# State-of-the-Art AI Methods for Pricing a Generic Weighted Portfolio of Cash, Currencies and Commodities

Soumadeep Ghosh

Kolkata, India

**Abstract**

This paper surveys contemporary artificial intelligence and machine learning methodologies for pricing portfolios comprising cash holdings, foreign exchange instruments, and commodity assets. We examine neural network architectures, reinforcement learning frameworks, and hybrid approaches that address the unique challenges of multi-asset portfolio valuation. The integration of deep learning with traditional quantitative finance provides enhanced predictive accuracy while managing computational complexity and model interpretability.

The paper ends with “The End”

## 1 Introduction

Portfolio valuation in multi-asset contexts presents significant computational and theoretical challenges. A generic weighted portfolio $P$ can be expressed as:

$$P(t) = \sum_{i=1}^{n} w_i \cdot A_i(t) \tag{1}$$

where $w_i$ represents the weight of asset $i$, and $A_i(t)$ denotes the value of asset $i$ at time $t$. For portfolios spanning cash, currencies, and commodities, traditional pricing models face limitations due to:

- Non-linear dependencies between asset classes
- Time-varying volatility and correlation structures
- Market regime changes and structural breaks
- High-dimensional feature spaces

Modern AI methods address these challenges through data-driven learning paradigms that adapt to market dynamics without restrictive parametric assumptions.

## 2 Taxonomy of AI Methods

### 2.1 Neural Network Architectures

#### 2.1.1 Feedforward Networks

Deep feedforward networks (DFNs) approximate the pricing function $f : \mathbb{R}^d \to \mathbb{R}$ through compositional transformations:

$$f(x) = f_L \circ f_{L-1} \circ \cdots \circ f_1(x) \tag{2}$$

where each layer $f_\ell(x) = \sigma(W_\ell x + b_\ell)$ applies an affine transformation followed by a non-linear activation $\sigma$.

### 2.1.2 Recurrent Neural Networks

For time-series modeling, Long Short-Term Memory (LSTM) networks capture temporal dependencies:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{3}$$
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{4}$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{5}$$
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{6}$$

### 2.1.3 Transformer Architectures

Attention mechanisms enable modeling of long-range dependencies:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V \tag{7}$$

Transformers excel at capturing cross-asset relationships in portfolio contexts.

## 2.2 Reinforcement Learning Frameworks

Portfolio management can be formulated as a Markov Decision Process (MDP) $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$ where:

- $\mathcal{S}$: State space (market conditions, portfolio positions)
- $\mathcal{A}$: Action space (rebalancing decisions)
- $\mathcal{P}$: Transition probabilities
- $\mathcal{R}$: Reward function (portfolio returns)
- $\gamma$: Discount factor

### 2.2.1 Deep Q-Networks (DQN)

DQN approximates the action-value function:

$$Q(s, a; \theta) \approx Q^*(s, a) \tag{8}$$

optimized through temporal difference learning:

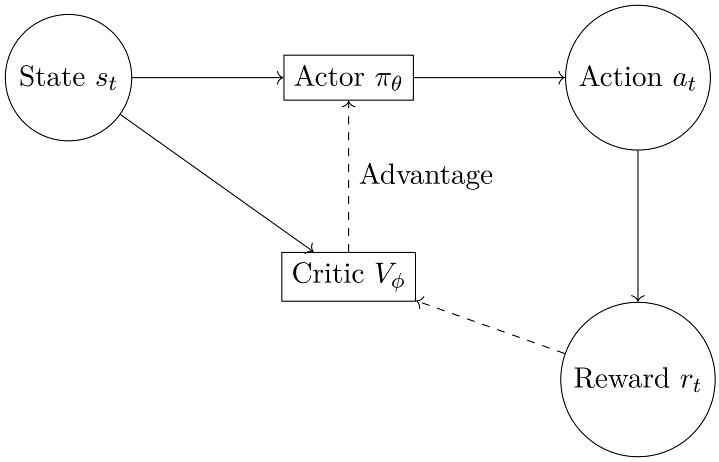
$$\mathcal{L}(\theta) = \mathbb{E}\left[\left(r + \gamma \max_{a'} Q(s', a'; \theta^-) - Q(s, a; \theta)\right)^2\right] \tag{9}$$

### 2.2.2 Policy Gradient Methods

Actor-critic algorithms optimize policies directly:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(a|s) A^{\pi_\theta}(s,a)\right] \tag{10}$$

where $A^{\pi_\theta}(s,a)$ is the advantage function.




