FinRL Contest Task I Presents PPO-Switch: a Sparse Ensemble of Diversified PPO Trading Algorithms

Zhong Anyang*
Chao Kaiyin*
Chen Geying*
zhonganyang2021@email.szu.edu.cn
chaokaiyin2022@email.szu.edu.cn
chengeying2022@email.szu.edu.cn
Shenzhen University
China

Yin Jianfei*
Joshua Zhexue Huang
yjf@szu.edu.cn
zx.huang@szu.edu.cn
Shenzhen University
China

ABSTRACT

This article presents PPO-Switch, a novel sparse ensemble algorithm that combines multiple proximal policy optimization (PPO) trading experts with insights from online portfolio selection (OPS) algorithms. The PPO trading experts are trained using diverse and effective OPS price features, resulting in a diversified expert pool. PPO-Switch ensures sparsity in trading actions through two steps: (i) expert selection based on long-term and short-term returns, and (ii) optimizing cash to maximize capital efficiency. Experimental results on six real-world datasets demonstrate that PPO-Switch outperforms current state-of-the-art reinforcement learning (RL) and OPS trading algorithms when transaction fees exceed 0.1%.

CCS CONCEPTS

• **Applied computing** → **Multi-criterion optimization and decision-making**.

KEYWORDS

Stock trading, Reinforcement learning, Online portfolio selection, Sparse ensemble learning

ACM Reference Format:
Zhong Anyang, Chao Kaiyin, Chen Geying, Yin Jianfei, and Joshua Zhexue Huang. 2023. FinRL Contest Task I Presents PPO-Switch: a Sparse Ensemble of Diversified PPO Trading Algorithms. In *Proceedings of 4th ACM International Conference on AI in Finance (ICAIF-23) (ICAIF 2023).* ACM, New York, NY, USA, 6 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Maximizing returns in volatile financial markets is a classic challenge in computational finance [9, 15, 17]. Reinforcement learning (RL) algorithms [4, 14, 16], available through open-source RL trading platforms like FinRL [12, 13], provide a promising solution with a user-friendly environment for algorithm development and testing. However, existing RL-based trading algorithms often overlook valuable features offered by online portfolio selection (OPS) algorithms [5, 7, 8, 18], which encompass various mathematical models for optimal investment decisions. In this paper, we introduce PPO-Switch, a novel sparse ensemble algorithm that integrates multiple proximal policy optimization (PPO) [16] trading experts. Our main contributions are as follows: (i) Leveraging diversified OPS price features to train multiple PPO trading models, generating to a diverse expert pool; (ii) Employing a novel evaluation method to select the most suitable expert based on their long- and short-term returns; (iii) Allocating all remaining cash in the portfolio to the optimized asset for sparse asset selection.

2 DESIGN OF PPO-SWITCH ALGORITHM

This section outlines the key design aspects of the PPO-Switch algorithm.

2.1 RL Environment

We utilized the FinRL programming environment [13] for training and testing the experts and the ensemble algorithm. The environment API used can be found in Table 1.

Table 1: Methods of the `StockTradingEnv` environment class

id	Symbol	Explain	Method
1	m	number of stocks	`stock_dim`
2	p_i	closing price of ith stock	`state[1+i]`
3	b_i	shares of ith stock held	`state[1+m+i]`
4	ω_{cash}	cash	`state[0]`
5	ω_t	cumulative wealth	`asset_memory[t]`
6	*subenv*	sub-environment	`get_sb_env`

2.2 Construction of the Expert Pool

Price features play a crucial role in training trading experts [2, 6]. To create an expert pool with diverse trading behaviors, we construct four distinct types of price vectors from a set of original price vector samples $\{\boldsymbol{p}_{t-w+1}, ..., \boldsymbol{p}_t\}$ within a recent time window of size $w = 6$. These price-feature vectors are defined as follows:

$$\boldsymbol{p}_t^{max} = \max_{0\le k\le w-1} \boldsymbol{p}_{t-k}, \qquad \boldsymbol{p}_t^{min} = \min_{0\le k\le w-1} \boldsymbol{p}_{t-k},$$
$$\boldsymbol{p}_t^{mean} = \frac{1}{w}\sum_{k=0}^{w-1} \boldsymbol{p}_{t-k}, \qquad \boldsymbol{p}_t^{ema} = \sum_{k=0}^{w-1} \beta(1-\beta)^k \boldsymbol{p}_{t-k}. \tag{1}$$

*Equal author contributions; partially supported by Key Basic Research Foundation of Shenzhen (JCYJ20220818100205012).

