European Union
European Regional Development Fund
EPAnEK 2014–2020
OPERATIONAL PROGRAMME
COMPETITIVENESS
ENTREPRENEURSHIP
INNOVATION
ΕΣΠΑ 2014-2020
ανάπτυξη - εργασία - αλληλεγγύη
Partnership Agreement 2014 - 2020

SPEEDLAB

TRAINING INTELLIGENT PORTFOLIO MANAGEMENT AGENTS USING REINFOCEMENT LEARNING

Reinforcement Learning

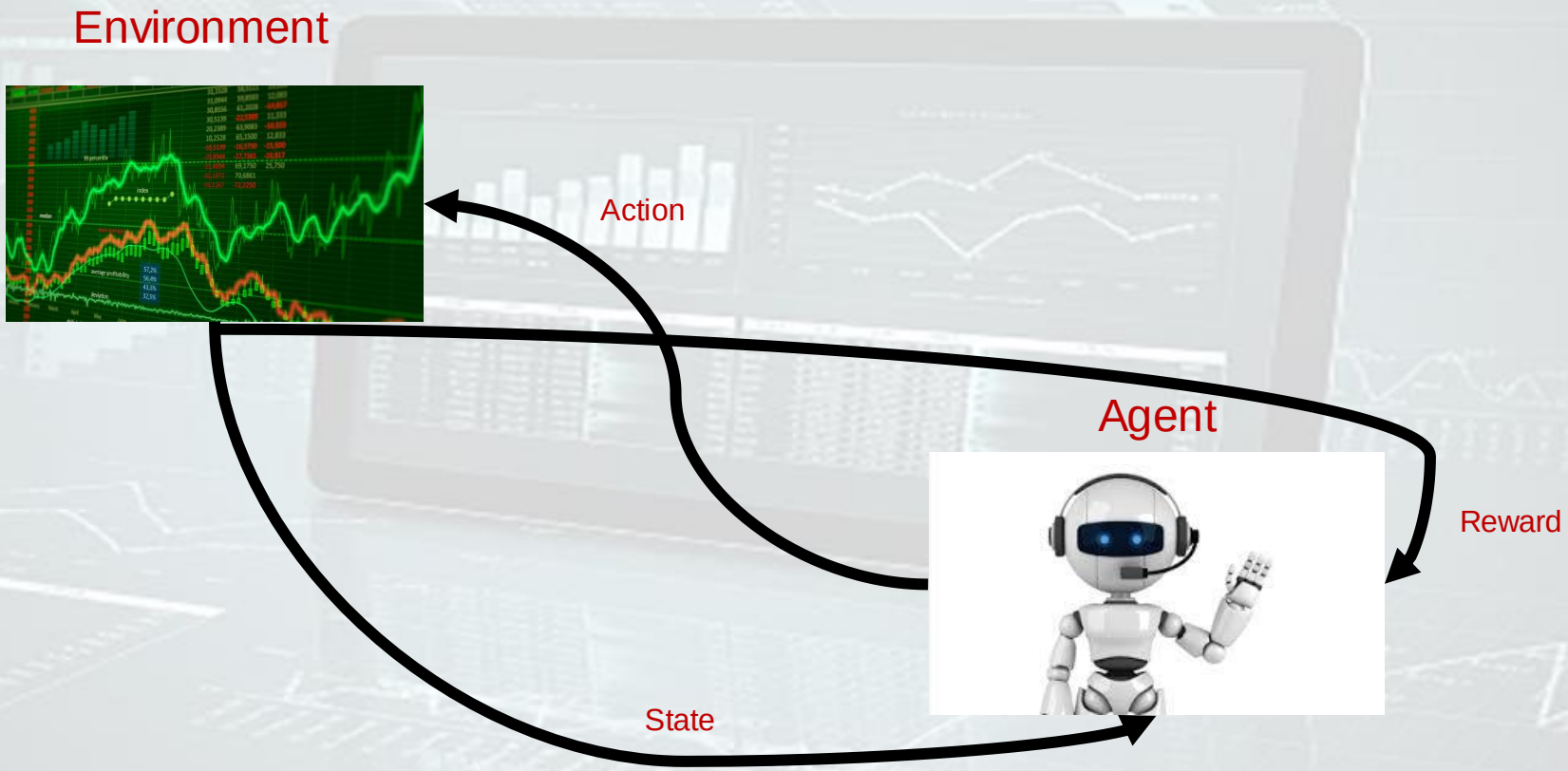
- Close to Human Learning and how human brain learns to act
- Algorithm learns a policy of how to act in a given environment
- Every action has some impact on the environment, and the environment provides rewards that guides the learning algorithm

