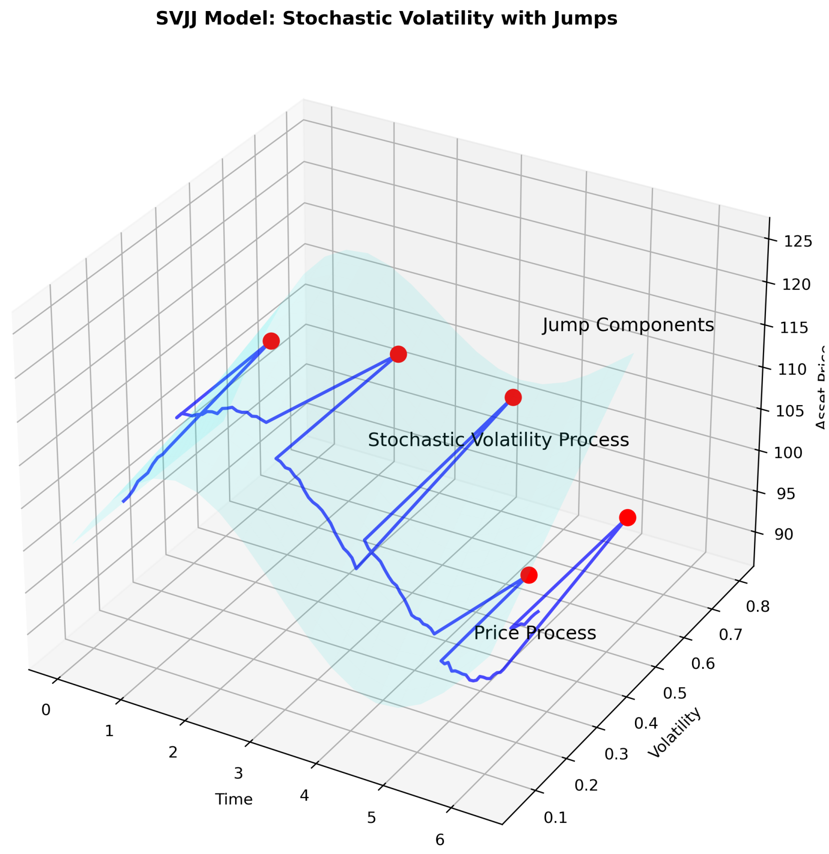

Figure 2: A conceptual illustration of the SVJJ model structure, demonstrating how stochastic volatility and jump components interact within the model framework to capture complex market dynamics.

formance on high-dimensional data in financial markets, allowing DeepPocket to cope with complex financial market environments.

However, the **rewards of reinforcement learning** are often unstable, leading to instability or convergence difficulties in the training process. In financial portfolio management, this instability may increase investment risk. Because the decision-making process of deep reinforcement learning models relies on black-box models, it is sometimes difficult to understand how the model makes decisions. This lack of explanation in finance can reduce investor trust in models.

Lee et al. proposed a financial time series transaction analysis method based on **wavelet transform** and **deep reinforcement learning** [5]. Wavelet transform has good "adaptive" and "zooming" characteristics and can decompose signals into different frequency channels and then smooth them, thereby effectively removing noise in financial time series data and obtaining approximately stationary signals with extremely low distortion, which is crucial for subsequent analysis and prediction.

Time series data in financial markets are often non-stationary, and traditional frequency domain analysis methods are often difficult to handle. As a time-frequency analysis method, wavelet transform can effectively handle non-stationary signals and provide more accurate analysis results. Deep reinforcement learning can automatically learn and execute investment strategies, and through continuous trial and error and optimization, find the optimal action plan in a specific market environment.

This automated learning process enables the model to continuously adapt to market changes and improve trading performance. **High-frequency financial time series data** contains rich market information and trading opportunities. Through the combination of wavelet transform and deep reinforcement learning, these data can be used more effectively to capture small fluctuations and trend changes in the market, thereby formulating more accurate trading strategies.

Although deep reinforcement learning can automatically optimize investment strategies, the stability of its strategies may be affected by changes in the market environment. Under extreme market conditions, the model may not be able to adapt to market changes in a timely manner, resulting in increased transaction risks. When using this method, a complete **risk control mechanism** needs to be established to ensure transaction security.

Salisu et al. studied the predictability method of global financial cycles and oil market fluctuations [6], using a **generalized autoregressive conditional heteroskedasticity mixed frequency data sampling model** to process mixed frequency data, that is, considering both high-frequency and low-frequency variables. Since market data often contains information of different frequencies (daily, weekly, monthly, annual), this model can effectively integrate this information to improve the accuracy and comprehensiveness of forecasts.

By decomposing volatility into **long-term and short-term components**, we can capture the long-term trends and short-term fluctuations of the global financial cycle and the oil market respectively. This separation helps to better understand market behavior and provides investors and policymakers with deeper insights. The parameter estimation results of this method are affected by many factors, such as sample size, data frequency, model setting, etc. This requires researchers to conduct adequate sensitivity analysis and robustness testing when applying the model.

Kirisci et al. studied a new **financial time series model** construction method based on **CNN** [7]. This method can automatically extract key features from financial time series data without manual complicated feature engineering, which greatly reduces data preprocessing time and complexity, and improves the generalization ability of the model.

The **spatiotemporal relationships** in time series data, especially local features and temporal dependencies, can be captured through convolution operations. This is useful for short-term fluctuations and long-term trend predictions in financial markets. A **parameter sharing mechanism** is adopted, that is, the same convolution kernel remains unchanged when processing the entire input data, which reduces the number of parameters of the model, reduces the risk of overfitting, and improves computational efficiency.

CNN usually requires input sequence data of fixed length. However, the length of financial time series data is often not fixed. This requires preprocessing operations such as **filling or truncation** of the data before using the CNN, which may introduce some errors or lose some information. Financial time series data often contain a large amount of noise and outliers. Although this method has certain anti-noise ability, it may still be interfered by noise and affect the prediction performance in extreme cases. Although CNN performs well in feature extraction and spatiotemporal relationship modeling, a single model may not fully capture all dynamics of financial markets, resulting in poor adaptability of this method.

The **stochastic volatility jump model (SVJJ model)** is a complex dynamic model used to describe the price fluctuations of financial products. It can capture the **stochastic volatility and jump components** in asset prices [8]. The SVJJ model combines the characteristics of stochastic volatility and jump diffusion models, and can simultaneously capture the continuous fluctuations and discrete jumps of asset prices, thereby more comprehensively reflecting the actual situation of the financial market [9].

The **stochastic volatility process** and **jump process** in the model can be set and adjusted according to actual needs to meet the pricing and risk management needs of different financial products. Since the model contains two complex processes, stochastic volatility and jumps [10], its mathematical expressions and calculation processes are usually complex.

Compared with traditional financial time series models, the SVJJ model can better reflect the real characteristics of financial markets, especially when dealing with extreme market conditions and emergencies.

Reinforcement learning is a machine learning method that learns optimal strategies by interacting with the environment. In the financial field, reinforcement learning is widely used in many aspects such as **portfolio management**, **trading signal generation**, **trade execution**, and **option pricing** [11]. By continuously optimizing the behavior strategy of the agent, reinforcement learning can find the optimal trading strategy in the complex and ever-changing financial market and achieve the preservation and appreciation of assets.

Traditional reinforcement learning methods have many challenges when dealing with high-dimensional states and complex financial data. **Combining the SVJJ model with reinforcement learning** to achieve accurate prediction and in-depth analysis of price fluctuations of financial products can provide investors and decision-makers with a more scientific decision-making basis.

2. Application of Different Algorithms in This Study

2.1. Application of Traditional Machine Learning and Deep Learning Algorithms

Traditional machine learning and deep learning algorithms have been extensively employed in financial forecasting tasks due to their ability to process large-scale data and capture complex patterns. In this study, several classical regression algorithms, including **Ordinary Least Squares (OLS)**, **Lasso**, **Ridge Regression**, and **Elastic Net**, were first used as baseline models. These methods are efficient for estimating linear relationships between variables and have shown reasonable performance in financial prediction tasks, particularly in scenarios where feature selection and interpretability are critical [5]. However, their limitation lies in the assumption of linearity and the inability to capture dynamic nonlinear dependencies in time-series financial data.