ICAIF 2023, November 27-29, 2023, 4 MetroTech Center, Brooklyn, NY

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/XXXXXXX.XXXXXXX

Table 2: Algorithm performance comparison (The highest score in each column is highlighted in bold).

Algorithm	DOW			HS			CRYPTO			HK			NYSE			FTSE		
	CW	SR	MDD (%)	CW	SR	MDD (%)	CW	SR	MDD (%)	CW	SR	MDD (%)	CW	SR	MDD (%)	CW	SR	MDD (%)
BAH	1.18	0.44	20.96	1.30	0.31	47.32	0.99	-0.02	35.46	1.13	0.23	33.27	1.50	0.92	18.03	1.27	0.62	16.10
PPO-Switch	1.47	0.84	22.95	**3.03**	**1.07**	53.71	**1.33**	0.72	48.47	**2.16**	**1.20**	**20.09**	1.62	0.89	17.06	1.53	0.79	30.47
GWR	0.96	-0.05	46.83	0.61	-0.38	66.04	0.22	-2.04	82.49	0.68	-0.25	65.61	1.14	0.15	37.94	1.45	0.43	48.35
OLMAR	1.02	0.03	40.65	0.77	-0.20	57.34	0.28	-1.68	78.25	0.78	-0.16	62.37	1.01	0.01	41.12	1.72	0.61	43.44
PPT	1.22	0.25	35.23	1.23	0.15	47.34	0.30	-1.53	76.93	0.29	-0.77	81.03	1.17	0.18	44.16	1.14	0.15	48.28
SPOLC	**2.11**	**1.03**	27.25	0.78	-0.20	49.52	0.50	-0.92	65.29	1.30	0.19	60.25	1.30	0.32	32.86	**1.82**	0.74	39.62
A2C	1.30	0.76	19.58	0.98	0.11	46.81	1.07	0.38	**31.49**	0.46	-0.39	68.19	1.21	0.50	24.89	1.53	**0.98**	23.38
SAC	1.21	0.54	21.04	1.25	0.49	**33.36**	1.31	**0.75**	32.24	0.82	-0.24	44.00	**1.77**	**1.23**	16.85	0.94	-0.02	31.94
TD3	1.19	0.54	**18.32**	1.16	0.32	52.85	0.83	-0.14	35.13	0.60	-0.33	68.86	1.54	1.15	**11.27**	1.30	0.77	**11.63**
DDPG	1.21	0.55	19.77	0.97	0.14	56.60	1.24	0.63	32.79	0.52	-0.46	63.79	1.33	0.73	19.51	1.07	0.24	33.94

The expert pool, trained using the price-feature vectors defined in Eq. (1) and the original price vector $\boldsymbol{p}_t$, is denoted by the symbol $E = \{E_{max}, E_{min}, E_{mean}, E_{ema}, E_{real}\}$. All experts are trained using the following objective function:

$$\min L_p + c_e L_e + c_v L_v, \tag{2}$$

where the policy loss L_p [16], the entropy loss L_e, and the value loss L_v are defined as follows:

$$\begin{aligned} L_p &= \min\left\{r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta),\ 1-\epsilon,\ 1+\epsilon)\hat{A}_t\right\}, \\ L_e &= -H[-\log \pi(\boldsymbol{a}_t)], \\ L_v &= ||R_t - (V_{t-1} + \text{clip}(V_t - V_{t-1}, -\epsilon_{\text{vf}},\ \epsilon_{\text{vf}}))\,||^2. \end{aligned}$$

The coefficients c_e and c_v are set to 0.0 and 0.5, respectively, following the stable training approach from the FinRL framework [13].