Reinforcement Learning is learning how to act in order to maximize a numerical goal

Applications of RL

- Games (Atari Games, Chess, Go)
- Robotics
- Autonomous Driving
- Energy Conservation
- Marketing
- Healthcare
- Finance
- ChatGPT

Typical RL Scenario



Deep Reinforcement Learning Approach

- Implementation of Deep Q Learning
  - We try to train an algorithm for portfolio management by changing the position it holds on a financial asset
  - Inputs (State) are the raw price of an instrument (or transformations of the raw price, like moving averages, technical indicators, etc.)
  - Reward is the unrealized PnL of the next bar (Sharpe Ratio, Drawdown or other methods like Prado's triple barrier reward can be used)
  - The agent learns based on rewards to perform an action on each bar (Long, Short, Exit)
  - The agent also takes into account the previous action and that choosing a different action in each bar results in a penalty (commission cost)

Deep Learning Implementation

- Q Learning NN using MLP architecture
- Q Learning NN using RNN (LSTM) architecture

We have explored both Q Learning(Deep Q Network) and policy gradient (Advantage Actor Critic, Proximal Policy Optimization, Soft Actor Critic) algorithms.

Both type of algorithms are used for discrete action space, while the state space is continuous.

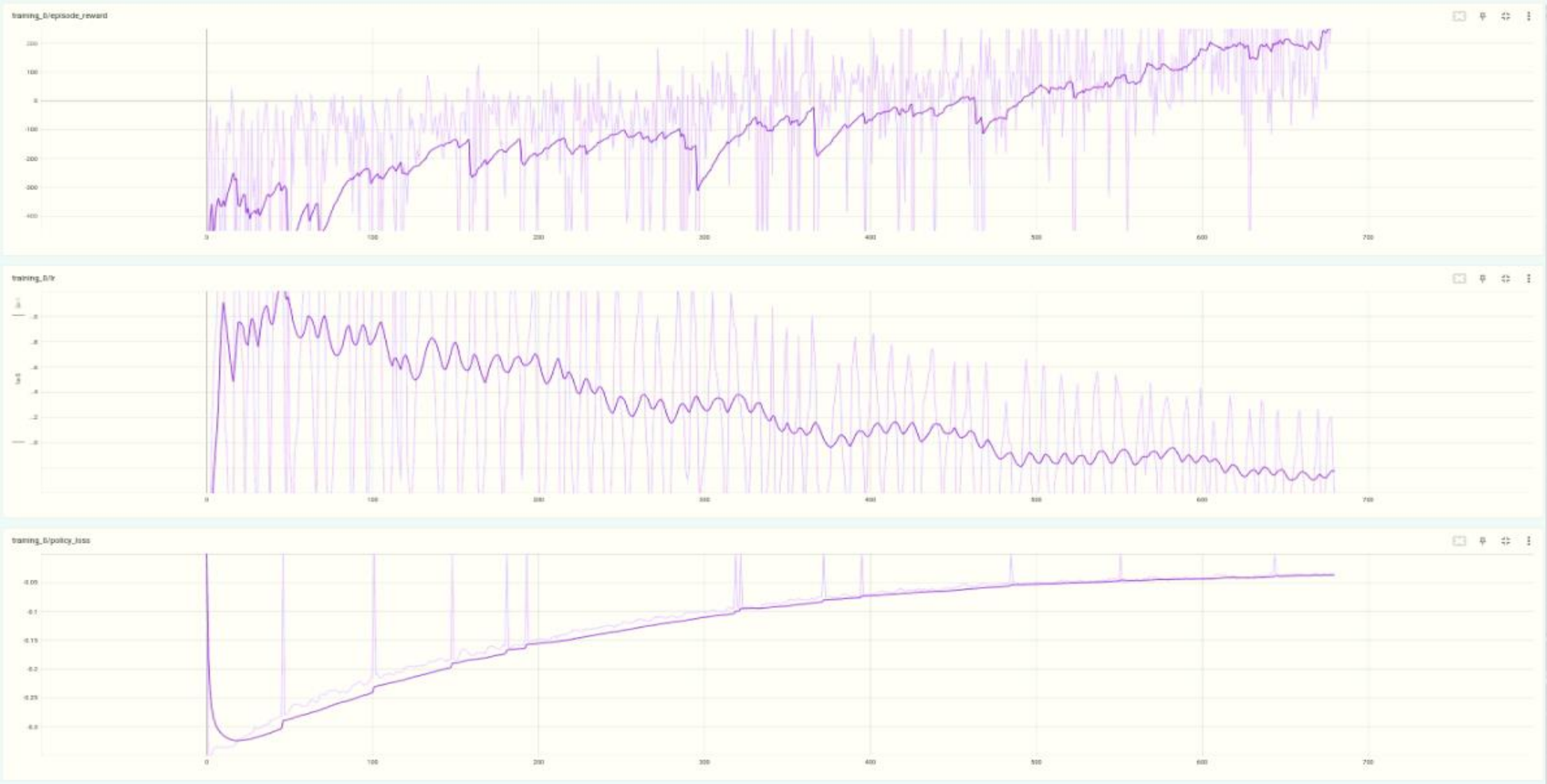
Environment Setup

1. Selection the financial instrument we want our agent to run.
2. Selection the timeframe we want our timeframe to run. For the timeframe we choose, there will be a discrete number of bars which will exist the steps of each episode the agent runs
3. Selection of the input features (state) and the output of the agent (actions). In our example, we have the following actions:
    - Long Position : The agent decides to have a long position for the next timestep
    - Short Position : The agent decides to have a short position for the next timestep
    - Exit : The agent decides to not have a position on the next timestep
4. Selection of the reward (PnL, Triple Barrier, Drawdown, Sharp Ratio)
5. Selection of the algorithm we would like to run (DQN, PPO, SAC)