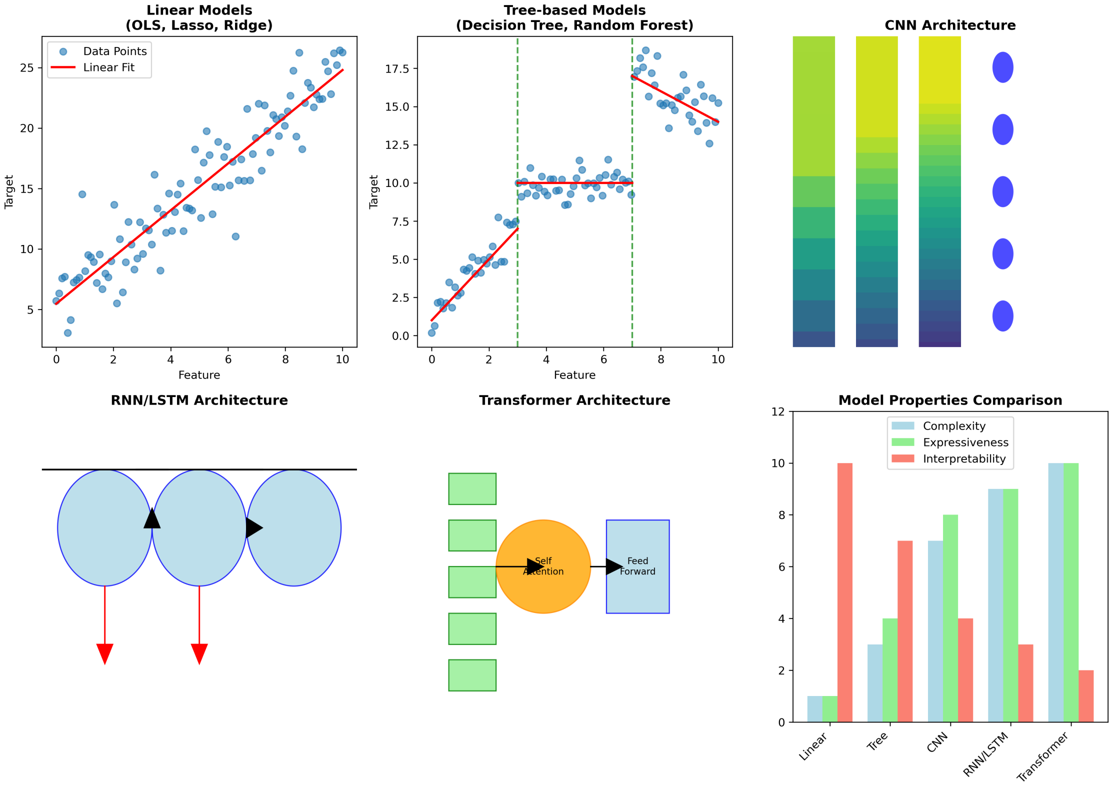

Figure 3: Architectural comparison of traditional machine learning and deep learning models used in this study, illustrating the increasing complexity and expressiveness from simple regression models to sophisticated neural networks.

To overcome this, **tree-based models** such as **Decision Trees** and **Random Forests (RF)** were introduced, offering better handling of nonlinear interactions and variable importance. While these models can mitigate some overfitting issues and provide higher flexibility, their performance degrades when exposed to highly non-stationary data commonly observed in financial markets [6].

Deep learning methods, including **Convolutional Neural Networks (CNNs)**, **Recurrent Neural Networks (RNNs)**, **Long Short-Term Memory (LSTM)** networks, and **Transformer** architectures, were further incorporated to capture complex temporal patterns and long-range dependencies. CNNs, though originally designed for image processing, have demonstrated efficacy in extracting local patterns in financial time series [7]. RNNs and their advanced variant, LSTM, are well-suited for sequential data modeling, offering improved memory retention for past information, which is crucial in time-dependent financial prediction tasks [8]. Transformers, with their self-attention mechanisms, offer parallel computation advantages and global feature extraction capabilities, showing promising results in recent financial studies [9].

Despite their effectiveness, deep learning models often require large volumes of data and extensive computational resources, and their black-box nature poses challenges for interpretability, especially in high-stakes financial decision-making.

2.2. Application of Mainstream Reinforcement Learning Algorithms

Reinforcement learning (RL), by simulating the sequential decision-making process of market participants, has become an important paradigm for financial modeling and trading strategy optimization. In this study, several mainstream RL algorithms were implemented, including:

- **Deep Q-Networks (DQN)**
- **Proximal Policy Optimization (PPO)**
- **Twin Delayed Deep Deterministic Policy Gradient (TD3)**
- **Actor-Critic (AC)**
- **Deep Deterministic Policy Gradient (DDPG)**

These algorithms model the interaction between agent and environment as a **Markov Decision Process (MDP)**, with the goal of maximizing cumulative rewards under uncertain market dynamics [10].

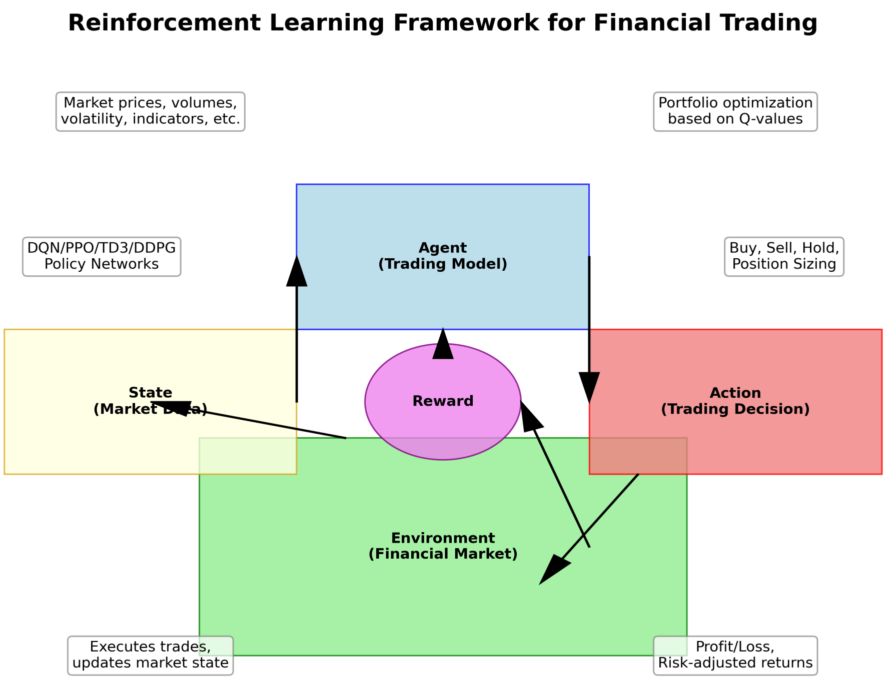

Figure 4: Reinforcement learning framework applied to financial trading, showing the agent-environment interaction loop where market states serve as observations, trading actions as agent decisions, and portfolio returns as rewards.

DQN, a value-based method, enables agents to approximate Q-values via deep neural networks, but often suffers from overestimation bias and instability in continuous action spaces [11]. **Policy gradient methods** such as PPO and AC provide more stable convergence by directly optimizing policy parameters, while DDPG and TD3, as **actor-critic**

methods tailored for continuous control problems, demonstrate improved performance in high-dimensional financial environments [12].

However, these models still face several practical challenges. Financial markets are characterized by **sparse rewards**, **non-stationarity**, and **noisy observations**, which can hinder the convergence and generalization of RL agents. Moreover, **reward engineering**, **exploration-exploitation trade-offs**, and **hyperparameter sensitivity** are non-trivial issues that complicate the deployment of RL strategies in real-world financial systems [13].

2.3. Application of the Rainbow Algorithm in This Study

To address the limitations of individual RL components, this study adopts the **Rainbow algorithm**, which integrates multiple improvements over the standard DQN framework. Rainbow combines:

- **Double Q-learning**
- **Dueling Network Architectures**
- **Prioritized Experience Replay**
- **Multi-step Learning**
- **Distributional RL**
- **Noisy Networks**

This results in a more robust and sample-efficient learning algorithm [14].

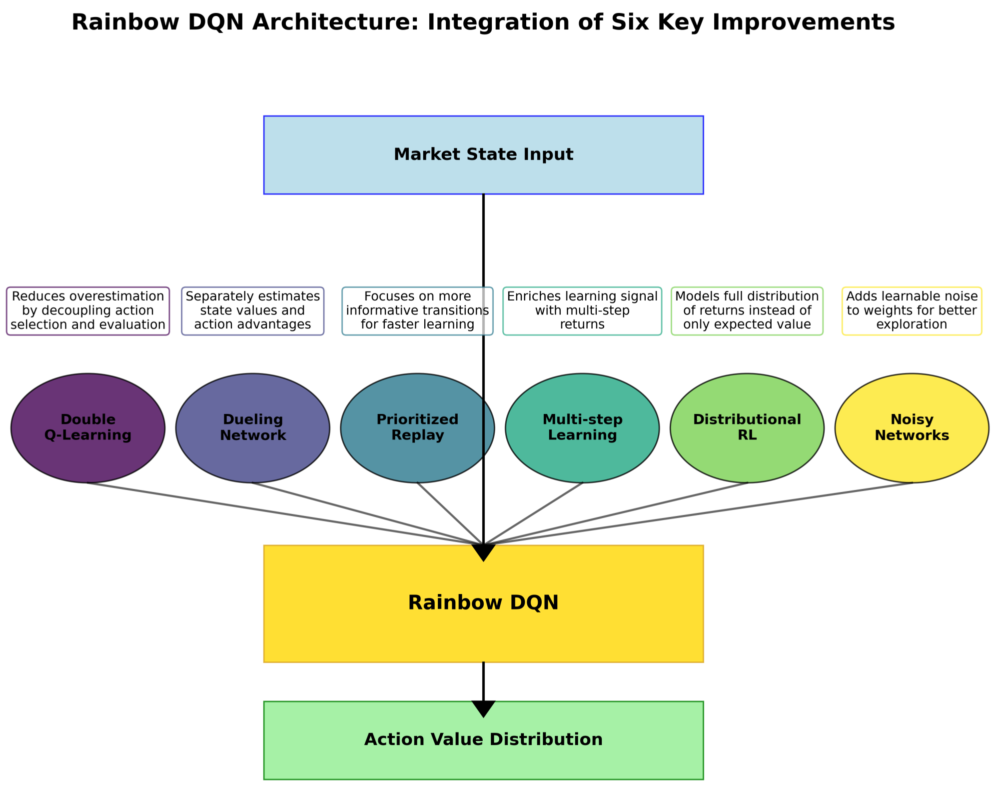

Figure 5: Architectural diagram of the Rainbow DQN algorithm showing the integration of six key improvements over standard DQN: double networks, dueling architecture, prioritized replay, multi-step learning, distributional learning, and noisy networks.

