**Abstract–** In this work, we develop a reinforcement learning (RL) trading system for the ETH/USDT market, leveraging years of historical hourly data obtained from Binance's API. The trading environment is designed to replicate real-world market conditions, with the agent making decisions to buy, sell, or hold based on a state representation derived from technical indicators like RSI and MACD. The core of the system is built on Proximal Policy Optimization (PPO), a reinforcement learning algorithm known for its reliability in optimizing complex tasks. To train the agent effectively, we preprocess ETH/USDT data to construct a sequence of meaningful features. The reward function is carefully designed to balance profitability and risk, penalizing unwise actions while rewarding favorable decisions. After training, the system was tested against the common Buy-and-Hold strategy. Results demonstrate the potential of PPO in producing a more adaptive and profitable trading framework. By integrating RL with high-frequency financial data, this study highlights a promising approach to cryptocurrency trading automation.

**Keywords–** Ethereum trading, Binance API, Reinforcement learning, Proximal Policy Optimization, Financial market simulation, Technical indicators

### **Introduction**

The cryptocurrency market has emerged as a dynamic and high-stakes domain for algorithmic trading, characterized by 24/7 operations, rapid price fluctuations, and a lack of regulatory constraints compared to traditional markets. Among the prominent assets, Ethereum (ETH) stands out due to its liquidity, use in decentralized finance (DeFi), and historical volatility, making it an attractive candidate for data-driven trading strategies. In this study, we leverage reinforcement learning (RL), a paradigm suited for sequential decision-making under uncertainty, to develop an adaptive trading agent for the ETH/USDT market.

The dataset for this study was sourced directly from Binance's API, providing a comprehensive record of hourly ETH/USDT price movements over several years. Preprocessing included feature engineering to derive market indicators like Relative Strength Index (RSI), Moving Average Convergence Divergence (MACD), and Bollinger Bands, which collectively represent momentum, trends, and volatility. These features form the agent’s state representation, enabling it to interpret market conditions effectively.

At the core of our approach is Proximal Policy Optimization (PPO), a state-of-the-art RL algorithm that uses a clipped objective function to ensure stable policy updates. The reward function was designed to align with trading objectives, rewarding actions like buying in oversold conditions or holding during neutral market phases. For example, the reward for holding when RSI lies between 30 and 70 is defined as:

By contrast, penalties are applied for actions misaligned with market conditions, such as selling in oversold states or buying in overbought states.

The trained agent demonstrates its capability to learn profitable and risk-aware trading strategies over time. By emphasizing the importance of feature engineering, reward customization, and rigorous evaluation, this research contributes to the advancement of reinforcement learning in financial trading systems.

### Formulas

### **1. Environment Definition**

**State (sts\_t):**

st={RSIt,MACDt,Pricet,Volumet,… }s\_t = \{ \text{RSI}\_t, \text{MACD}\_t, \text{Price}\_t, \text{Volume}\_t, \dots \}

**Action (ata\_t):**

at∈{0,1,2},where 0=Hold,1=Buy,2=Sell.a\_t \in \{ 0, 1, 2 \}, \quad \text{where } 0 = \text{Hold}, 1 = \text{Buy}, 2 = \text{Sell}.

**Reward (rtr\_t):** A scalar evaluating the quality of ata\_t in sts\_t.

**Transition Function (TT):**

st+1=T(st,at).s\_{t+1} = T(s\_t, a\_t).

### **2. Policy**

**Policy (πθ(at∣st)\pi\_\theta(a\_t \mid s\_t)):**

πθ(at∣st)=Softmax(fθ(st)),\pi\_\theta(a\_t \mid s\_t) = \text{Softmax}(f\_\theta(s\_t)),

where fθ(st)f\_\theta(s\_t) outputs logits for each action.

### **3. Value Function**

**Value Function (Vπ(st)V\_\pi(s\_t)):**

Vπ(st)=Eπ[∑k=0∞γkrt+k+1∣st],V\_\pi(s\_t) = \mathbb{E}\_\pi \left[ \sum\_{k=0}^\infty \gamma^k r\_{t+k+1} \mid s\_t \right],

where γ∈[0,1)\gamma \in [0, 1) is the discount factor.