# 3 Asset-Specific Considerations

## 3.1 Currency Pairs

Foreign exchange modeling requires consideration of:

- Interest rate differentials
- Carry trade dynamics
- Central bank policy expectations

Convolutional neural networks (CNNs) can extract patterns from FX orderbook data, while graph neural networks (GNNs) model currency network effects.

## 3.2 Commodity Markets

Commodity prices exhibit unique characteristics:

$$S_t = F_{t,T} - \text{Convenience Yield}_t + \text{Storage Cost}_t \tag{11}$$

AI methods must incorporate:

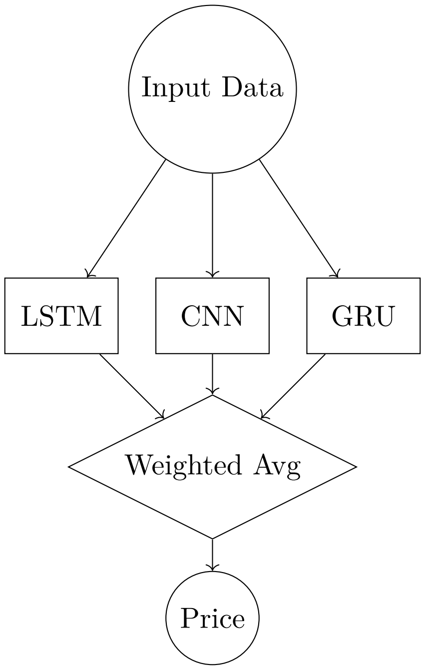
- Seasonality patterns
- Supply chain disruptions
- Weather and geopolitical factors

### 3.3 Ensemble Methods

Combining multiple models enhances robustness:

$$\hat{P}(t) = \sum_{k=1}^{K} \alpha_k M_k(t) \tag{12}$$

where $M_k$ represents individual models and $\alpha_k$ are learned weights satisfying $\sum_k \alpha_k = 1$.




## 4 Training Methodologies

### 4.1 Loss Functions

Portfolio pricing models optimize domain-specific objectives:

- Mean Squared Error: $\mathcal{L}_{MSE} = \frac{1}{N}\sum_{i=1}^{N}(\hat{P}_i - P_i)^2$
- Sharpe Ratio Maximization: $\mathcal{L}_{SR} = -\frac{\mathbb{E}[R_p - R_f]}{\sigma(R_p)}$
- Value-at-Risk: $\mathcal{L}_{VaR} = \mathbb{P}(P(t) < \text{VaR}_\alpha)$

### 4.2 Regularization Techniques

Preventing overfitting in financial time series:

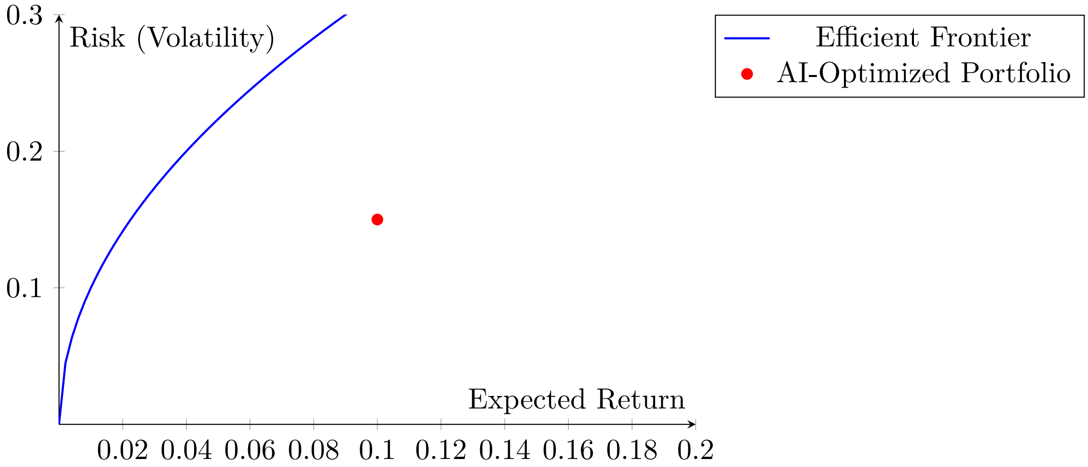
$$\mathcal{L}_{total} = \mathcal{L}_{pred} + \lambda_1\|\theta\|_2^2 + \lambda_2\|\theta\|_1 \tag{13}$$