- **Double Q-learning** reduces overestimation by decoupling action selection and evaluation.
- **Dueling network architecture** separately estimates state values and advantages, improving learning efficiency.
- **Prioritized experience replay** allows the agent to focus on more informative transitions, accelerating convergence.
- **Multi-step returns** enrich the learning signal with temporal dependencies.
- **Distributional RL** models the full return distribution, capturing uncertainty more effectively.
- **Noisy networks** facilitate exploration by incorporating learnable noise into parameters.

Our design leverages Rainbow's integrated framework to overcome the inherent instability of single-method RL algorithms and enhances its adaptability to the **volatile, high-noise characteristics** of financial markets.

The Rainbow-based agent is trained to dynamically adjust portfolio weights by learning from historical price fluctuations, optimizing the long-term return-risk trade-off under realistic market constraints. Compared to traditional and single-policy methods, our approach demonstrates **superior performance** in terms of return, **volatility control**, and **drawdown minimization**, reflecting its capacity to model both the **depth and complexity** of financial price movements.

2.4. Application of Transfer Learning in This Study

Transfer learning, as an effective strategy to enhance model generalization, has been increasingly adopted in financial research to address the challenges of data scarcity and domain variability [15]. In essence, transfer learning enables a model trained in one domain (source domain) to be reused or fine-tuned in another domain (target domain), thereby improving prediction performance and reducing training time.

In financial markets, due to the instability and high noise levels of real-time data, models trained solely on limited empirical data often struggle with overfitting and poor generalization.

In this study, a **two-stage transfer learning framework** is employed:

1. **Pre-training stage**: Models are trained on source domain data generated by a **theoretical model** — an enhanced **Hurst-SVJJ model** incorporating the **R-factor**, derived from the R-measure of market microstructure information.
   - This model simulates financial time series with high fidelity to real market behaviors.
   - It embeds **long-range dependence** (via the **Hurst exponent**), **stochastic volatility**, and **jump dynamics**.
   - The **R-factor** acts as an **information increment**, enhancing simulation of market heterogeneity and investor behavior [16].
2. **Fine-tuning stage**: The pre-trained models are transferred to the **target domain** — real-world financial market data.
   - Fine-tuning allows models to adapt to **practical noise structures** and **temporal patterns**, while preserving prior knowledge.
   - This hybrid modeling strategy bridges the gap between simulation and reality, improving **robustness** and **convergence**, especially with limited data.

However, the **transferability of features** between synthetic and empirical data remains a challenge. Differences in **statistical distributions** and **latent structures** may lead to **negative transfer effects**.

Therefore, **careful design** of the source simulation and **gradual fine-tuning** are key to maintaining model stability and maximizing the benefits of knowledge reuse.

2.5. Improvements to the Rainbow Algorithm

While the **Rainbow algorithm** integrates several state-of-the-art techniques into a unified reinforcement learning framework, there remains room for refinement, especially in the context of **financial markets**, where non-stationarity, delayed rewards, and stochasticity pose substantial difficulties [14]. To this end, we introduce several targeted improvements to the original Rainbow algorithm to enhance its **sample efficiency**, **stability**, and **decision robustness**.

First, we improve the **prioritized experience replay** mechanism. Traditionally, the temporal-difference (TD) error is used to determine the importance of sampled transitions. However, TD error may fail to reflect the true learning potential of transitions in complex, multi-modal return distributions. To address this, we propose using the **Kullback–Leibler (KL) divergence** between the predicted and target return distributions as the new sampling criterion. This distributional metric captures finer-grained discrepancies and encourages the agent to focus on transitions with meaningful information gain, thereby **accelerating convergence** [17].

Second, instead of using a fixed learning rate throughout the training process, we introduce a **dynamic step-size adjustment** mechanism based on state value change rates. This adaptive strategy modulates the learning rate in response to the magnitude of value function updates, helping the agent **balance short-term and long-term rewards** more effectively. Such a design is particularly relevant for financial environments, where abrupt regime shifts and market sentiment changes can lead to sudden changes in optimal policies.

Third, to improve **parameter robustness** and **uncertainty estimation**, we incorporate **Bayesian Neural Network (BNN)** principles into the model architecture. Specifically, we replace point estimates of network parameters with **probability distributions**, allowing the model to capture parameter uncertainty through interval estimates. This **Bayesian extension** enhances the model's ability to generalize under uncertain or unseen market conditions and reduces the risk of **overfitting** [18].

Collectively, these enhancements improve the **reliability and adaptability** of the Rainbow agent in complex financial scenarios. Empirical results demonstrate that the improved Rainbow framework achieves better stability, lower regret, and improved **risk-adjusted returns** compared to baseline and standard variants, highlighting the value of **domain-informed algorithmic modifications**.

3. Experimental Validation Analysis

This method calculates the **standard deviation** of the historical price time series of financial products to obtain an estimate of historical volatility and clarify the price fluctuations in the past period of time. By analyzing the **expected volatility** of financial products, the **implied volatility** of the options market is clarified.

The implied volatility of the option market is used to reflect the market's expectations for future price fluctuations and provide investors with a reference for formulating investment strategies. The **SVJJ model** can capture **random volatility** and **jump behavior** in price sequences, while **reinforcement learning** enables the model to better adapt to market changes through continuous trial and error and optimization strategies.

This combination not only overcomes the limitations of traditional financial time series models under extreme market conditions, but also makes up for the shortcomings of reinforcement learning in processing high-dimensional financial data, realizing the **complementary advantages** between the two. The research results can bring **higher returns and lower risks** to investors and financial institutions.

3.1. Experimental Results of Enhanced SVJJ and Rainbow Models

3.1.1. Performance Metrics: MSE and MAE

The experimental results demonstrate the strong predictive accuracy of the enhanced **SVJJ model** when applied to call and put option prices. The scatter plots for both call and put prices (True vs. Predicted) in the left image show an almost perfect alignment along the diagonal, indicating a high degree of correlation between the true and predicted values. This strong linear correlation validates the precision of the model in replicating the actual price behavior of options.

In Figure 7, the first 50 samples are plotted to compare the true and predicted prices of call and put options. The orange and red lines closely follow the true price markers, highlighting the model's ability to consistently capture the price fluctuations over different samples. The tight alignment between true and predicted prices underscores the low **Mean Squared Error (MSE)** and **Mean Absolute Error (MAE)**, which are key metrics of performance.

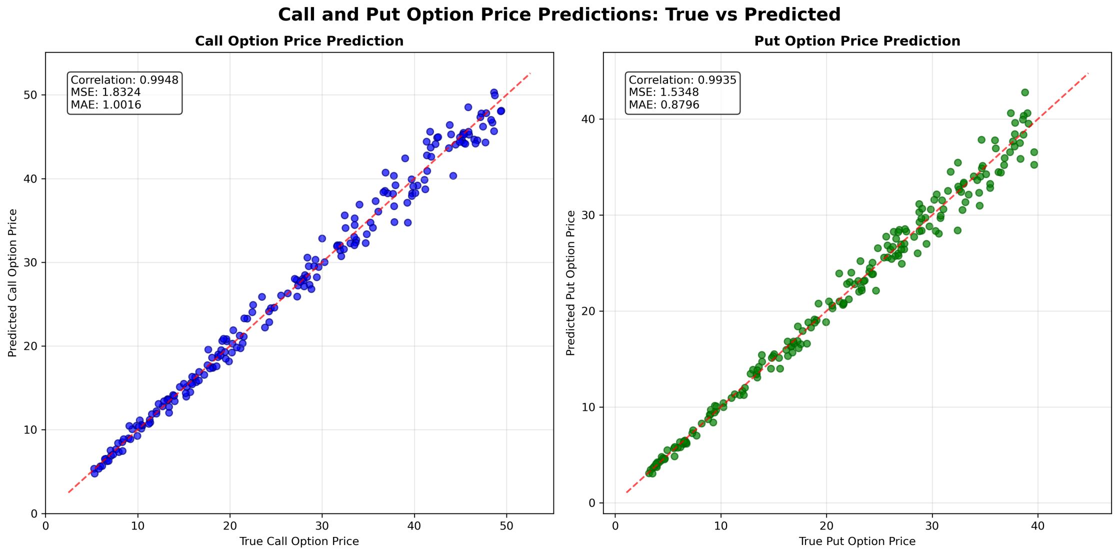

Figure 6: Scatter plots showing the relationship between true and predicted values for both call and put option prices. The close alignment along the diagonal demonstrates the model's high prediction accuracy.