2.3 Expert Selection

For our expert pool E, the expert performance vector $\boldsymbol{s}_t \in \mathbb{R}^{|E|}$ is defined as follows:

$$\boldsymbol{s}_t = \alpha \frac{\boldsymbol{\omega}_t - \boldsymbol{\omega}_{t-5}}{5} + (1-\alpha)(\boldsymbol{\omega}_t - \boldsymbol{\omega}_c), \tag{3}$$

where the first term captures the short-term change of cumulative wealth (CW), and the second term represents the long-term change of CW. The CW vector $\boldsymbol{\omega}_t \in \mathbb{R}_+^{|E|}$ is obtained by calling the 5th method defined in Table 1. The symbol $\boldsymbol{\omega}_c \in \mathbb{R}_+^{|E|}$ denotes the previous CW vector at the switching period $c \in [1, t]$. The coefficient α is a preset constant, with a value of 0.3 in this case. Utilizing the definition of $\boldsymbol{s}_t$, we select the expert indexed by $i = \arg\max_i s_{t,i}$ from the expert pool E. Then, we generate the next trading action $\boldsymbol{a}_{t+1}$ by invoking the model of expert i through the API [13]:

```
model[i].predict(subenv, deterministic=True),
```

where the sub-environment subenv is obtained by applying the 6th method defined Table 1.

2.4 Action Sparsification

After obtaining the predicted action $\boldsymbol{a}_{t+1}$ from the selected expert of the expert pool E, it is necessary to sparsify the action to enhance capital efficiency [5]. Let $i = \arg\max_i a_{t+1,i}$. To satisfy the sparse condition $\left\|\boldsymbol{a}'_{t+1}\right\|_0 = 1$, we update $\boldsymbol{a}_{t+1}$ to $\boldsymbol{a}'_{t+1}$ using the formula:

$$a'_j = -b_j,\ \forall j \neq i \wedge b_j > 0, \quad a'_i = \frac{1}{p_i}\left(\omega_{cash} + \sum_{j \neq i} p_j b_j\right), \tag{4}$$

where p_i is the price of the ith stock, $b_j \in \mathbb{R}_{++}$ represents the number of shares of the jth stock, and ω_{cash} is the available cash.

2.5 Optimal Setting of Non-trading interval periods

To reduce the impact of transaction costs, it is necessary to temporarily suspend trading for a specific number of periods [19]. We determine the optimal non-trading interval periods d_* as follows:

$$d_* = \arg\max_{d \in D} \mathbf{1}^\intercal\ (\boldsymbol{\omega}_t - \boldsymbol{\omega}_{t-d}), \tag{5}$$

where $D = \{2, 5, 7\}$ represents the candidate set of non-trading interval periods.

3 EXPERIMENT RESULTS

We conducted experiments comparing PPO-Switch, four RL algorithms [3, 4, 11, 14], and five OPS algorithms [1, 6, 7, 10] on six datasets (Table 3). The results are shown in Table 2.

Table 3: Overview of the datasets

Dataset	Train Period	Test Period
DOW	02/07/2010 - 01/07/2011	04/01/2021 - 30/06/2023
HS	05/01/2011 - 05/01/2012	05/01/2015 - 30/12/2017
CRYPTO	07/10/2020 - 07/10/2021	28/09/2022 - 30/06/2023
HK	27/11/2019 - 27/11/2020	04/01/2021 - 30/06/2023
NYSE	04/02/2013 - 04/02/2014	04/01/2021 - 30/06/2023
FTSE	03/02/2019 - 03/02/2020	03/02/2021 - 30/06/2023

PPO-Switch consistently ranked first in the HS, CRYPTO, and HK datasets, and second in the DOW and NYSE datasets among the six datasets. Notably, in the HS and HK datasets, it outperformed the second-place algorithm by 2.33 and 1.66 times, respectively. PPO-Switch's exceptional performance is attributed to its diverse expert pool and ability to generate sparse actions, enabling effective adaptation to dynamic market conditions. The CW dynamics of all algorithms on the HS dataset (Figure 1) highlight PPO-Switch's robust trading behavior, particularly from the 400th day onwards, where other algorithms fall short.

