Can Artificial Intelligence Trade the Stock Market?

Jedrzej Maskiewicz, Paweł Sakowski

Presentation plan

- Introduction
- Reinforcement Learning
- Research Methodology
- Results

What is algorithmic trading

- **Definition:** Automated execution of trades based on pre-defined rules or algorithms
- **Advantages**: Speed, precision, and elimination of emotional bias

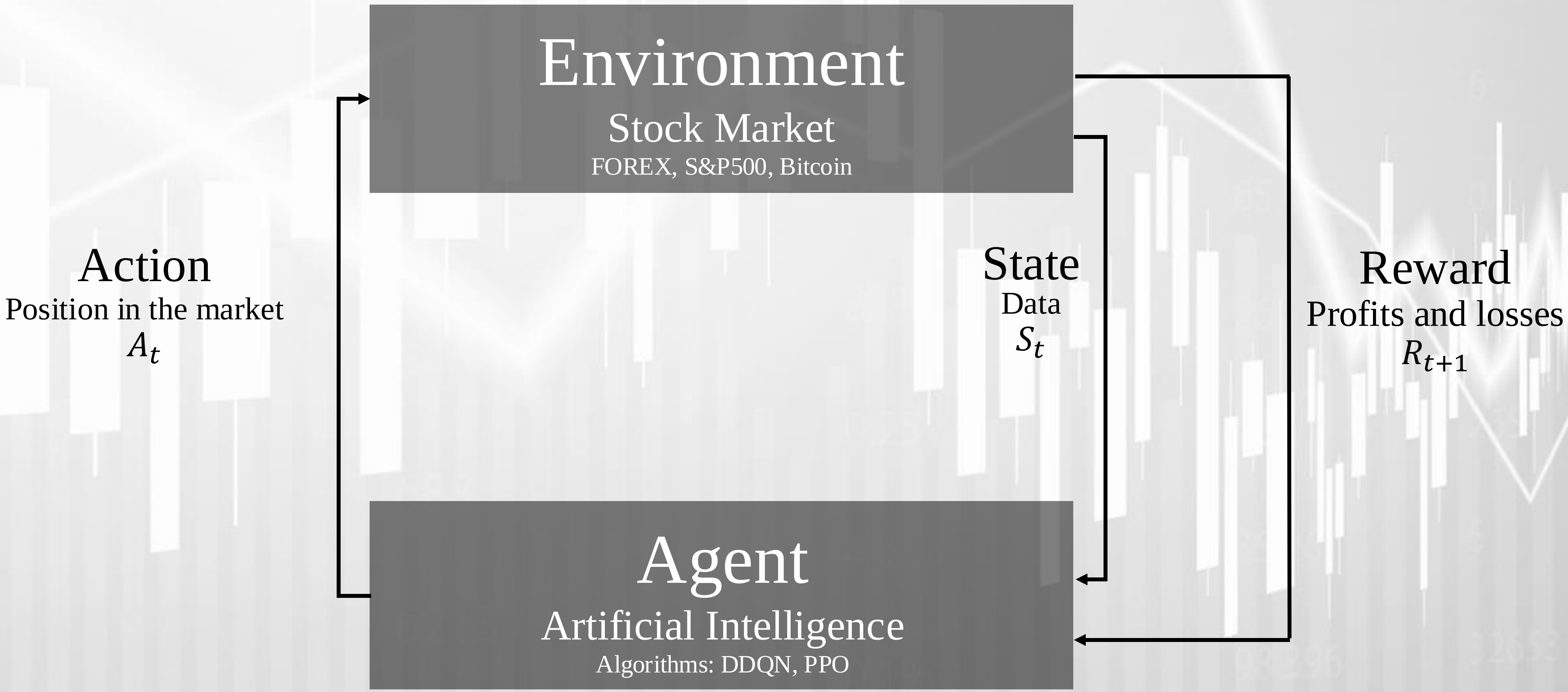
Introduction

- Context:
  - 70% of all equities traded in the US
  - 90% of FOREX market
- Key developments:
  - Rise of Deep Reinforcement Learning (DRL) in dynamic decision-making, across fields
  - Potential to surpass classical supervised models
- Focus
  - Test DRL algorithms (e.g., DDQN, PPO) on multiple assets
  - Benchmark against traditional “Buy and Hold”