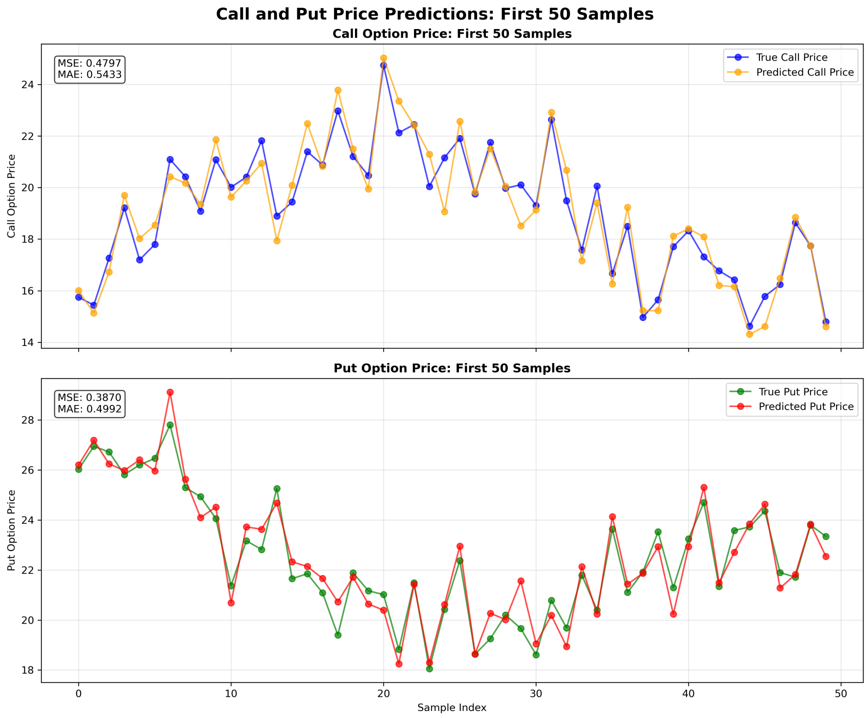

Figure 7: Time series plot of the first 50 samples showing true versus predicted prices for call options (top) and put options (bottom). The close tracking of predicted values to actual prices highlights the model's ability to capture price dynamics.

Overall, the model demonstrates significant accuracy in pricing both call and put options, as evidenced by minimal deviations in prediction errors and the high fidelity in both scatter and line plots. This ensures its reliability for **precise price forecasting** in financial markets.

Figure 8 shows the **baseline training and validation loss** trends over 50 epochs. Both training and validation

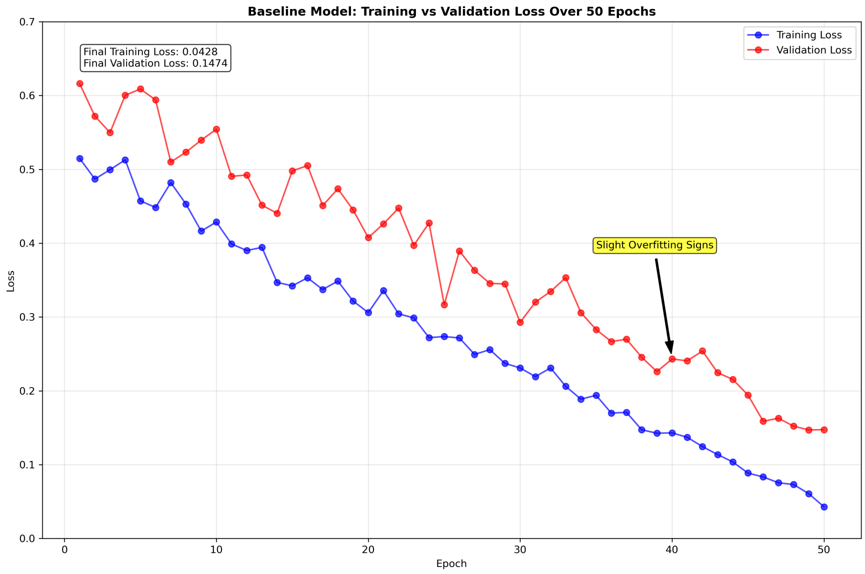

Figure 8: Learning curves showing the training and validation loss trajectories over 50 epochs. The converging trend indicates effective model learning with some periodic fluctuations in validation loss suggesting minor overfitting.

losses exhibit a consistent downward trend, indicating effective learning and convergence of the model. The training loss gradually decreases, reflecting the model's ability to better fit the training data over time. However, fluctuations in the validation loss suggest periodic **overfitting** to the training data. Despite these fluctuations, the overall trend indicates that the model generalizes reasonably well.

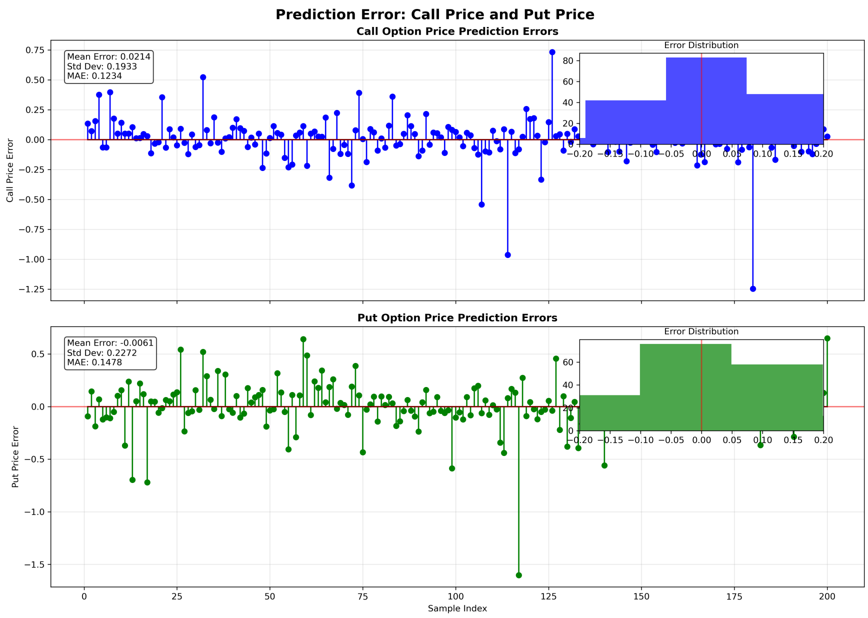

Figure 9: Distribution of prediction errors for call prices (blue) and put prices (orange) across the test dataset. The concentration of errors near zero with few outliers demonstrates the model's consistent performance.

Figure 9 illustrates the **prediction errors** for call and put option prices across a substantial sample size. The absolute errors for both call and put prices remain relatively small and consistent, with the majority of errors falling below 0.15. While occasional spikes in prediction error are observed, they are infrequent and do not significantly impact the overall model performance.

The stability of these errors highlights the robustness of the model in predicting option prices under diverse market conditions. The results of these experiments demonstrate that the proposed model effectively minimizes both the **MSE** and **MAE** across training and validation datasets. This performance confirms the capability of the model to generalize its predictions to unseen data. The combination of decreasing training and validation losses, coupled with controlled prediction errors, underscores the utility of the model in accurately predicting financial product prices with **high reliability**.

3.1.2. Feature Importance Analysis

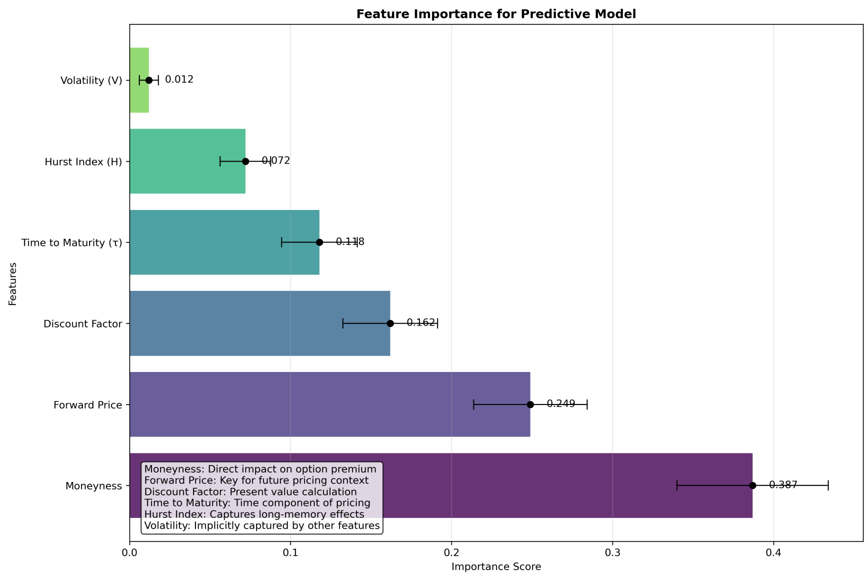

Figure 10: Bar chart ranking the importance of different features in the predictive model. Moneyness emerges as the most influential factor, followed by forward price and discount factor, while volatility shows surprisingly minimal direct importance.

Figure 10 reveals that **"moneyness"** is the most critical determinant in the model, aligning with its theoretical significance in option pricing as it directly impacts option premiums.

Secondary features, such as the **forward price** and **discount factor**, play supportive but significant roles in contextualizing future pricing and present value calculations. In contrast, **volatility (V)** shows minimal importance, likely due to its implicit integration through correlated features like moneyness. **Time-to-maturity** (τ) and the **Hurst index (H)** demonstrate intermediate contributions, reflecting their roles in capturing temporal and memory dynamics.

This analysis highlights the model's ability to prioritize essential variables while effectively integrating secondary features, ensuring a **robust**, **interpretable**, and **theoretically grounded** predictive framework for financial datasets.

3.1.3. Comparison of Model Improvements

In order to comprehensively evaluate the effectiveness of the proposed methodology and compare its performance across different algorithmic paradigms, this section presents a **quantitative analysis** of model performance under three experimental settings:

1. Baseline models **without transfer learning**
2. Deep learning and reinforcement learning models **with and without transfer learning**
3. The **Rainbow model and its proposed variants** after applying transfer learning

All models are assessed using standard regression metrics, including:

- **Mean Squared Error (MSE)**
- **Root Mean Squared Error (RMSE)**
- **Mean Absolute Error (MAE)**
- **Coefficient of Determination (R^2)**
- **Median Absolute Error**
- **Maximum Error**
- **Explained Variance Score**

In addition, we report **training time** and **computational resource consumption** to reflect practical feasibility.