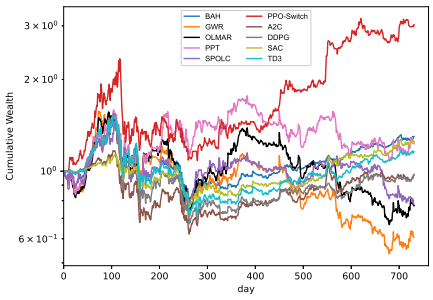


Figure 1: Cumulative wealth dynamics on the HS dataset

REFERENCES

[1] Xia Cai and Zekun Ye. 2019. Gaussian weighting reversion strategy for accurate online portfolio selection. *IEEE Transactions on Signal Processing* 67, 21 (2019), 5558–5570.

[2] Hong-Liang Dai, Chu-Xin Liang, Hong-Ming Dai, Cui-Yin Huang, and Rana Muhammad Adnan. 2022. An online portfolio strategy based on trend promote price tracing ensemble learning algorithm. *Knowledge-Based Systems* 239 (2022), 107957.

[3] Scott Fujimoto, Herke Hoof, and David Meger. 2018. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*. PMLR, 1587–1596.

[4] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*. PMLR, 1861–1870.

[5] Philipp J Kremer, Sangkyun Lee, Małgorzata Bogdan, and Sandra Paterlini. 2020. Sparse portfolio selection via the sorted L1-Norm. *Journal of Banking & Finance* 110 (2020), 105687.

[6] Zhao-Rong Lai, Dao-Qing Dai, Chuan-Xian Ren, and Ke-Kun Huang. 2017. A peak price tracking-based learning system for portfolio selection. *IEEE Transactions on Neural Networks and Learning Systems* 29, 7 (2017), 2823–2832.

[7] Zhao-Rong Lai, Liming Tan, Xiaotian Wu, and Liangda Fang. 2020. Loss control with rank-one covariance estimate for short-term portfolio optimization. *The Journal of Machine Learning Research* 21, 1 (2020), 3815–3851.

[8] Zhao-Rong Lai and Haisheng Yang. 2022. A survey on gaps between mean-variance approach and exponential growth rate approach for portfolio optimization. *ACM Computing Surveys (CSUR)* 55, 2 (2022), 1–36.

[9] Bin Li and Steven CH Hoi. 2014. Online portfolio selection: A survey. *ACM Computing Surveys (CSUR)* 46, 3 (2014), 1–36.

[10] Bin Li, Steven CH Hoi, Doyen Sahoo, and Zhi-Yong Liu. 2015. Moving average reversion strategy for on-line portfolio selection. *Artificial Intelligence* 222 (2015), 104–123.

[11] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971* (2015).

[12] Xiao-Yang Liu, Ziyi Xia, Jingyang Rui, Jiechao Gao, Hongyang Yang, Ming Zhu, Christina Wang, Zhaoran Wang, and Jian Guo. 2022. FinRL-Meta: Market environments and benchmarks for data-driven financial reinforcement learning. *Advances in Neural Information Processing Systems* 35 (2022), 1835–1849.

[13] Xiao-Yang Liu, Hongyang Yang, Jiechao Gao, and Christina Dan Wang. 2021. FinRL: Deep reinforcement learning framework to automate trading in quantitative finance. In *Proceedings of the second ACM international conference on AI in finance*. 1–9.

[14] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*. PMLR, 1928–1937.

[15] Iyad Rahwan, Manuel Cebrian, Nick Obradovich, Josh Bongard, Jean-François Bonnefon, Cynthia Breazeal, Jacob W Crandall, Nicholas A Christakis, Iain D Couzin, Matthew O Jackson, et al. 2019. Machine behaviour. *Nature* 568, 7753 (2019), 477–486.

[16] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

[17] Hongyang Yang, Xiao-Yang Liu, Shan Zhong, and Anwar Walid. 2020. Deep reinforcement learning for automated stock trading: An ensemble strategy. In *Proceedings of the first ACM international conference on AI in finance*. 1–8.

[18] Jianfei Yin, Ruili Wang, Yeqing Guo, Yizhe Bai, Shunda Ju, Weili Liu, and Joshua Zhexue Huang. 2021. Wealth flow model: Online portfolio selection based on learning wealth flow matrices. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 16, 2 (2021), 1–27.

