**Finlatics Project – AlgoTrading’24**

Name**:** Harsh Pandey Branch: PI Enrollment Number: 20119022 Email:harsh\_p2@me.iitr.ac.in

Problem Description

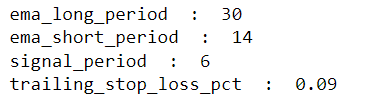
The task involves developing algorithmic trading strategies for the ETH/USDT cryptocurrency market with the objective of outperforming benchmark returns. Participants are required to create trading algorithms that can generate returns while effectively managing risk in the ETH/USDT market.

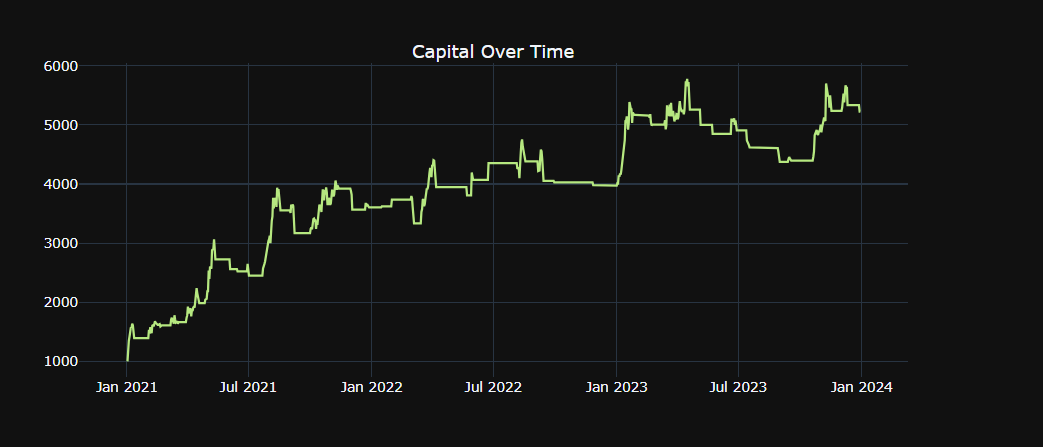
Strategy

This trading method merges the Moving Average Convergence Divergence (MACD) indicator with a trailing stop-loss system. The calculate\_macd function determines the MACD and signal lines using exponential moving averages of the closing stock prices. The apply\_trading\_strategy function creates entry and exit points based on when the MACD line overtakes the signal line. Entry (buy) points are set when the MACD line rises above the signal line, while exit (sell) points are set when it falls below. Reinforcement learning is employed to fine-tune the parameters for the moving averages and the trailing stop-loss, which makes the strategy more adaptive to changes in the market and boosts its effectiveness in trading.

Using reinforcement learning, we find optimal parameters via best\_parameters(), then fit the model for maximum profit and minimum risk, creating a robust and adaptive trading strategy.

The following are the parameters we find out work best for ETH/USDT market:



The equity curve we get after applying these parameters:

The Strategy\_Macd class is designed for trading analysis, encapsulating various methods for processing and evaluating trading data.

* **Initialization**: The \_\_init\_\_ method sets up the class with initial data, which is stored for use by other class methods.
* **MACD Calculation**: calculate\_macd calculates the MACD indicator by finding the difference between short-term and long-term EMAs of closing prices, and then derives the signal line from the MACD line’s EMA.
* **Trading Signals**: apply\_trading\_strategy produces entry and exit signals using MACD line crossovers with a trailing stop-loss to control losses, considering transaction costs and tracking the highest price to set the stop-loss level.
* **Optimization**: best\_parameters conducts a grid search to optimize the MACD strategy settings, testing various EMA periods, signal periods, and stop-loss percentages to maximize profits.
* **Performance Metrics**: calculate\_performance assesses the strategy’s success through metrics like total returns, profit, win rate, drawdowns, and risk-adjusted returns, and calculates gross profits and losses from trade outcomes.
* **Visualization**: plot\_trades uses Plotly to graphically represent trades, showing buy/sell signals, MACD and EMAs, and capital changes over time for strategy assessment.
* **Win/Loss Analysis**: calculate\_win\_loss\_ratio evaluates the strategy’s effectiveness by comparing the number of successful trades to unsuccessful ones, based on closing prices.
* **Comprehensive Analysis**: analysis applies the optimized trading strategy, visualizes trades, and provides a win/loss ratio for a detailed strategy review.
* **Trade Details**: print\_trades outputs the specific buy and sell dates for each trade, offering an in-depth look at trade timings.

This class structure provides a systematic approach to applying and analyzing a MACD-based trading strategy with a trailing stop-loss mechanism.

Strategy Metrics