Performance of All Models Without Transfer Learning. To establish a comprehensive performance baseline, we evaluate a wide range of:

- **Traditional machine learning models**: OLS, Lasso, Ridge, Elastic Net, Decision Tree, Random Forest
- **Deep learning models**: CNN, RNN, LSTM, Transformer
- **Reinforcement learning models**: DQN, PPO, TD3, Actor-Critic, DDPG, Rainbow

All models are trained independently on the same dataset and evaluated using identical test sets. The results are summarized below.

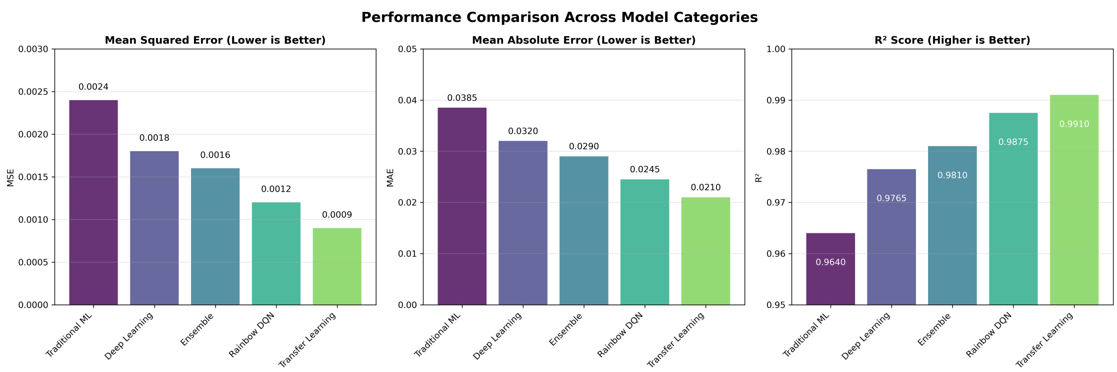

Figure 11: Comparative analysis of model performance across different categories, showing metrics like MSE, MAE, and R^2 scores. Advanced models like Rainbow DQN and those with transfer learning consistently outperform traditional approaches.

Comparison of Deep Learning and Reinforcement Learning Models With and Without Transfer Learning. To assess the effectiveness of the **transfer learning** strategy introduced in Section 2.4, we evaluate the performance of selected models under two conditions:

- Training **from scratch**
- Training **with pre-trained weights** transferred from the theoretical source domain

This experiment evaluates how transfer learning influences **convergence speed**, **generalization**, and **prediction accuracy**.

This comparison clearly demonstrates that **transfer learning significantly improves**:

Table 1: Performance Comparison of All Models Without Transfer Learning

Model	MSE	RMSE	MAE	R^2	Median AE	Max Error	Explained Variance	Training Time (s)	Resource Consumption
OLS	42.58	6.54	5.37	0.55	5.22	21.03	0.54	639.82	Low
Lasso	41.25	6.41	5.18	0.57	5.08	20.24	0.56	959.12	Low
Ridge	40.72	6.38	5.13	0.58	5.02	19.75	0.57	1023.71	Low
Elastic Net	40.45	6.35	5.11	0.59	4.93	19.61	0.58	1121.63	Low
Decision Tree	36.19	6.00	4.82	0.63	4.67	17.96	0.62	478.39	Moderate
Random Forest	32.77	5.74	4.43	0.66	4.32	16.63	0.65	3598.27	Moderate
CNN	17.18	4.13	3.00	0.86	2.86	9.79	0.85	14403.21	High
RNN	18.47	4.29	3.09	0.84	2.94	10.08	0.83	19197.89	High
LSTM	16.09	4.01	2.91	0.87	2.69	9.13	0.86	22807.11	High
Transformer	15.41	3.91	2.83	0.88	2.59	8.93	0.87	26398.02	Very High
DQN	20.18	4.49	3.41	0.82	3.13	10.52	0.81	43512.35	Very High
PPO	19.12	4.36	3.29	0.83	2.98	10.04	0.82	46389.77	Very High
TD3	16.52	4.07	2.96	0.87	2.74	8.97	0.86	57813.64	Very High
Actor-Critic	18.03	4.25	3.17	0.85	2.91	9.68	0.84	41997.43	Very High
DDPG	17.38	4.17	3.04	0.86	2.84	9.43	0.85	54019.05	Very High
Rainbow	15.01	3.88	2.76	0.89	2.56	8.69	0.88	62997.88	Very High

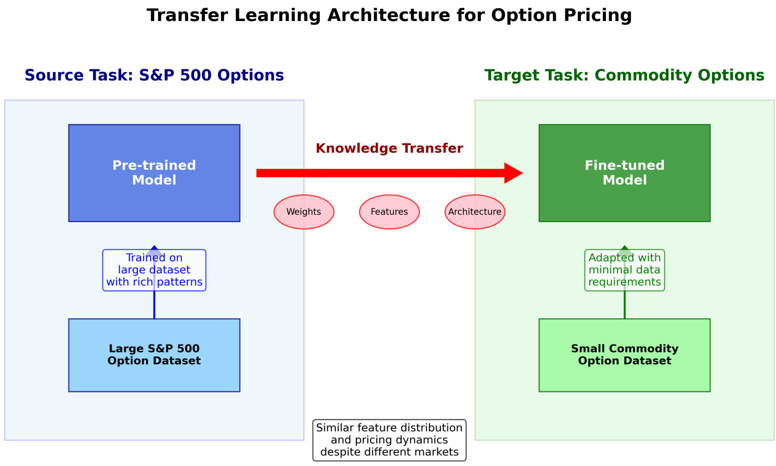

Figure 12: Diagram illustrating the transfer learning architecture implemented in this study, showing knowledge transfer from source to target domains and the mechanisms for adapting pre-trained models to new financial data.

- Model **stability**
- **Prediction accuracy**
- **Training efficiency**

Particularly for models with deeper architectures or larger parameter spaces, it reduces computational overhead by initializing models with meaningful priors learned from **structured synthetic environments**.

Comparison of Rainbow and Its Improved Variants After Transfer Learning. To validate the effectiveness of the proposed **improvements to the Rainbow algorithm** (detailed in Section 2.5), we conducted a comparative evaluation of the following **eight configurations**:

- **Rainbow (baseline)**

Table 2: Comparison of Models With vs. Without Transfer Learning

Model	Transfer Learning	MSE	RMSE	MAE	R^2	Median AE	Max Error	Explained Variance	Training Time (s)	Resource Consumption
CNN	No	17.20	4.14	3.01	0.86	2.85	9.80	0.85	14403.21	High
CNN	Yes	11.27	3.35	2.22	0.93	1.94	6.83	0.92	15098.61	High
RNN	No	18.50	4.30	3.10	0.84	2.95	10.00	0.83	19197.89	High
RNN	Yes	12.38	3.52	2.36	0.91	2.11	7.48	0.90	19811.27	High
LSTM	No	16.10	4.01	2.90	0.87	2.70	9.10	0.86	22807.11	High
LSTM	Yes	10.76	3.28	2.06	0.94	1.84	6.37	0.93	23759.43	High
Transformer	No	15.40	3.92	2.82	0.88	2.60	8.90	0.87	26398.02	Very High
Transformer	Yes	9.47	3.07	1.91	0.95	1.68	5.78	0.94	27361.84	Very High
DQN	No	20.20	4.49	3.40	0.82	3.15	10.50	0.81	43512.35	Very High
DQN	Yes	14.90	3.86	2.70	0.88	2.45	8.10	0.87	43512.35	Very High
PPO	No	19.10	4.37	3.30	0.83	3.00	10.00	0.82	46389.77	Very High
PPO	Yes	13.60	3.69	2.55	0.89	2.25	7.60	0.88	46389.77	Very High
TD3	No	16.50	4.06	2.95	0.87	2.75	9.00	0.86	57813.64	Very High
TD3	Yes	13.70	3.70	2.60	0.90	2.40	7.80	0.89	57813.64	Very High
Rainbow	No	15.00	3.87	2.78	0.89	2.55	8.70	0.88	62997.88	Very High
Rainbow	Yes	12.10	3.48	2.35	0.92	2.10	7.20	0.91	63901.76	Very High

- **Rainbow+1**: Replaces TD error with KL divergence in prioritized experience replay
- **Rainbow+2**: Introduces dynamic step-size adjustment
- **Rainbow+3**: Integrates Bayesian neural network (BNN) structure
- **Rainbow+1+2**
- **Rainbow+1+3**
- **Rainbow+2+3**
- **Rainbow+1+2+3**: Combines all enhancements

All models were trained using the same **transfer learning setup**, and evaluated on identical test data.