[19] Yifan Zhang, Peilin Zhao, Qingyao Wu, Bin Li, Junzhou Huang, and Mingkui Tan. 2020. Cost-sensitive portfolio selection via deep reinforcement learning. *IEEE Transactions on Knowledge and Data Engineering* 34, 1 (2020), 236–248.

A THE PPO-SWITCH ALGORITHM

The training algorithm for each PPO expert can be found in the FinRL framework[1]. In this context, we present the ensemble algorithm PPO-Switch in Algorithm 1, which utilizes the trained PPO experts $E = \{E_{max}, E_{min}, E_{mean}, E_{ema}, E_{real}\}$ for neural network inference on the test datasets to generate trading actions as output.

[1]https://github.com/AI4Finance-Foundation/FinRL

Algorithm 1 PPO-Switch algorithm

Input:

Expert pool: $E = \{E_{max}, E_{min}, E_{mean}, E_{ema}, E_{real}\}$,
trade-off coefficient: $\alpha = 0.3$,
non-trading interval periods: $D = \{2, 5, 7\}$
number of stocks: m
switch environment: switchEnv

Procedure:

Initialize $\omega = 1,000,000$, $d_* = 2$, $i_c = 1$, $\boldsymbol{\omega_c} = \mathbf{0}$
for $k = 1,...,|E|$ **do**
  env$[k]$, obs$[k]$ = StockTradingEnv(ω).get_sb_env()
end for
for $t = 1 \rightarrow T$ **do**
  for $k = 1,...,|E|$ **do**
    $\boldsymbol{a}[k] = E[k]$.predict(obs$[k]$)$[0]$
    obs$[k]$ = env$[k]$.step$(a[k])[0]$
  end for
  $\boldsymbol{a'} = \boldsymbol{a}[1]$
  if $t <$ max(D) **then**
    switchEnv.step$(\boldsymbol{a'})$
    continue
  end if
  $\boldsymbol{\omega_t}$ = [env[1].asset_memory[t], ..., env[$|E|$].asset_memory[t]]$^\intercal$
  $\boldsymbol{\omega_{t-5}}$ = [env[1].asset_memory[$t-5$], ..., env[$|E|$].asset_memory[$t-5$]]$^\intercal$
  $\boldsymbol{s_t} = \alpha \frac{\boldsymbol{\omega_t} - \boldsymbol{\omega_{t-5}}}{5} + (1-\alpha)(\boldsymbol{\omega_t} - \boldsymbol{\omega_c})$
  $i_* = \arg\max_i s_{t,i}$ //select the best expert i_*.
  $\boldsymbol{a'} = \mathbf{0}$
  if $t \,\%\, d_* == 0$ **then**
    $d_* = \arg\max_{d \in D} \mathbf{1}^\intercal (\boldsymbol{\omega_t} - \boldsymbol{\omega_{t-d}})$
    if $i_c \neq i_*$ **then**
      $\boldsymbol{\omega_c} = \boldsymbol{\omega_t}$
      $i_c = i_*$
    end if
    // get the shares of m stocks.
    $\boldsymbol{b}$ = switchEnv.state$[m+1 : 2*m]$
    $j_* = \arg\max_j \boldsymbol{a}[i_*]_j$ //select the best stock j_*.
    // action sparsification
    $a'_j = -b_j, \ \forall j \neq j_* \wedge b_j > 0$
    $a'_{j_*} = \frac{1}{p_{j_*}} \left(\omega_{cash} + \sum_{j \neq j_*} p_j b_j\right)$
  end if
  switchEnv.step$(\boldsymbol{a'})$
end for
Output: switchEnv.asset_memory[T]

The time complexity of the PPO-Switch algorithm is $O(T(|E|r + m))$, where r represents the inference time of a neural network $E[k]$. The majority of the time cost is incurred during Step 12 and Step 13, where we perform neural network inference for each PPO neural network of $|E|$ experts.

B PARAMETER SETTING OF PPO-SWITCH

The PPO-Switch algorithm utilizes a predefined set of parameters, which are listed in Table 4.