Dropout, batch normalization, and early stopping provide additional regularization.

# 5 Risk Management Integration

AI pricing models must incorporate risk constraints:

$$\begin{aligned} \text{maximize} \quad & \mathbb{E}[R_p] && (14) \\ \text{subject to} \quad & \text{VaR}_\alpha(P) \leq \kappa && (15) \\ & \sum_{i=1}^{n} w_i = 1 && (16) \\ & w_i \geq 0 \quad \forall i && (17) \end{aligned}$$




# 6 Practical Implementation Considerations

## 6.1 Data Requirements

High-quality training data encompasses:

- Historical price time series (tick, minute, daily frequencies)
- Macroeconomic indicators (GDP, inflation, employment)
- Market microstructure data (volume, bid-ask spreads)
- Alternative data sources (sentiment, satellite imagery)

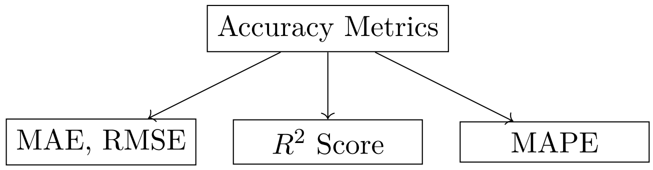
## 6.2 Computational Infrastructure

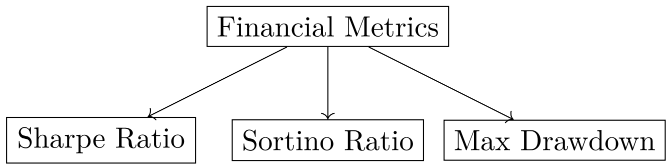
Modern AI pricing systems require:

- GPU acceleration for neural network training
- Distributed computing for hyperparameter optimization
- Low-latency inference pipelines for real-time pricing
- Robust backtesting frameworks

# 7 Evaluation Metrics

Model performance assessment uses multiple criteria:




# 8 Challenges and Future Directions

## 8.1 Interpretability

Black-box AI models face regulatory scrutiny. Approaches include:

- SHAP (SHapley Additive exPlanations) values
- LIME (Local Interpretable Model-agnostic Explanations)
- Attention weight visualization

## 8.2 Adversarial Robustness

Portfolio pricing models must resist adversarial perturbations:

$$\max_{\|\delta\|\leq\epsilon} \mathcal{L}(f(x+\delta), y) \tag{18}$$

## 8.3 Transfer Learning

Pre-trained financial models enable knowledge transfer across:

- Different asset classes
- Geographic markets
- Time periods

## 8.4 Quantum Machine Learning

Emerging quantum algorithms promise exponential speedups for portfolio optimization:

$$|\psi\rangle = \sum_{i=1}^{2^n} \alpha_i |i\rangle \tag{19}$$

where quantum superposition encodes portfolio configurations.

## 9 Conclusion

State-of-the-art AI methods have transformed portfolio pricing across cash, currency, and commodity markets. The integration of deep learning architectures with reinforcement learning frameworks provides unprecedented flexibility and accuracy. Future developments in interpretable AI, adversarial robustness, and quantum computing will further enhance these capabilities while addressing regulatory and practical constraints.

The optimal approach combines multiple AI techniques in an ensemble framework, leveraging the strengths of different architectures while maintaining computational efficiency and interpretability. As financial markets evolve, continuous model adaptation and retraining remain essential for sustained performance.

## References


[1] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

[2] Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: An Introduction* (2nd ed.). MIT Press.

[3] Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2), 654–669.