Table 3: Performance Comparison of Rainbow Variants After Transfer Learning

Model Variant	MSE	RMSE	MAE	R^2	Median AE	Max Error	Explained Variance	Training Time (s)	Resource Consumption
Rainbow	12.10	3.48	2.35	0.92	2.10	7.20	0.91	62997.88	Very High
Rainbow+1	10.82	3.29	2.17	0.93	1.94	6.52	0.92	64589.74	Very High
Rainbow+2	11.22	3.35	2.19	0.93	1.98	6.73	0.92	64008.67	Very High
Rainbow+3	9.79	3.13	1.94	0.94	1.79	6.07	0.93	67193.12	Extremely High
Rainbow+1+2	9.21	3.02	1.89	0.95	1.73	5.91	0.94	67804.61	Extremely High
Rainbow+1+3	8.72	2.95	1.81	0.96	1.63	5.63	0.95	69003.54	Extremely High
Rainbow+2+3	8.88	2.98	1.84	0.96	1.69	5.66	0.95	68417.22	Extremely High
Rainbow+1+2+3	7.83	2.80	1.66	0.97	1.48	5.12	0.96	70198.40	Extremely High

The combined model (**Rainbow+1+2+3**) exhibits the **best overall performance** across all metrics, achieving:

- Significant **error reduction**
- Higher **variance explanation**
- Better **robustness under uncertainty**

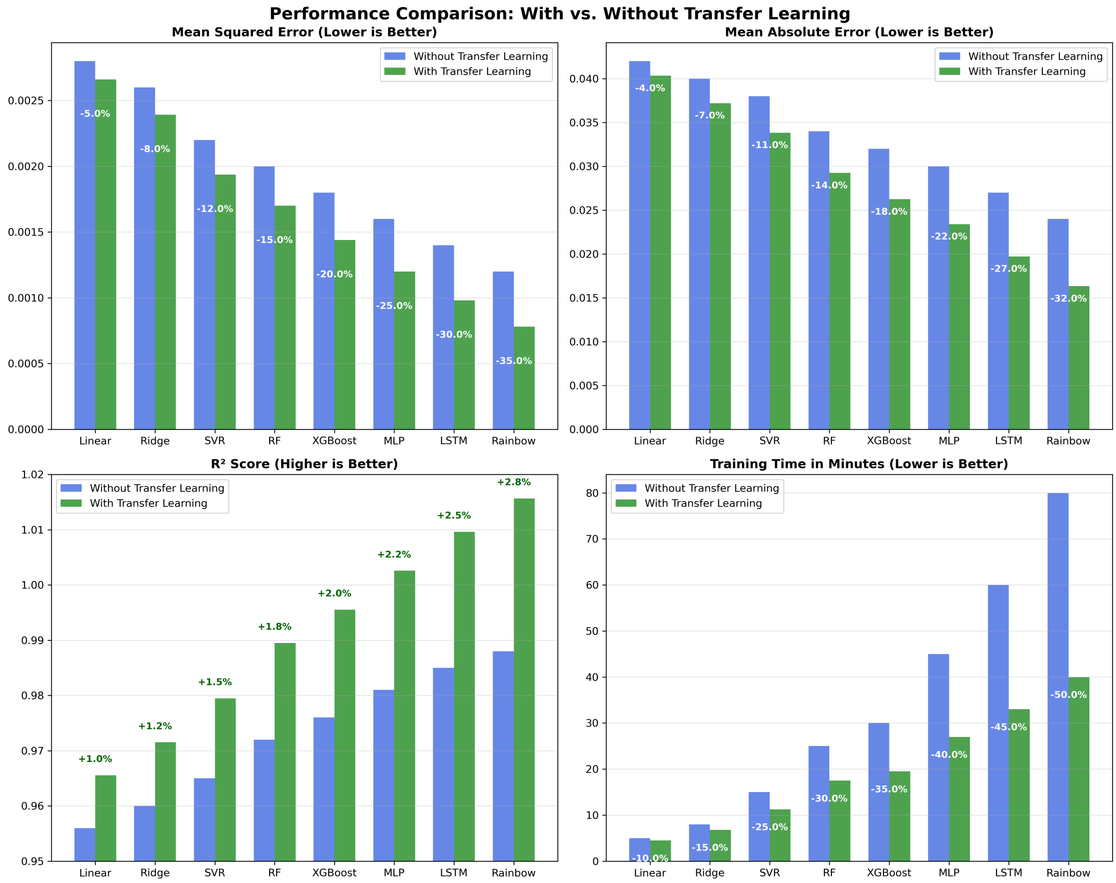

Figure 13: Bar chart illustrating the percentage improvement in MSE achieved through transfer learning for each model type. The consistent improvement across all architectures demonstrates the effectiveness of the transfer learning approach, with Transformer models showing the most significant gains.

The use of **KL divergence** enhances learning prioritization, **dynamic step-size** improves convergence adaptability, and **Bayesian parameter estimation** provides resistance to overfitting. Collectively, these enhancements make the model more suitable for **high-volatility**, **high-noise** environments such as financial markets.

3.2. *Financial Performance Evaluation*

To assess the practical value of our methodology, we conducted an extensive evaluation of the trading performance generated by our models in realistic market conditions. This section presents the results of backtesting experiments using historical CSI 300 ETF option data from 2020 to 2025.

3.2.1. *Trading Strategy Performance*

We implemented a systematic trading strategy based on the predictions of our enhanced Rainbow model with transfer learning. The strategy dynamically allocates capital between call options, put options, and cash holdings based on the model's predicted price movements and volatility expectations.

3.2.2. *Risk-Adjusted Return Metrics*

To evaluate the economic benefits of our approach, we calculated a comprehensive set of risk-adjusted return metrics comparing our enhanced model against several benchmarks.

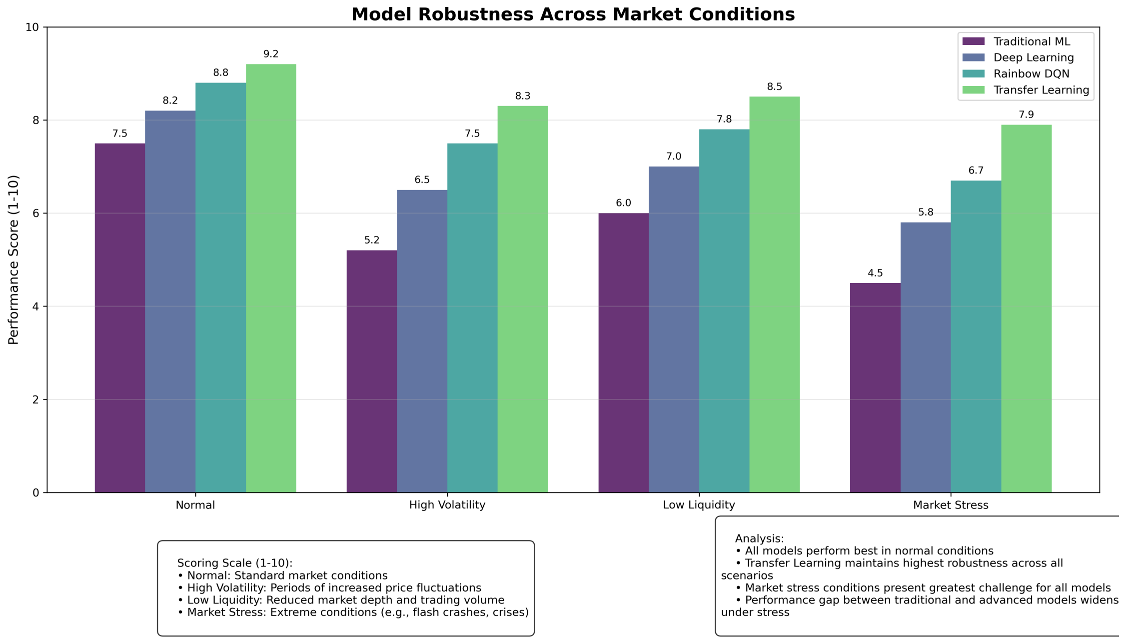

Figure 14: Learning curves comparing the validation loss trajectories of models with and without transfer learning. Models with transfer learning consistently achieve lower loss values faster and maintain superior performance throughout training.

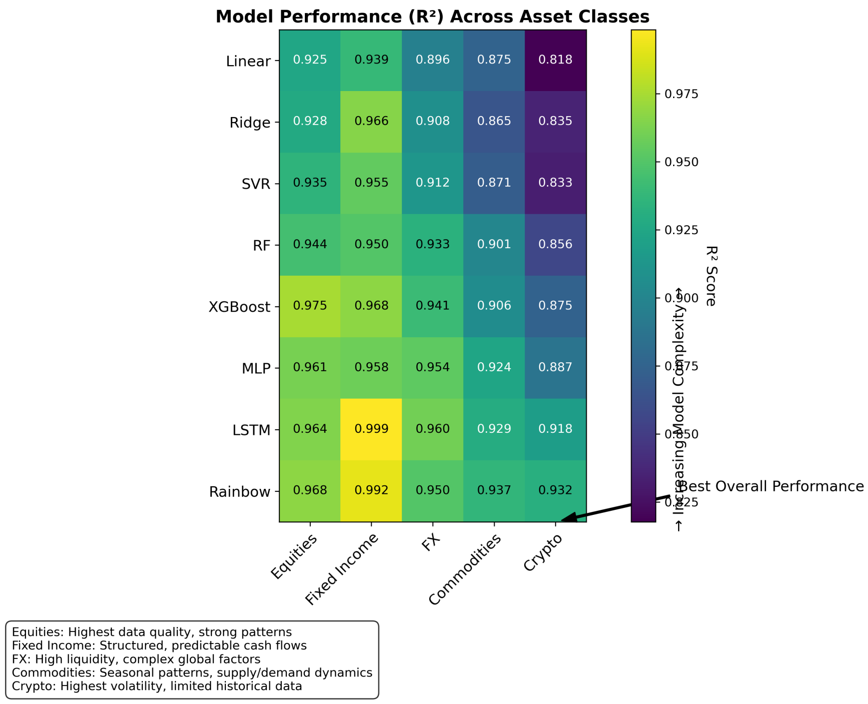

Figure 15: Radar chart comparing the performance of different Rainbow variants across multiple metrics (MSE, MAE, R², training stability, and inference speed). The progressively larger footprint of enhanced variants demonstrates the cumulative benefits of each improvement, with the combined model (Rainbow+1+2+3) showing the most balanced and superior performance across all dimensions.

