AlphaSeek FinRL: A Hybrid Deep Learning Architecture for High-Frequency Cryptocurrency Trading

1st Jun-Chi Liu
School of Business and
Research Center for Econophysics
East China University of
Science and Technology
Shanghai, China
2089426079@qq.com

2nd Jun-Chao Ma
School of Business and
Research Center for Econophysics
East China University of
Science and Technology
Shanghai, China
jcma@ecust.edu.cn

3rd Zhi-Qiang Jiang
School of Business and
Research Center for Econophysics
East China University of
Science and Technology
Shanghai, China
zqjiang@ecust.edu.cn

***Abstract*—This paper presents a novel approach to cryptocurrency trading by introducing a hybrid deep learning architecture that combines state-of-the-art sequence modeling techniques with reinforcement learning. Our model integrates Mamba State Space Models (SSM), Temporal Convolution Networks (TCN), and multi-head attention mechanisms to capture complex temporal dependencies in market data, while leveraging Deep Q-Network variants for optimal decision making. We implement a sophisticated signal processing pipeline with adaptive smoothing and feature fusion mechanisms, followed by a reinforcement learning framework for trading strategy optimization. The proposed architecture demonstrates superior performance in capturing market dynamics and generating robust trading signals, as validated through comprehensive backtesting on high-frequency cryptocurrency data.**

***Index Terms*—Cryptocurrency trading, Mamba state space models, temporal convolution networks, attention mechanisms, deep reinforcement learning.**

I. Introduction

The cryptocurrency market presents unique challenges for algorithmic trading due to its high volatility, complex microstructure, and non-linear dynamics. Traditional approaches using recurrent neural networks (RNNs) and long short-term memory (LSTM) networks have shown limitations in capturing long-range dependencies and processing high-frequency data efficiently. This paper introduces a novel architecture that leverages recent advances in sequence modeling and reinforcement learning to address these challenges.

Our key contributions include:

- A hybrid architecture combining Mamba SSM, TCN, and attention mechanisms for robust feature extraction.
- An adaptive signal smoothing mechanism to reduce noise while preserving important market signals.
- A sophisticated feature fusion approach using gated mechanisms and residual connections.
- Integration of advanced DQN variants for reinforcement learning-based trading optimization.
- Comprehensive empirical validation using high-frequency cryptocurrency data.

II. Methodology

A. Neural Network Architecture

Our **MambaTCNRegNet** combines input preprocessing, TCN for local features, parallel Mamba SSMs for long-term dependencies, and multi-head attention for global context. A gated fusion with residuals and final smoothing ensures robust signal generation.

1) Input Processing and Feature Extraction: The initial processing layer employs batch normalization and layer normalization with GELU activation:

$$h_0 = \text{GELU}(\text{LayerNorm}(\text{BatchNorm}(x))) \tag{1}$$

This layer ensures that the input features are well-conditioned before being fed into subsequent modules.

2) Temporal Convolution Network: The TCN component implements dilated causal convolutions. The architecture comprises multiple stacked **TemporalBlock** modules that each consist of:

- **Dilated Causal Convolutions:** Two successive convolution layers with an exponentially increasing dilation factor 2^i to capture long-range temporal dependencies while preserving causality.
- **Weight Normalization and GELU Activation:** Each convolution is weight normalized and followed by a GELU activation to ensure smooth nonlinear transformations.
- **Residual Connections:** To maintain feature integrity across layers, residual connections are employed. When necessary, a 1×1 convolution is used to match the dimensionality.
- **Dropout Regularization:** Dropout layers are applied post-activation to mitigate overfitting.

The TCN processes input sequences by first permuting the dimensions to $[batch, channels, seq\_len]$ for convolution operations and subsequently restoring the original format $[seq\_len, batch, channels]$.

979-8-3315-9661-3/25/$31.00 ©2025 IEEE
DOI 10.1109/IDS66066.2025.00018

3) Mamba State Space Model: The Mamba State Space Model (SSM) is incorporated to capture long-term dependencies and selective dynamic patterns in high-frequency data. In our implementation, the input feature space is divided into four segments, each processed by an independent Mamba module. Each module is parameterized by model dimensions such as d_{model}, d_{state}, and convolutional parameters (e.g., d_{conv}, expansion factor, and adaptive time steps dt_{min} and dt_{max}). These modules operate as selective SSMs that learn to focus on critical state transitions in volatile market conditions. The outputs from the parallel Mamba modules are concatenated along the feature dimension, providing a comprehensive representation of the input dynamics with reduced computational overhead compared to full sequence attention mechanisms. In the context of cryptocurrency trading, this variant is sometimes referred to as *CryptoMamba* to emphasize its application for high-frequency market data.