What is Reinforcement Learning (RL)

Aspect	Supervised Learning	Unsupervised Learning	Reinforcement Learning
Objective	Learn from labelled data to map input to output	Discover hidden patterns in unlabelled data	Learn actions to maximize cumulative reward
Input Data	Labelled examples	Unlabelled examples	Interaction with an environment
Feedback	Direct feedback (correct/incorrect label)	No explicit feedback	Rewards (or penalties)
Examples	Image classification	Clustering	Self driving cars

Reinforcement Learning



- **Scenario:** Teaching a robot to reach a finish line.
  - **Goal:** Reward of 100 for reaching the finish line.
  - **Problem:** Crediting only the last step ignores the importance of earlier decisions

- **Why is This Wrong?**
  - Earlier actions also contribute to success.
  - Rewards should be distributed across all steps leading to the finish line.

- Solution: Use Temporal Difference (TD) to credit actions proportionally.

Temporal Difference (TD) Learning

$$V(s_t) = r_t + \gamma * r_{t+1} + \gamma^2 * r_{t+2} + \gamma^3 * r_{t+3} \ldots$$

$$V(s_t) \leftarrow V(s_t) + \alpha \, [r_{t+1} + \gamma \, V(s_{t+1}) - V(s_t)]$$

$V(s_t)$ – **Value Function**

- Represents the **expected total reward** an agent will accumulate starting from a state s and following a specific policy π
- It evaluates how "good" a state is in terms of future rewards

γ - **Discount factor** — determines the importance of future rewards

α - **Learning rate** — controls how much the value function is updated with new information

Policy vs Value based approach

- Value-Based Approach
  - Learns to estimate the value function $V(s)$ or Q(s, a)
  - Selects actions that have highest value

- Policy-Based Approach
  - Directly optimizes the policy $\pi(a|s)$, mapping states to actions
  - Learns the probabilities of actions without explicitly learning a value function

Double Deep Q Network

- Q-Learning

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma\,(\,max_{a'}\,Q(s_{t+1}, a') - Q(s_t, a_t)]$$

- Deep Q-Network (DQN)
  - Replaces the Q-table with a Deep Neural Network to approximate $Q(s_t, a_t)$
  - With $Q(s_t, a_t;\ \theta)$

Double Deep Q Network

- DQN suffers from **overestimation bias** because the same network is used for both selecting and evaluating actions

- **Action Selection:** Uses the primary network to choose the best action:
$$a' = \arg max_{a'} Q(s_{t+1}, a'; \theta)$$

- **Action Evaluation:** Uses the target network to evaluate the selected action:
$$Q(s_{t+1}, a'; \theta^{-})$$

Double Deep Q Network

DQN formula:

$$Q(s_t, a_t;\ \theta)\ \leftarrow\ Q(s_t, a_t;\ \theta)\ +\ \alpha[r_{t+1} + \gamma\ (\ max_{a'}\ Q(s_{t+1}, a';\ \theta^-)) - Q(s_t, a_t;\ \theta)]$$

DDQN formula:

$$Q(s_t, a_t;\ \theta) \leftarrow Q(s_t, a_t;\ \theta) + \alpha[r_{t+1} + \gamma Q(s_{t+1}, \arg max_{a'}\ Q(s_{t+1}, a'; \theta); \theta^-) - Q(s_t, a_t; \theta)]$$

$Q(s_t, a_t;\ \theta)$ – Value function; estimation of Q-value at state s_t with action a_t calculated by Neural Network θ