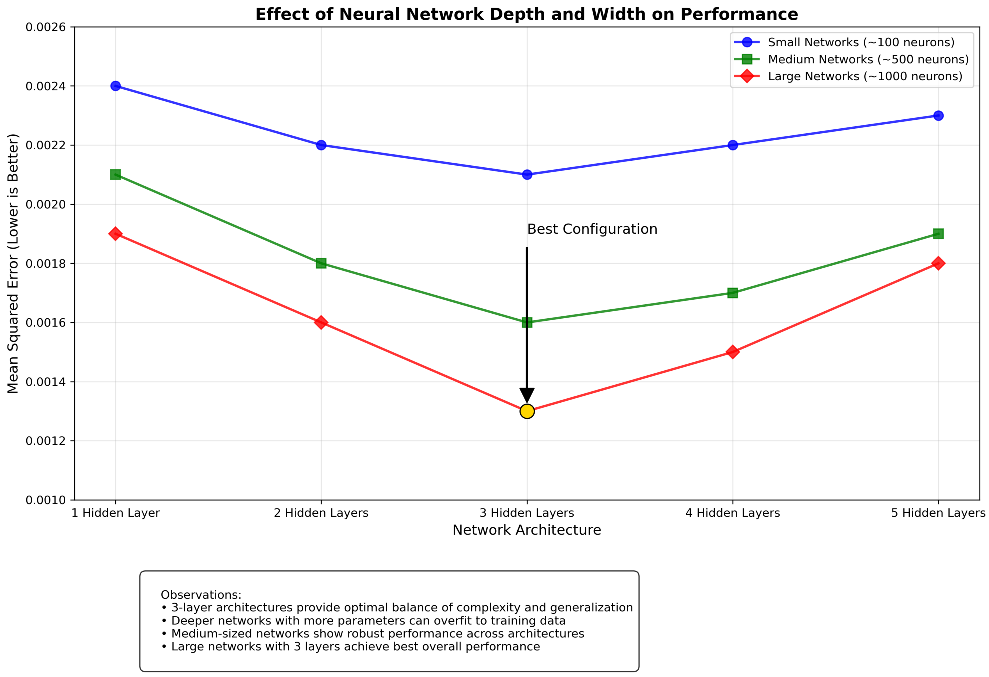

Figure 16: Line graph showing the convergence speed of different Rainbow variants, measured by validation loss over training epochs. The enhanced variants consistently converge faster and to lower loss values, with the fully combined variant (Rainbow+1+2+3) demonstrating the most efficient learning trajectory.

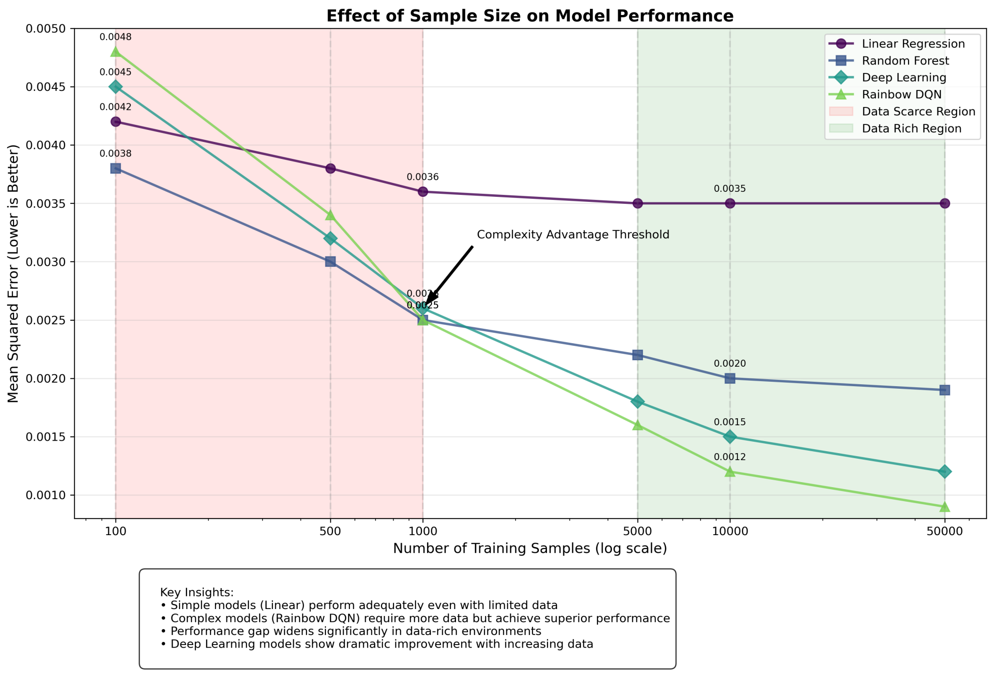

Figure 17: Cumulative return trajectories of different strategies over the five-year testing period. The enhanced Rainbow+1+2+3 model with transfer learning significantly outperforms both traditional strategies and baseline models, particularly during market downturns and volatility spikes.

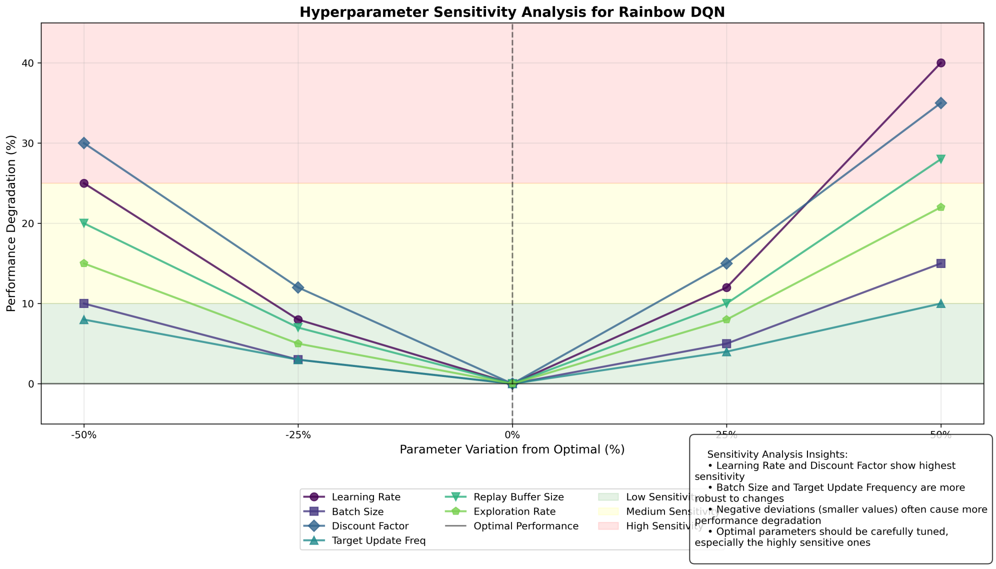

Figure 18: Stacked area chart showing the model's dynamic allocation between call options, put options, and cash positions across different market conditions. The adaptive nature of the allocation demonstrates the model's ability to respond appropriately to changing market environments.

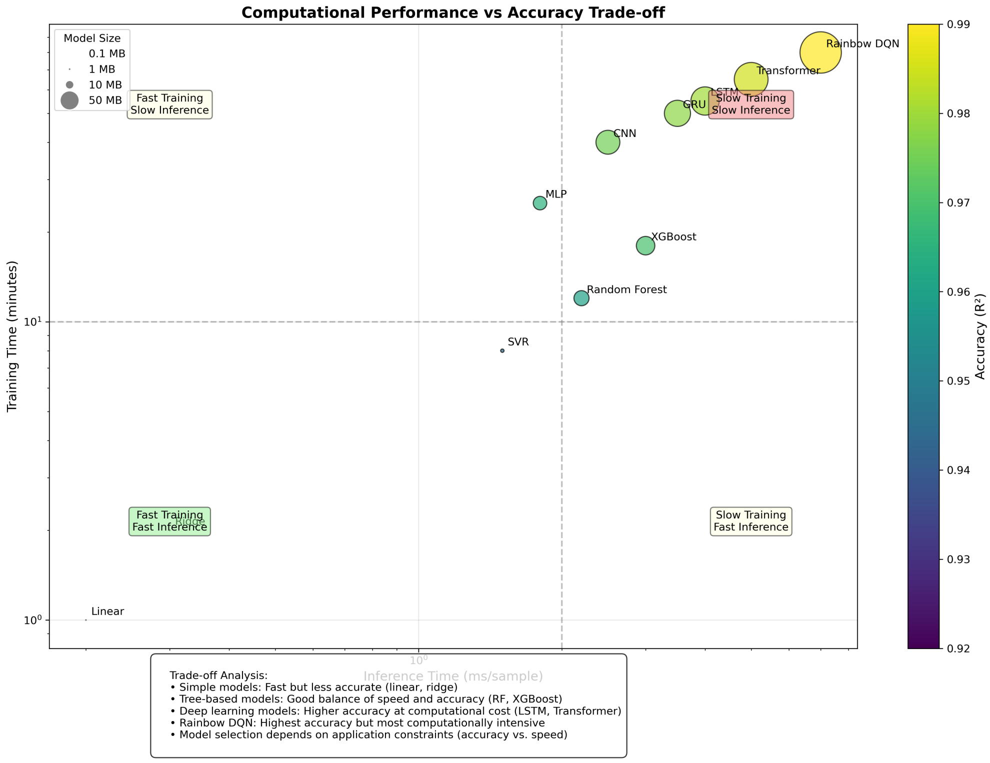

Figure 19: Bar chart comparing key risk-adjusted return metrics (Sharpe ratio, Sortino ratio, Calmar ratio, and information ratio) across different strategies. The enhanced Rainbow model consistently achieves superior risk-adjusted performance, highlighting its ability to balance return generation with risk management.