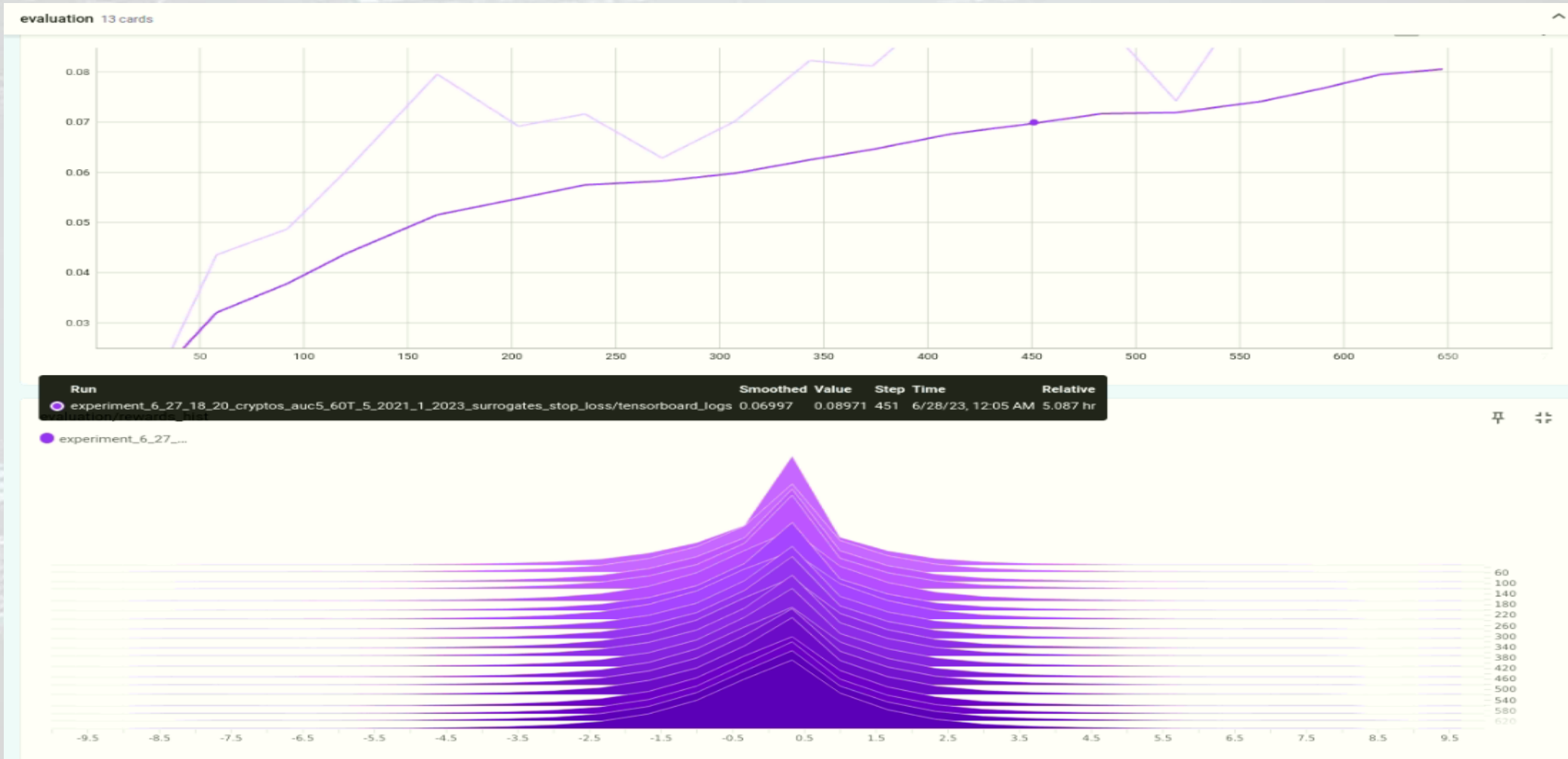
Environment Setup (2)

6. Split of our dataset into Train and Evaluation. Ideally train period would include different market situations. In the evaluation period we assess the