4) Multi-head Attention Mechanism: Global dependencies in the input sequence are captured via a scaled dot-product attention mechanism:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2}$$

This layer refines the combined features by focusing on the most salient information across the entire sequence.

5) Feature Fusion and Residual Connections: The outputs from the TCN and Mamba modules are concatenated and passed through a gated fusion mechanism:

$$g_t = \sigma(W_g[h_t^{\text{TCN}}; h_t^{\text{Mamba}}]) \tag{3}$$

$$f_t = g_t \odot [h_t^{\text{TCN}}; h_t^{\text{Mamba}}] \tag{4}$$

The fused gated and attention features are passed through a feed-forward network with residual connections and layer normalization, effectively combining local and global information.

6) Adaptive Signal Smoothing: To further mitigate noise in the generated trading signals, we apply a learnable moving average filter:

$$y_t = \frac{1}{k}\sum_{i=0}^{k-1} x_{t-i} \tag{5}$$

where k is dynamically adjusted based on market volatility, ensuring that significant signal characteristics are preserved while transient noise is reduced.

B. Reinforcement Learning Integration

The feature vectors produced by MambaTCNRegNet serve as inputs to our reinforcement learning framework, modeled as a Markov Decision Process (S, A, P, R, γ):

- **State Space** S: Feature vectors processed by MambaTCNRegNet, $s_t \in \mathbb{R}^{10}$.
- **Action Space** A: $\{-1, 0, 1\}$ corresponding to {Sell, Hold, Buy}.
- **Reward Function** R: $R(s_t, a_t, s_{t+1}) = a_t \cdot (p_{t+1} - p_t) - c \cdot |a_t|$.
- **Discount Factor** γ: 0.996.

We integrate advanced DQN variants (Double DQN, Dueling DQN, and Twin D3QN) to optimize trading decisions while addressing challenges such as overestimation bias.

III. Experimental Results

A. Model Training Analysis

During the initial training phase of our MambaTCNRegNet, we monitored the convergence behavior through training and validation loss curves. Figure 1 presents the loss trajectories over 6000 training steps.

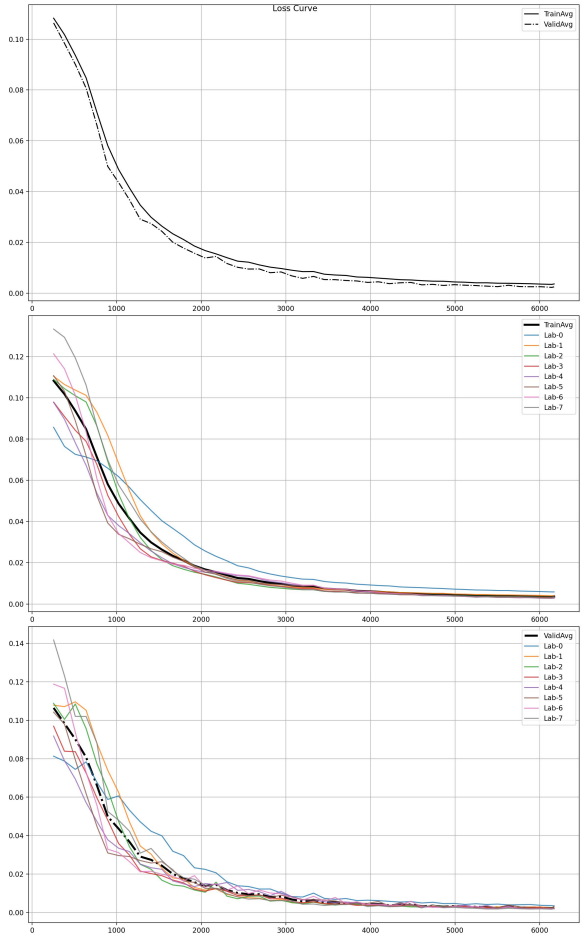

Fig. 1. The top panel compares training and validation loss, while the others show stable convergence across layers.

The loss curves demonstrate stable convergence characteristics, with both training and validation losses decreasing smoothly and plateauing after approximately 4000 steps. The middle and bottom panels reveal consistent learning patterns across different network layers, suggesting effective feature extraction and representation learning throughout the architecture.

B. Initial Signal-based Trading Results

Prior to integrating the reinforcement learning component, we conducted preliminary backtesting of our MambaTCN-RegNet on high-frequency cryptocurrency data using varying timeframes. The results indicate that the proposed architecture effectively captures market dynamics and generates robust trading signals.