[4] Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

[5] Deng, Y., Bao, F., Kong, Y., Ren, Z., & Dai, Q. (2017). Deep direct reinforcement learning for financial signal representation and trading. *IEEE Transactions on Neural Networks and Learning Systems*, 28(3), 653–664.

[6] Jiang, Z., Xu, D., & Liang, J. (2017). A deep reinforcement learning framework for the financial portfolio management problem. *arXiv preprint arXiv:1706.10059*.

[7] Heaton, J. B., Polson, N. G., & Witte, J. H. (2017). Deep learning for finance: deep portfolios. *Applied Stochastic Models in Business and Industry*, 33(1), 3–12.

[8] Krauss, C., Do, X. A., & Huck, N. (2017). Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. *European Journal of Operational Research*, 259(2), 689–702.

[9] Bao, W., Yue, J., & Rao, Y. (2017). A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PloS one*, 12(7), e0180944.

[10] Moody, J., & Saffell, M. (2001). Learning to trade via direct reinforcement. *IEEE Transactions on Neural Networks*, 12(4), 875–889.

[11] Chen, K., Zhou, Y., & Dai, F. (2015). A LSTM-based method for stock returns prediction: A case study of China stock market. *IEEE International Conference on Big Data*, 2823–2824.

[12] Sezer, O. B., Gudelek, M. U., & Ozbayoglu, A. M. (2020). Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied Soft Computing*, 90, 106181.

[13] Lipton, Z. C., Berkowitz, J., & Elkan, C. (2015). A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*.

[14] Dixon, M., Halperin, I., & Bilokon, P. (2020). *Machine Learning in Finance: From Theory to Practice*. Springer.

[15] Lopez de Prado, M. (2018). *Advances in Financial Machine Learning*. Wiley.

## Glossary

**Actor-Critic** A reinforcement learning architecture combining policy-based (actor) and value-based (critic) methods for decision-making optimization.

**Attention Mechanism** A neural network component that weighs the importance of different input features dynamically, enabling focus on relevant information.

**Convolutional Neural Network (CNN)** A deep learning architecture using convolution operations to extract hierarchical features, originally designed for image processing but applicable to time-series data.

**Deep Q-Network (DQN)** A reinforcement learning algorithm combining Q-learning with deep neural networks for approximating optimal action-value functions.

**Ensemble Learning** The technique of combining multiple machine learning models to improve prediction accuracy and robustness.

**Feature Engineering** The process of creating informative input variables from raw data to improve model performance.

**Graph Neural Network (GNN)** A neural network architecture designed to process data structured as graphs, capturing relationships between entities.

**Hyperparameter Optimization** The process of tuning model configuration parameters (learning rate, architecture depth, etc.) to maximize performance.

**Long Short-Term Memory (LSTM)** A recurrent neural network architecture with gating mechanisms to capture long-range temporal dependencies while mitigating vanishing gradients.

**Markov Decision Process (MDP)** A mathematical framework for modeling sequential decision-making under uncertainty, fundamental to reinforcement learning.

**Mean Absolute Error (MAE)** A loss function measuring average absolute differences between predicted and actual values: $\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$.

**Overfitting** The phenomenon where a model learns training data patterns too specifically, reducing generalization to new data.

**Policy Gradient** A class of reinforcement learning algorithms that optimize decision policies directly through gradient ascent on expected returns.

**Regularization** Techniques to prevent overfitting by constraining model complexity, including L1/L2 penalties and dropout.

**Sharpe Ratio** A risk-adjusted performance metric: $\frac{\mathbb{E}[R_p - R_f]}{\sigma(R_p)}$, measuring excess return per unit volatility.

**SHAP Values** SHapley Additive exPlanations, a game-theoretic approach to explaining individual predictions by quantifying feature contributions.

**Temporal Difference Learning** A reinforcement learning approach that updates value estimates based on differences between successive predictions.

**Transformer** A neural network architecture based entirely on attention mechanisms, achieving state-of-the-art results in sequence modeling.

**Transfer Learning** The practice of applying knowledge learned from one domain/task to improve performance on related domains/tasks.

**Value-at-Risk (VaR)** A risk metric estimating the maximum potential loss over a time horizon at a given confidence level.

# The End