portfolio management agent performance based on various metrics.
7. Save the trained model in order to be able to use it on live environment.

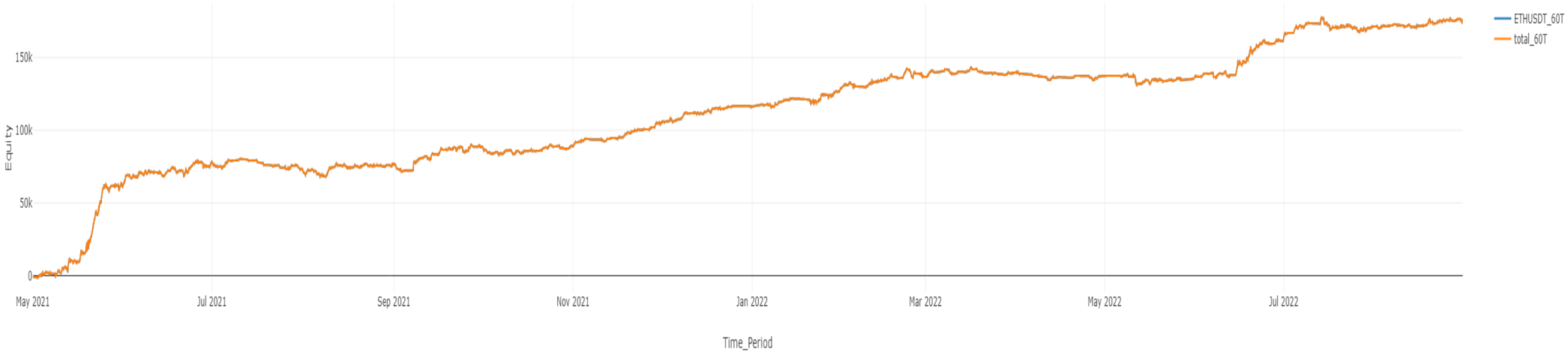
Training Charts



Training Charts (2)