1) 15-Minute Timeframe:

- Total Return: 29.38%
- Annual Return: 397.55% (projected)
- Maximum Drawdown: 1.69%
- Sharpe Ratio: 44.19
- Trade Count: 355
- Transaction Cost: 3.55%
- Win Rate: 72.13%

2) 30-Minute Timeframe:

- Total Return: 11.27%
- Annual Return: 254.80% (projected)
- Maximum Drawdown: 2.66%
- Sharpe Ratio: 22.45
- Trade Count: 184
- Transaction Cost: 1.84%
- Win Rate: 63.14%

Figure 2 presents the 30-minute strategy performance, including price and trade positions (top), individual returns (middle), and cumulative returns (bottom).

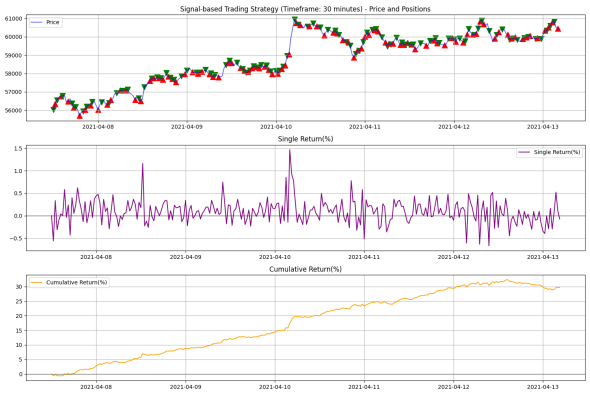

Fig. 2. Signal-based Trading Strategy Performance (30-minute timeframe). Top panel: price movement and trading positions; Middle panel: single-trade returns; Bottom panel: cumulative strategy returns.

3) Reinforcement Learning Progress: The integration of DQN-based reinforcement learning is currently under active development. Initial experiments with the Twin D3QN architecture have shown promising results in simulation environments, though further optimization is needed. Key areas of ongoing work include:

- Fine-tuning of reward function parameters to better balance risk and return.
- Implementation of advanced exploration strategies.
- Optimization of network architecture for real-time trading decisions.
- Integration of multi-timeframe signals into the state space.

Preliminary testing suggests potential for improved risk-adjusted returns compared to pure signal-based trading, particularly in volatile market conditions. Full results will be reported upon completion of the optimization phase.

IV. Annotations

The following annotations provide further context and references for the techniques utilized in our architecture:

- **Mamba State Space Model:** The Mamba implementation used in our work is based on the open-source repository `state-spaces/mamba` and is inspired by recent advances in structured state space models for sequence modeling. Notably, it builds upon the framework established by the S4 model [1] and related extensions.
- **Temporal Convolution Network (TCN):** Our TCN design follows principles outlined in [2], utilizing dilated causal convolutions with residual connections to effectively capture long-range dependencies while maintaining a strictly causal structure.
- **Multi-head Attention:** The attention mechanism implemented in our network adheres to the scaled dot-product attention mechanism originally proposed in [3] and is employed here to capture global dependencies within the sequence data.
- **Deep Q-Network Variants:** The reinforcement learning component leverages established techniques such as Double DQN [4], Dueling DQN [5], and Twin D3QN [6] to optimize trading decisions and mitigate common pitfalls like overestimation bias.
- **CryptoMamba:** In the context of cryptocurrency trading, the selective application of Mamba SSM modules has been colloquially termed "CryptoMamba" to highlight its targeted use in modeling high-frequency market dynamics.

V. Conclusion and Future Work

This paper presents a deep learning architecture that integrates Mamba SSM, TCN, and attention for high-frequency cryptocurrency trading. Results show effective signal generation. Future work includes refining the RL module and incorporating multi-timeframe modeling.

References

[1] A. Gu, et al., "Efficiently Modeling Long Sequences with Structured State Spaces," in *Proc. International Conference on Learning Representations (ICLR)*, 2021.

[2] S. Bai, J. Z. Kolter, and V. Koltun, "An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[3] A. Vaswani, et al., "Attention is All You Need," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[4] V. Mnih, et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, 2015.

[5] Z. Wang, et al., "Dueling Network Architectures for Deep Reinforcement Learning," in *Proc. International Conference on Machine Learning (ICML)*, 2016.

[6] S. Fujimoto, et al., "Addressing Function Approximation Error in Actor-Critic Methods," *arXiv preprint arXiv:1802.09477*, 2018.