γ – discount factor

r_{t+1} - reward after action a_t

θ^- - target network

DDQN

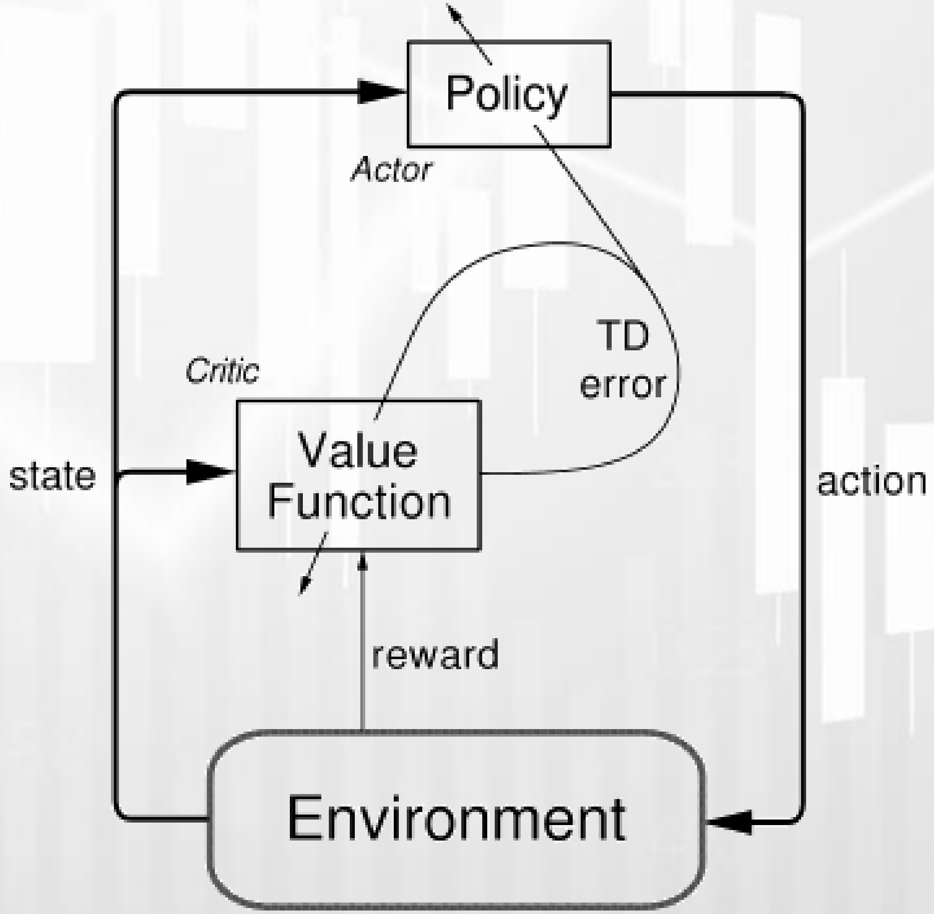
- Loss function:

$$L(\theta) = E[(r_{t+1} + \gamma \; max_{a'} Q(s', a'; \theta^-) - Q(s_t, a_t; \theta))^2]$$

Actor – Critic

- **Actor**:
  - Directly optimizes the policy $\pi(a|s;\theta)$, which maps states to actions
  - Determines the best action to take in a given state
- **Critic:**
  - Evaluates the Actor's actions by estimating the value function V(s;ω)
  - Provides feedback using Temporal Difference (TD) error or Advantage $A(s,a)$
  - $A(s, a) = Q(s, a) - V(s)$
- **Workflow:**
  - Actor proposes an action a
  - Critic evaluates a by estimating how good it is based on the current policy π
  - Actor updates its policy using the feedback from the Critics

Actor – Critic



Source: Sutton, R. S., & Barto, A. G. „Actor-Critic method”. Reinforcement Learning: An Introduction. (2018)

Proximal Policy Optimization (PPO)