Through an evaluation of the PPO-Switch algorithm with fixed non-trading interval periods $d = 5$ on six test datasets, we collected cumulative wealth for various settings of α, as depicted in Figure 2. Taking into account the algorithm's performance across all datasets, we have chosen $\alpha = 0.3$ as the default value for subsequent evaluations.

Table 4: Parameter Setting

Parameter	Value	Explain
β	$\frac{2}{7}$	Decay factor for EMA used in training PPO experts
α	0.3	Trade-off coefficient for evaluating experts performance across long and short-term durations
D	$\{2, 5, 7\}$	Candidate set of non-trading interval periods

In order to determine the members of the candidate set of non-trading interval periods D, we evaluated the performance of the PPO-Switch algorithm fixed trade-off coefficient $\alpha = 0.3$ on six test datasets. The results, shown in Figure 3, guided our selection of $D = \{2, 5, 7\}$. This choice was made considering that the inclusion of additional members in D would result in longer run times for the PPO-Switch algorithm.

C DIVERSIFICATION OF EXPERT POOL

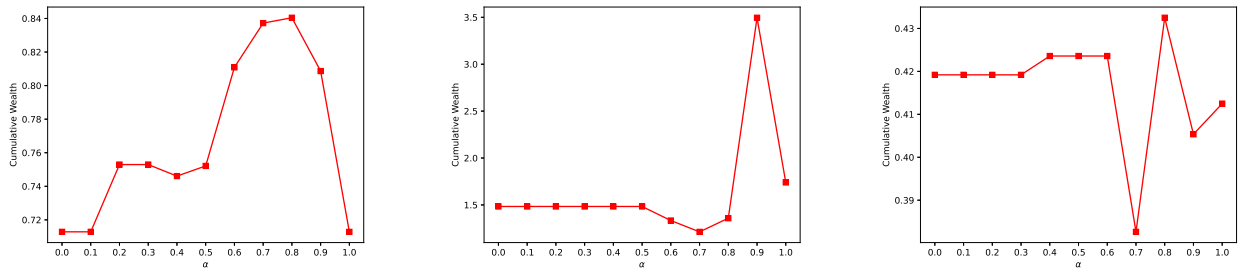
To evaluate the diversification of the expert pool, we conducted tests on six different datasets using the experts in the expert pool $E = \{E_{max}, E_{min}, E_{mean}, E_{ema}, E_{real}\}$ along with the PPO-Switch algorithm. The outcomes of these evaluations are displayed in Figure 4. The results demonstrate that these experts exhibit diverse profit patterns, indicating a range of trading behaviors. For instance, the expert E_{min} (PPOmin) achieved top rankings in the DOW and NYSE datasets, while it performed poorly in the HK dataset, where it was outperformed by another expert E_{ema} (PPOema). This experiment underscores the significance of diversifying the expert pool.

D CUMULATIVE WEALTH DYNAMICS ON SIX DATASETS

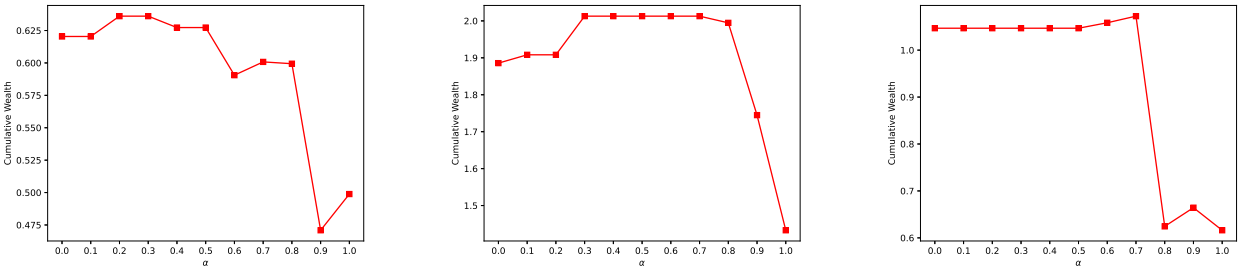
To assess the overall performance of the PPO-Switch algorithm on six test datasets, we present cumulative wealth of 10 algorithms, including BAH, GWR, OLMAR, PPT, SPOLC, A2C, DDPG, SAC, TD3, and the PPO-Switch algorithm itself. The figure 5 displays a process chart depicting the changes in income over time. Notably, on the HS and HK datasets, the PPO-Switch algorithm performs exceptionally well during the middle and late stages of trading. This can be attributed to its diverse expert pool and effective trend-switching technique.