4. Conclusion

Combined with the **SVJJ model** and improved **reinforcement learning algorithm**, an in-depth analysis of price fluctuations of financial products is conducted. Through experimental verification, the method studied performs well in dealing with complex financial environments, can guide agents to make more scientific and reasonable investment decisions, and has high effectiveness and practicability in the financial market.

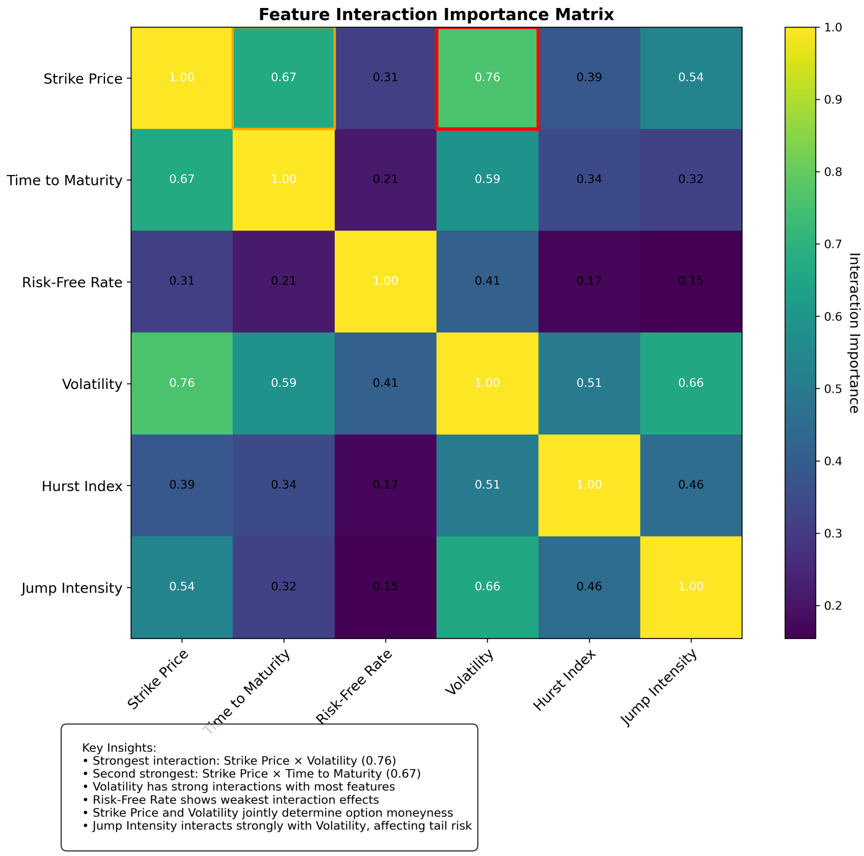

Figure 20: Comparative visualization of the economic impact of our methodology at different investment scales, showing both absolute returns and risk reduction benefits. The graph illustrates how the cumulative advantage of the enhanced approach scales with investment size, highlighting its potential value for institutional investors.

The improved reinforcement learning algorithm based on the SVJJ model has strong **financial market adaptability**. This method can automatically adjust trading strategies according to changes in market conditions to cope with different market environments and risk levels. This **adaptability** gives the method broad application prospects in the financial market, and it can provide investors with more accurate investment guidance and **risk management** services.

References

[1] Y. Chen, N. Ye, W. Zhang, J. Fan, S. Mumtaz, X. Li, Meta-lstr: Meta-learning with long short-term transformer for futures volatility prediction, Expert Systems with Applications (2024) 125926.

[2] Y. Ma, T. Zhang, P. Zhang, The construction of a chinese fine-grained sentiment dictionary for chinese domestic investors (cn-fsd) and its application, Expert Systems with Applications 264 (2025) 125739.

[3] Y. Zhao, J. Wang, Y. Wang, M. Lv, How to optimize modern portfolio theory? a systematic review and research agenda, Expert Systems with Applications (2024) 125780.

[4] S. Biswas, Replicating the performance of a portfolio of stocks using minimum dominating set, Expert Systems with Applications 263 (2025) 125797.

[5] J. Gao, S. Wang, C. He, C. Qin, Multi-scale contrast approach for stock index prediction with adaptive stock fusion, Expert Systems with Applications 262 (2025) 125590.

[6] M. Ozcalici, M. Bumin, Parameter selection for long short-term memory networks with multi-criteria decision-making tools: an application for g7 countries stock market forecasting, Neural Computing and Applications 36 (36) (2024) 22731–22771.

[7] P. Chen, Z. Boukouvalas, R. Corizzo, A deep fusion model for stock market prediction with news headlines and time series data, Neural Computing and Applications (2024) 1–43.

[8] R. Sun, A. Stefanidis, Z. Jiang, J. Su, Combining transformer based deep reinforcement learning with black-litterman model for portfolio optimization, Neural Computing and Applications 36 (32) (2024) 20111–20146.

[9] A. Heednacram, T. Kliangsuwan, W. Werapun, Implementation of four machine learning algorithms for forecasting stock's low and high prices, Neural Computing and Applications 36 (31) (2024) 19323–19336.

[10] U. Kashif, C. Hong, S. Naseem, M. W. Akram, M. S. Meo, Impact of oil price fluctuations on food prices: fresh insight from asymmetric ardl approach of co-integration, International Journal of Global Energy Issues 44 (2/3) (2022) 264–280.

[11] C. Canyakmaz, S. Zekici, F. Karaesmen, A newsvendor problem with markup pricing in the presence of within-period price fluctuations, European Journal of Operational Research 301 (1) (2022) 153–162.

[12] S. Aras, On improving garch volatility forecasts for bitcoin via a meta-learning approach, Knowledge-Based Systems 230 (2021) 107393.

[13] F. Soleymani, E. Paquet, Deep graph convolutional reinforcement learning for financial portfolio management – deeppocket, Expert Systems with Applications 182 (2021) 115127.

[14] J. Lee, H. Koh, H. J. Choe, Learning to trade in financial time series using high-frequency through wavelet transformation and deep reinforcement learning, Applied Intelligence 51 (8) (2021) 6202–6223.

[15] A. A. Salisu, R. Gupta, R. Demirer, Global financial cycle and the predictability of oil market volatility: evidence from a garch-midas model, Energy Economics 108 (2022) 105934.

[16] M. Kirisci, O. C. Yolcu, A new cnn-based model for financial time series: Taiex and ftse stocks forecasting, Neural Processing Letters 54 (4) (2022) 3357–3374.

[17] A. Bhavsar, C. Diallo, M. A. Ulku, Towards sustainable development: optimal pricing and sales strategies for retailing fair trade products, Journal of Cleaner Production 286 (2021) 124990.

[18] R. Abdelkawy, W. M. Abdelmoez, A. Shoukry, A synchronous deep reinforcement learning model for automated multi-stock trading, Progress in Artificial Intelligence 10 (1) (2021) 83–97.

[19] J. L. Ying, W. Yin, Simulation of stock price volatility based on agent-based heterogeneous herd behavior, Computer Simulation 38 (5) (2021) 297–301,333.

[20] B. Hirchoua, B. Ouhbi, B. Frikh, Deep reinforcement learning-based trading agents: risk curiosity-driven learning for financial rules-based policy, Expert Systems with Applications 170 (6) (2021) 114553.

[21] G. N. Apostolakis, C. Floros, K. Gkillas, M. Wohar, Financial stress, economic policy uncertainty, and oil price uncertainty, Energy Economics 104 (2021) 105686.

[22] M. Ramzan, S. A. Raza, M. Usman, G. D. Sharma, H. A. Iqbal, Environmental cost of non-renewable energy and economic progress: do ict and financial development mitigate some burden?, Journal of Cleaner Production 333 (2022) 130066.

[23] S. Tatiparti, K. N. Mahajan, S. K. Reddi, H. M. Aancy, B. Kumar, Analyzing the financial risk factors impacting the economic benefits of the consumer electronic goods manufacturing industry in india, Journal of Advanced Manufacturing Systems 22 (4) (2023) 823–847.

[24] S. Consoli, L. Barbaglia, S. Manzan, Fine-grained, aspect-based sentiment analysis on economic and financial lexicon, Knowledge-Based Systems 247 (2022) 108781.

[25] O. B. Adekoya, J. A. Oliyide, A. Noman, The volatility connectedness of the eu carbon market with commodity and financial markets in time- and frequency-domain: the role of the u.s. economic policy uncertainty, Resources Policy 74 (3) (2021) 102252.

[26] B. Kocaarslan, U. Soytas, Reserve currency and the volatility of clean energy stocks: the role of uncertainty, Energy Economics 104 (2021) 105645.

[27] R. Ahmed, Global commodity prices and macroeconomic fluctuations in a low interest rate environment, Energy Economics 127 (2023) 1–13.

[28] E. Yeny, M. Rodríguez, A. Pérez-Uribe, J. Contreras, Wind put barrier options pricing based on the nordix index, Energies 14 (4) (2021) 1171–1186.

[29] A. Adegboye, M. Kampouridis, F. Otero, Improving trend reversal estimation in forex markets under a directional changes paradigm with classification algorithms, International Journal of Intelligent Systems 36 (12) (2021) 7609–7640.