- Actor – Critic family

- Very stable:
  - Ensures smooth policy updates by restricting large changes using the clipping mechanism
  - Reduces the risk of instability during training

- Clipping:

$$L^{CLIP}(\theta) = E[\min(R_t(\theta)\,, CLIP(R_t(\theta), 1-\ \varepsilon, 1+\ \varepsilon)\ )]$$

$R_t(\theta) = \dfrac{\pi(a|\ s_t;\theta)}{\pi_{OLD}(a|\ s_t;\theta)}$ - probability ratio of the action under current policy π to previous policy π_{OLD}

Methodology Overview

- Assets Analyzed:
  - EUR/USD and S&P 500 Index
- Daily data
- Features
- Reinvest profits after every trade
- Walk forward optimization

Walk forward optimization

Iteration\Years	2005-2016	2017	2018	2019	2020	2021	2022	2023
First optimisation	train		validation	test				
Second optimisation	train			validation	test			
Third optimisation	train				validation	test		
Fourth optimisation	train					validation	test	
Fifth final optimisation	train						validation	test

Evaluation Metrics

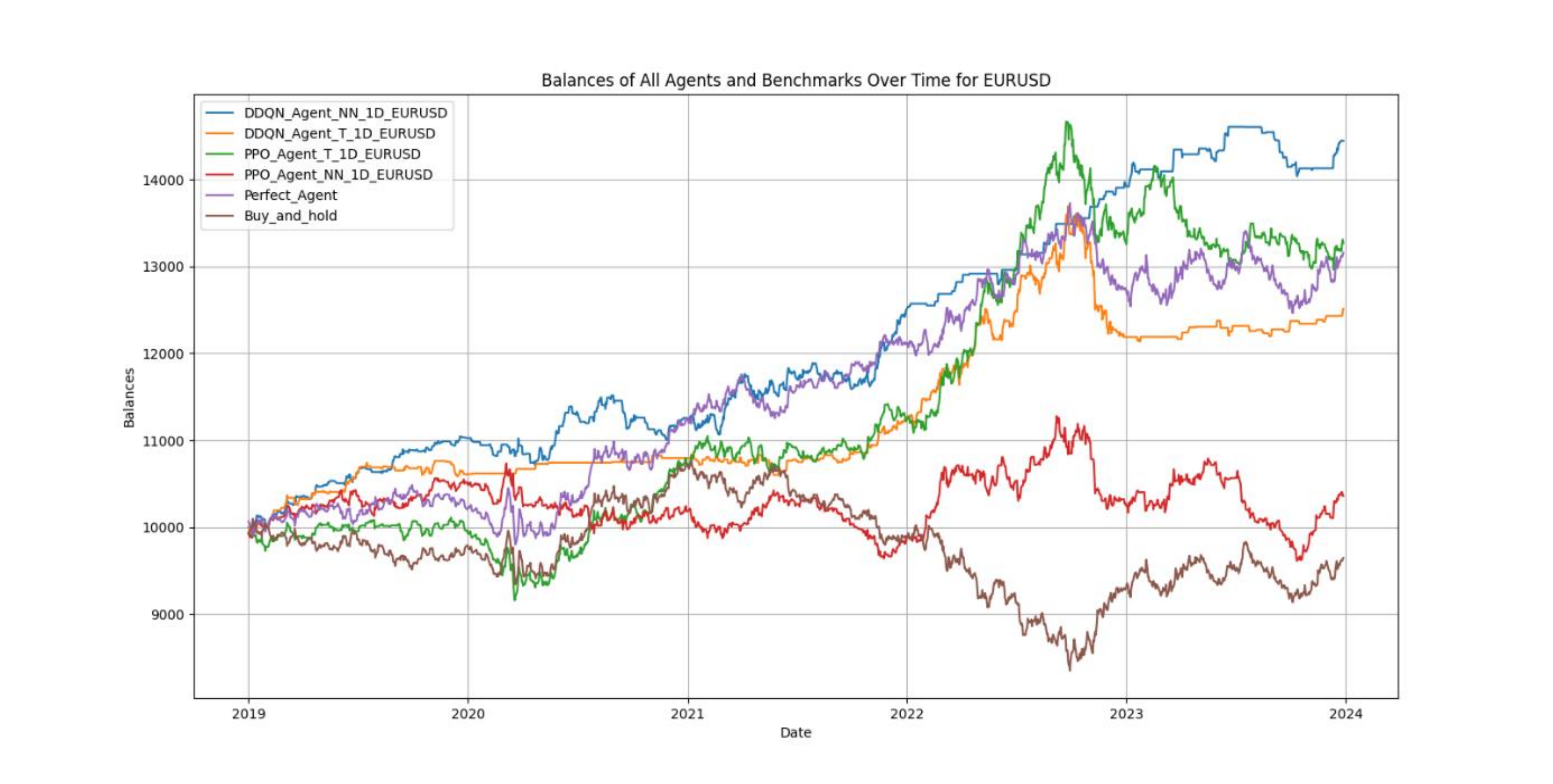
- Sharpe Ratio = $\frac{R}{\sigma} * \sqrt{N}$
  - R – return of the strategy
  - σ – standard deviation of strategy
  - N – annualization factor (number of intervals in the year)