In the case of the DOW and NYSE datasets, the cumulative wealth of the PPO-Switch algorithm exhibited an overall growth trend, although PPO-Switch was later surpassed by the SPOLC and SAC algorithms, respectively. A similar scenario unfolded with the FTSE dataset. These results indicate that there is room for performance enhancement in the PPO-Switch algorithm, such as incorporating action prediction mechanisms from the SPOLC and SAC algorithms.

Received 18 Novermber 2023

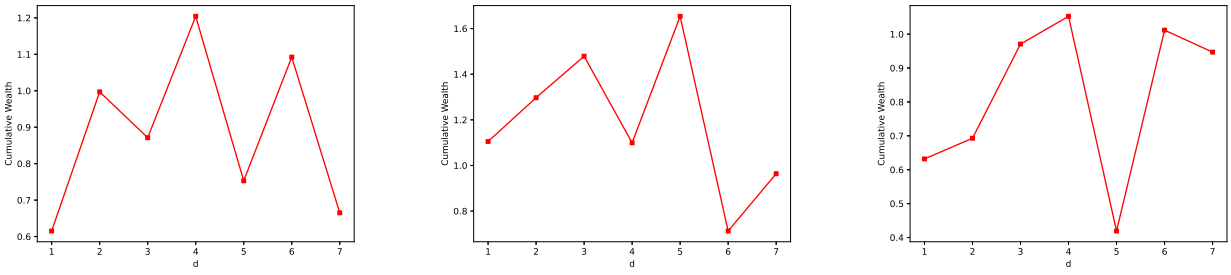


(a) DOW (b) HS (c) CRYPTO

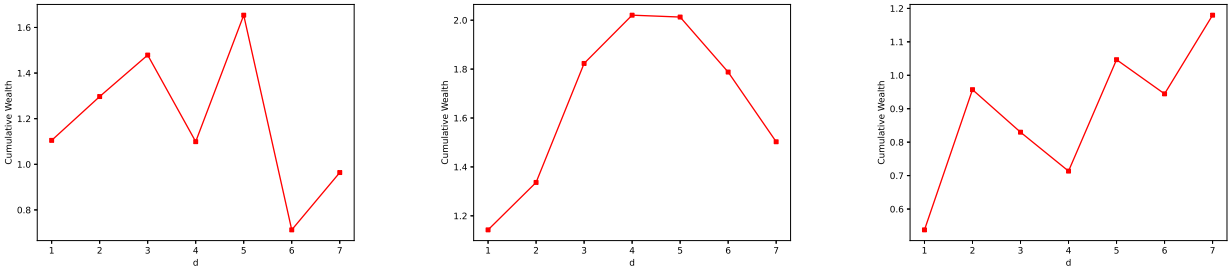


(d) HK (e) NYSE (f) FTSE

Figure 2: Exploration of trade-off coefficient α with fixed non-trading interval periods $d = 5$.