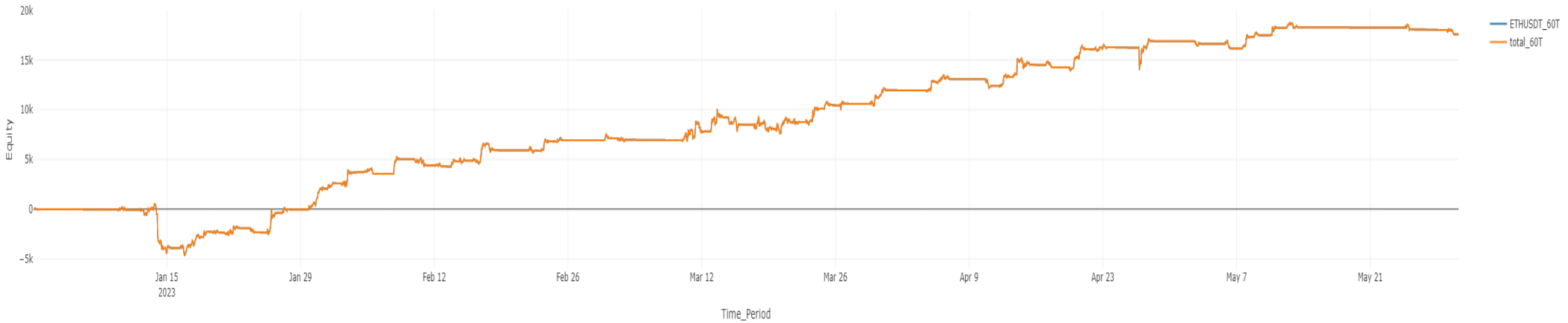


Trading Results (Train Period)



symbol	timeframe	trades	short_trades	long_trades	winning_trades	average_trade	std_trade	trade_ratio	long_percent	lot	short	losing_trades	hodl	spread	alpha	beta	gamma	drawdown	sharp_ratio	total_pnl	max_pnl	min_pnl	one_bar_percent	commision	mcc	winning_per
ETHUSDT	60T	877	439	438	565	224.92	478.61	0.38	0.33	1	0.31	312	-44548.61	0	-3.93	0.31	1878247.95	-12941.22	0.68	175193.92	177654.49	-1441.91	1	2	0.29	0.64

Trading Results (Evaluation Period)



symbol	timeframe	trades	short_trades	long_trades	winning_trades	average_trade	std_trade	trade_ratio	long_percent	lot	short	losing_trades	hodl	spread	alpha	beta	gamma	drawdown	sharp_ratio	total_pnl	max_pnl	min_pnl	one_bar_percent	commision	mcc	winning_perc
ETHUSDT	60T	877	439	438	565	224.92	478.61	0.38	0.33	1	0.31	312	-44548.81	0	-3.93	0.31	1878247.95	-12941.22	0.68	175193.92	177654.49	-1441.91	1	2	0.29	0.64

Goals of Reinforcement Learning

- Creation of different agents that could result in more diversified portfolios
- Using different features and rewards to achieve that
- Each agent will have a separate goal
- Ensemble learning different agents into a global one

Future Work

- Add sentiment into the features (part of this work). Features will have something different than just price transformations
- Add fundamental data into the features (mainly for stocks)
- Trade execution (more choices when performing an action, like more or less lot allocation, appropriate price to enter the trade – limit price)

Thank You

Questions?