### **4. Reward Function**

**(a) Buy Action (at=1a\_t = 1):**

rt={rbase+rRSI+rMACD,if RSIt<30 and MACDt>MACD Signalt,rbase+rRSI,if RSIt<30 and MACDt≤MACD Signalt,rpenalty,otherwise.r\_t = \begin{cases} r\_\text{base} + r\_\text{RSI} + r\_\text{MACD}, & \text{if } \text{RSI}\_t < 30 \text{ and } \text{MACD}\_t > \text{MACD Signal}\_t, \\ r\_\text{base} + r\_\text{RSI}, & \text{if } \text{RSI}\_t < 30 \text{ and } \text{MACD}\_t \leq \text{MACD Signal}\_t, \\ r\_\text{penalty}, & \text{otherwise}. \end{cases}

**(b) Sell Action (at=2a\_t = 2):**

rt={rbase+rRSI+rMACD,if RSIt>70 and MACDt<MACD Signalt,rbase+rRSI,if RSIt>70 and MACDt≥MACD Signalt,rpenalty,otherwise.r\_t = \begin{cases} r\_\text{base} + r\_\text{RSI} + r\_\text{MACD}, & \text{if } \text{RSI}\_t > 70 \text{ and } \text{MACD}\_t < \text{MACD Signal}\_t, \\ r\_\text{base} + r\_\text{RSI}, & \text{if } \text{RSI}\_t > 70 \text{ and } \text{MACD}\_t \geq \text{MACD Signal}\_t, \\ r\_\text{penalty}, & \text{otherwise}. \end{cases}

**(c) Hold Action (at=0a\_t = 0):**

rt={rneutral,if 30≤RSIt≤70,rpenalty,otherwise.r\_t = \begin{cases} r\_\text{neutral}, & \text{if } 30 \leq \text{RSI}\_t \leq 70, \\ r\_\text{penalty}, & \text{otherwise}. \end{cases}

**(d) Profit-Based Reward (Optional):**

rt=Net Wortht+1−Net WorthtNet Wortht.r\_t = \frac{\text{Net Worth}\_{t+1} - \text{Net Worth}\_t}{\text{Net Worth}\_t}.

### **5. Proximal Policy Optimization (PPO)**

**(a) Objective Function:**

L(θ)=Et[min⁡(rt(θ)A^t,clip(rt(θ),1−ϵ,1+ϵ)A^t)],L(\theta) = \mathbb{E}\_t \left[ \min \left( r\_t(\theta) \hat{A}\_t, \text{clip}(r\_t(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}\_t \right) \right],

where:

rt(θ)=πθ(at∣st)πold(at∣st).r\_t(\theta) = \frac{\pi\_\theta(a\_t \mid s\_t)}{\pi\_\text{old}(a\_t \mid s\_t)}.

**(b) Entropy Regularization:**

Lentropy(θ)=−β∑atπθ(at∣st)log⁡πθ(at∣st).L\_\text{entropy}(\theta) = -\beta \sum\_{a\_t} \pi\_\theta(a\_t \mid s\_t) \log \pi\_\theta(a\_t \mid s\_t).

**(c) Value Loss:**

Lvalue(θ)=Et[(Vπ(st)−V^t)2],L\_\text{value}(\theta) = \mathbb{E}\_t \left[ \left( V\_\pi(s\_t) - \hat{V}\_t \right)^2 \right],

where:

V^t=∑k=0∞γkrt+k+1.\hat{V}\_t = \sum\_{k=0}^\infty \gamma^k r\_{t+k+1}.

**(d) Total Loss:**

Ltotal(θ)=L(θ)+c1Lvalue(θ)−c2Lentropy(θ).L\_\text{total}(\theta) = L(\theta) + c\_1 L\_\text{value}(\theta) - c\_2 L\_\text{entropy}(\theta).

### **6. Performance Metrics**

**Net Worth:**

Net Wortht=Balancet+Crypto Ownedt×Pricet.\text{Net Worth}\_t = \text{Balance}\_t + \text{Crypto Owned}\_t \times \text{Price}\_t.

**Cumulative Reward:**

R=∑t=0Trt.R = \sum\_{t=0}^T r\_t.

**Sharpe Ratio:**

Sharpe Ratio=μRσR,\text{Sharpe Ratio} = \frac{\mu\_R}{\sigma\_R},

where μR\mu\_R and σR\sigma\_R are the mean and standard deviation of portfolio returns.