- Benchmarks:
  - Buy and Hold
  - „Perfect annualized” Agent

Results EUR/USD

EURUSD	Final balance	CAGR	Sharpe ratio	Sortino Ratio	Maximum Drawdown	Win Rate	In Long	In Short	Out of the market
DDQN_NN	14 444.74	7.63%	1.842	2.043	-4.40%	58.6%	17.7%	26.2%	56%
DDQN_T	12 512.91	4.58%	1.035	0.853	-11.30%	55.4%	2.5%	30.7%	66.7%
PPO_NN	10 360.83	0.71%	0.108	0.136	-14.70%	50.3%	49.5%	30.1%	20.3%
PPO_T	13 271.26	5.82%	0.847	1.197	-11.60%	48.5%	40.4%	45.6%	13.9%
Benchmarks:									
Buy and hold	9 641.81	-0.84%	-0.108	-0.159	-22.50%	100%	100%	0%	0%
"perfect" annual strategy	13 150.82	6.61%	0.935	1.396	-9.20%	100%	40%	60%	0%

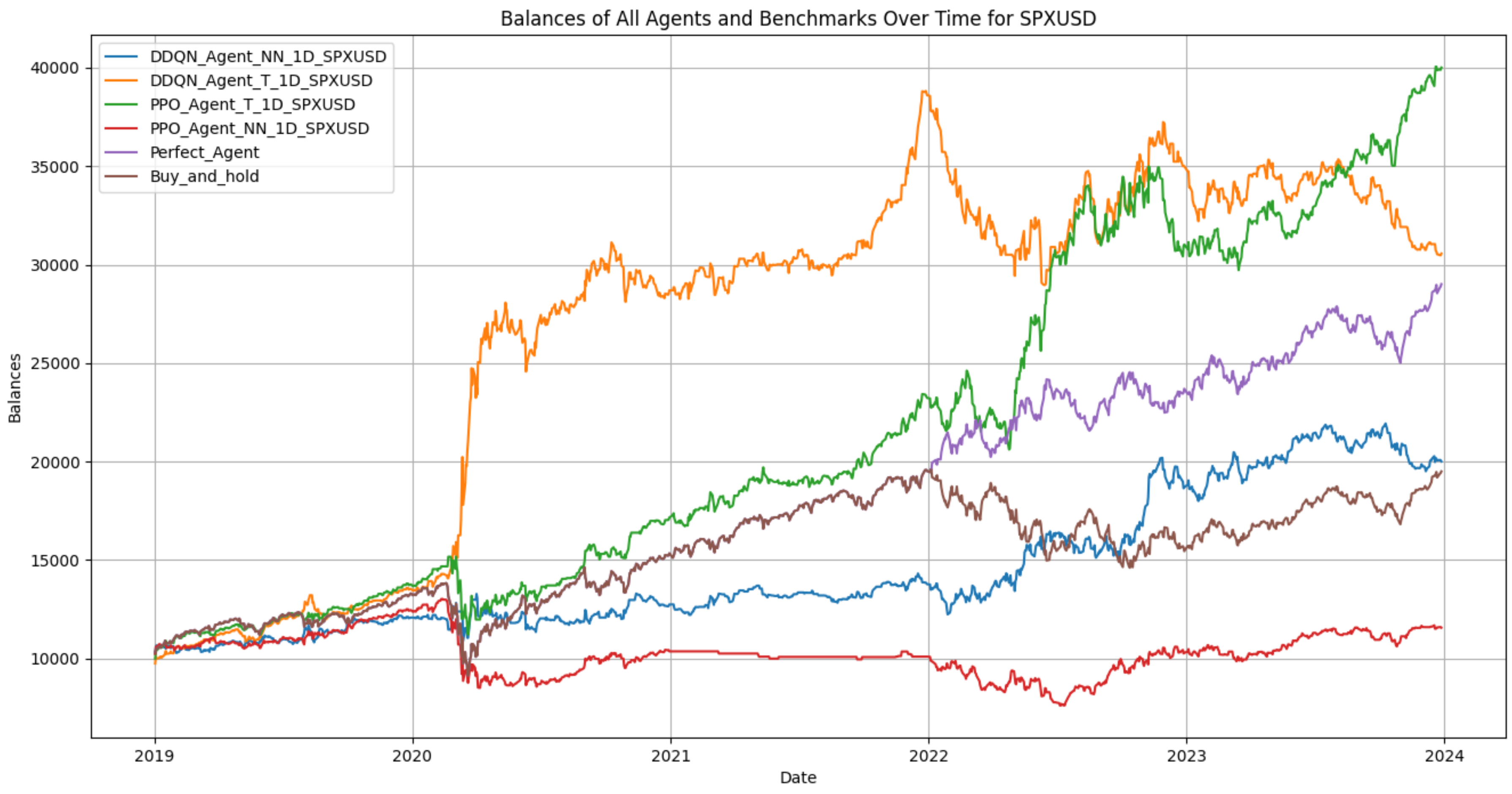
Results EUR/USD



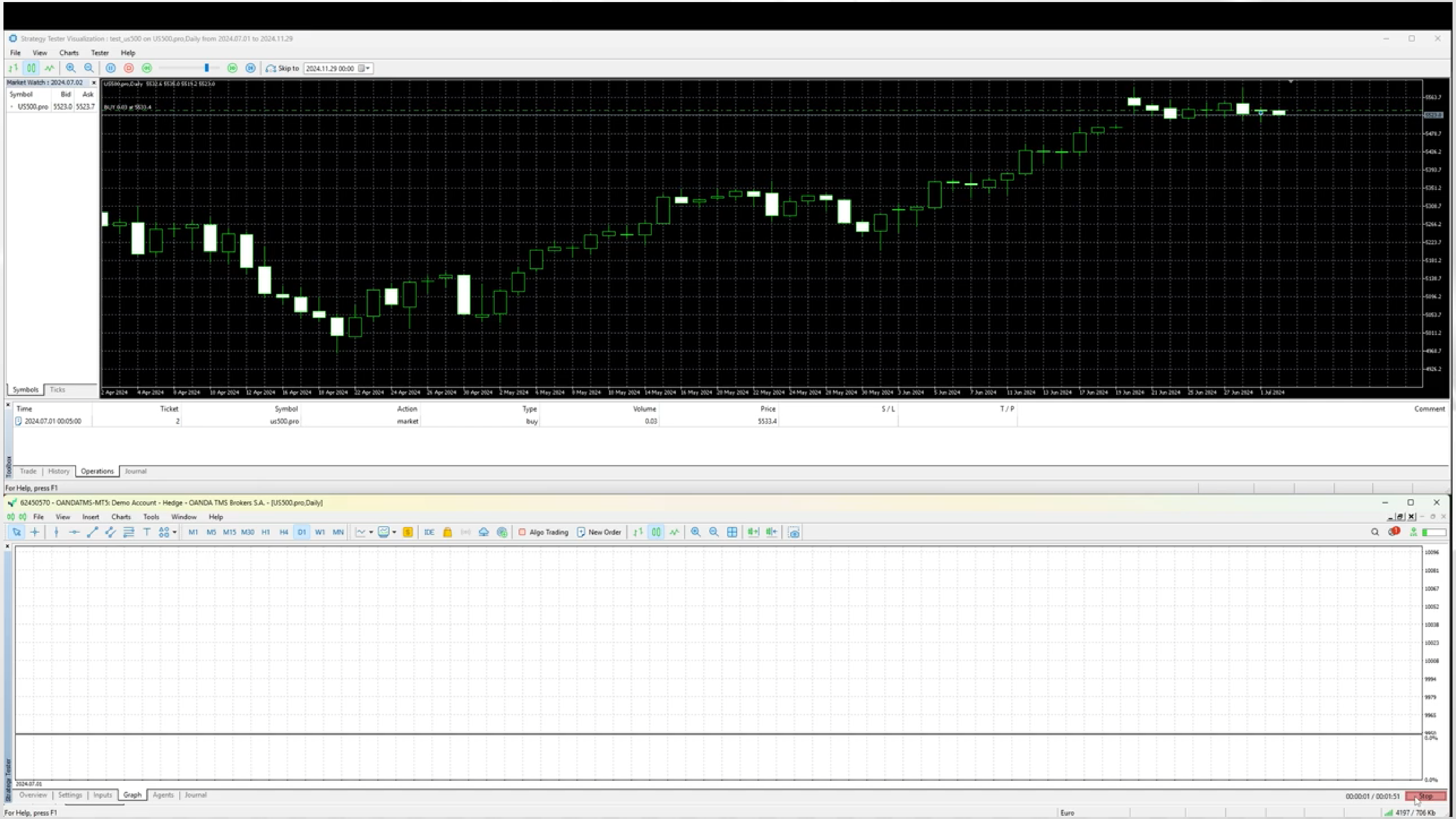
Results SP500

S&P 500	Final balance	CAGR	Sharpe ratio	Sortino Ratio	Maximum Drawdown	Win Rate	In Long	In Short	Out of the market
DDQN_NN	20 009.45	22.43%	0.993	1.225	-15%	51.3%	51.7%	33.5%	14.7%
DDQN_T	30 565.62	38.41%	1.6	2.218	-25.4%	51.7%	62.6%	22%	15.4%
PPO_NN	11 567.96	4.3%	0.183	0.193	-41.6%	54.7%	59.8%	13.4%	26.7%
PPO_T	40 010.83	49.76%	2.158	2.651	-26.1%	58.6%	73.9%	16.9%	9.1%
Benchmarks:									
Buy and hold	19 482.77	21.37%	0.834	1.014	-33.9%	100%	100%	0%	0%
"perfect" annual strategy	28 982.19	36.23%	1.505	1.791	-33.9%	100%	80%	20%	0%

Results SP500



Results SP500 – test sample from 1/7/2024



Methodological Challenges in Applying DRL to Stock Market Trading

- In financial markets, the agent's actions (e.g., buy, sell, out of the market) do not influence the broader market state

- TD algorithms attribute a discounted premium of future rewards to present actions
  - Example:

    In a sequence like *long*, *long*, *short*, *short*, future rewards from short positions are incorrectly attributed to earlier long positions

Recomendations

Sutton, R. S., & Barto, A. G. Reinforcement Learning: An Introduction (2018)

Thank you for your attention