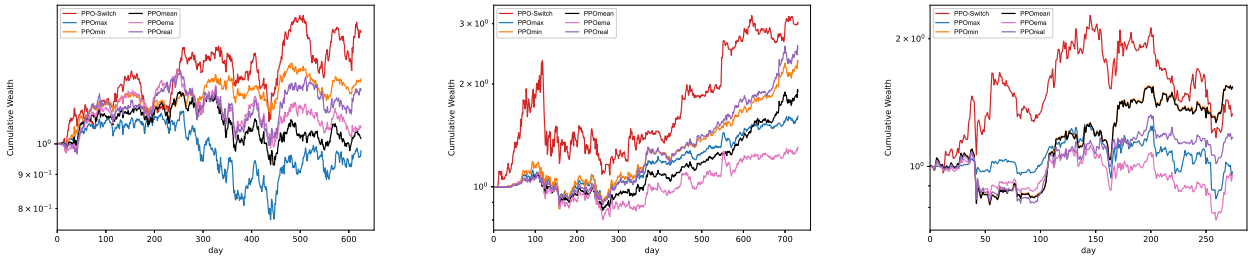


(a) DOW (b) HS (c) CRYPTO

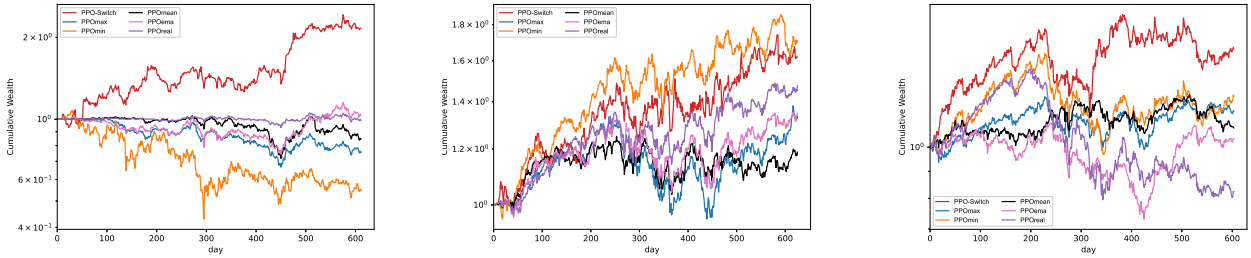


(d) HK (e) NYSE (f) FTSE

Figure 3: Exploration of non-trading interval periods d with fixed trade-off coefficient $\alpha = 0.3$.

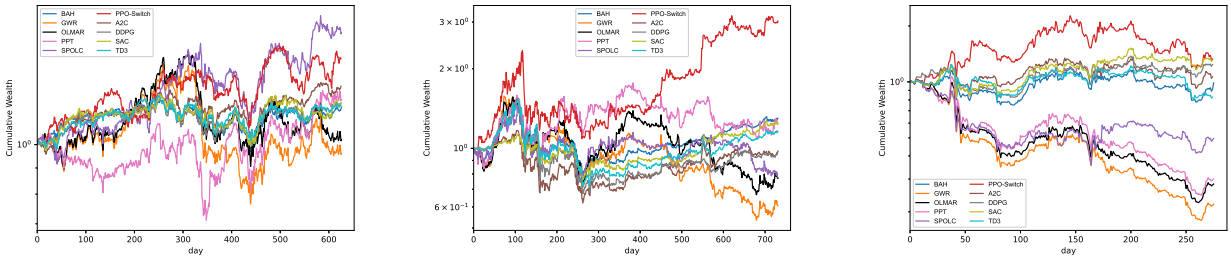


(a) DOW (b) HS (c) CRYPTO

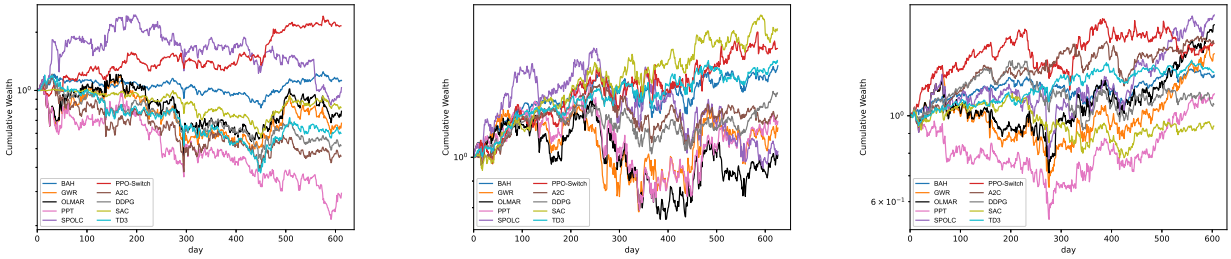


(d) HK (e) NYSE (f) FTSE

Figure 4: Cumulative wealth dynamics of expert pool E.



(a) DOW (b) HS (c) CRYPTO



(d) HK (e) NYSE (f) FTSE

Figure 5: Cumulative wealth dynamics of 10 algorithms.