KAN & FinRL

Group 24 - Artificial Idiot

Introduction

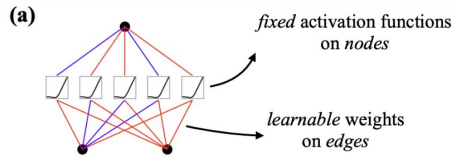
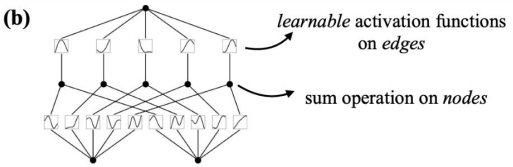
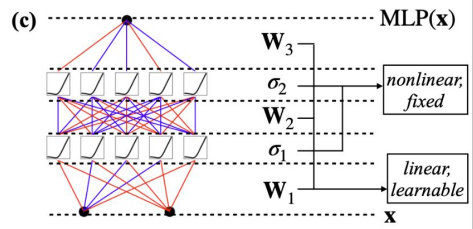
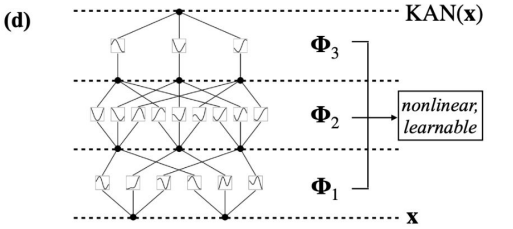
Introduction

- Improve performance of RL on finRL
  - Two Way approach
    - Feature Engineering of dataset
    - KAN to visualize the trained model
- FinRL
  - from NIPS Datasets and Benchmarks 2022

Related work - PPO

- Proximal Policy Optimization (PPO)
- Aims to improve reinforcement learning (RL) stability and sample efficiency
  - By balancing between the efficiency of policy gradient methods and the stability of trust region methods.

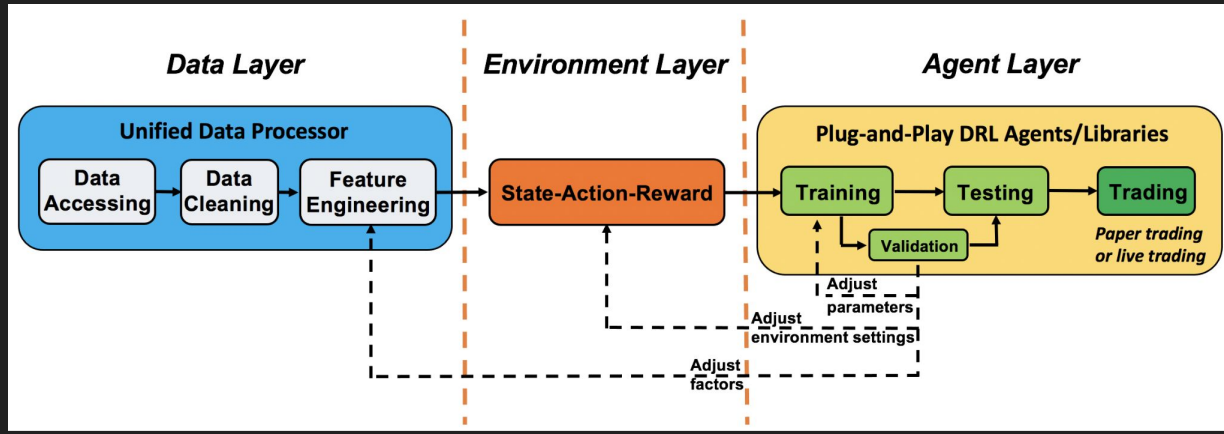
Related work - KAN

Model	Multi-Layer Perceptron (MLP)	Kolmogorov-Arnold Network (KAN)
Theorem	Universal Approximation Theorem	Kolmogorov-Arnold Representation Theorem
Formula (Shallow)	$f(\mathbf{x}) \approx \sum_{i=1}^{N(\epsilon)} a_i \sigma(\mathbf{w}_i \cdot \mathbf{x} + b_i)$	$f(\mathbf{x}) = \sum_{q=1}^{2n+1} \Phi_q \left(\sum_{p=1}^{n} \phi_{q,p}(x_p) \right)$
Model (Shallow)		
Formula (Deep)	$\mathrm{MLP}(\mathbf{x}) = (\mathbf{W}_3 \circ \sigma_2 \circ \mathbf{W}_2 \circ \sigma_1 \circ \mathbf{W}_1)(\mathbf{x})$	$\mathrm{KAN}(\mathbf{x}) = (\mathbf{\Phi}_3 \circ \mathbf{\Phi}_2 \circ \mathbf{\Phi}_1)(\mathbf{x})$
Model (Deep)		

https://arxiv.org/abs/2404.19756

Dataset - FinRL

https://github.com/AI4Finance-Foundation/FinRL-Meta



Dataset - FinRL

	Training Data	Testing Data
Dataset	Dow Jones Industrial Average (DJIA) Components (Data from Yahoo Finance)	
Time Period	2024-01-01~2024-04-30 (Time interval: 1Hr)	2024-05-01~2024-05-15 (Time interval: 15Min)
Size	17430×15	8580×15

Baseline & Evaluation Metric

- The initial amount: 1 Million US Dollars
- Baseline
  - The origin model of FinRL
- Evaluation Metric
  - Return Rate = Return / initial amount = (account_value - initial amount) / initial amount
  - Annualized Return Rate =
    [Return Rate * (trading days each year / trading days in testing period) - 1] * 100%

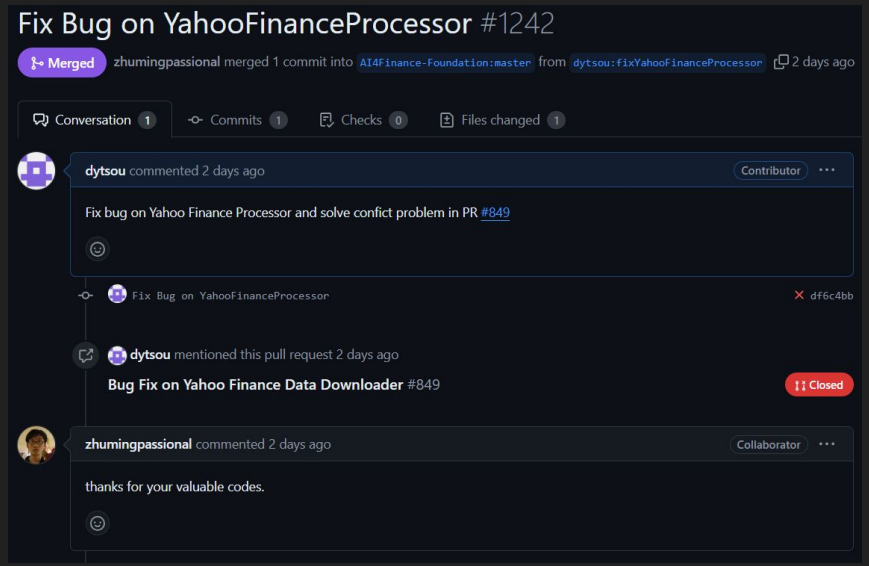
Main Approach

KAN for Visualization

- Train KAN by the input/output of trained PPO
  - Direct replace MLP in PPO will result in long training time
- Plot the trained model to show the relationship between input and output

FinRL feature Engineering

- Implement data fetcher to get financial data from yahoo finance
- Fix the bug in finRL

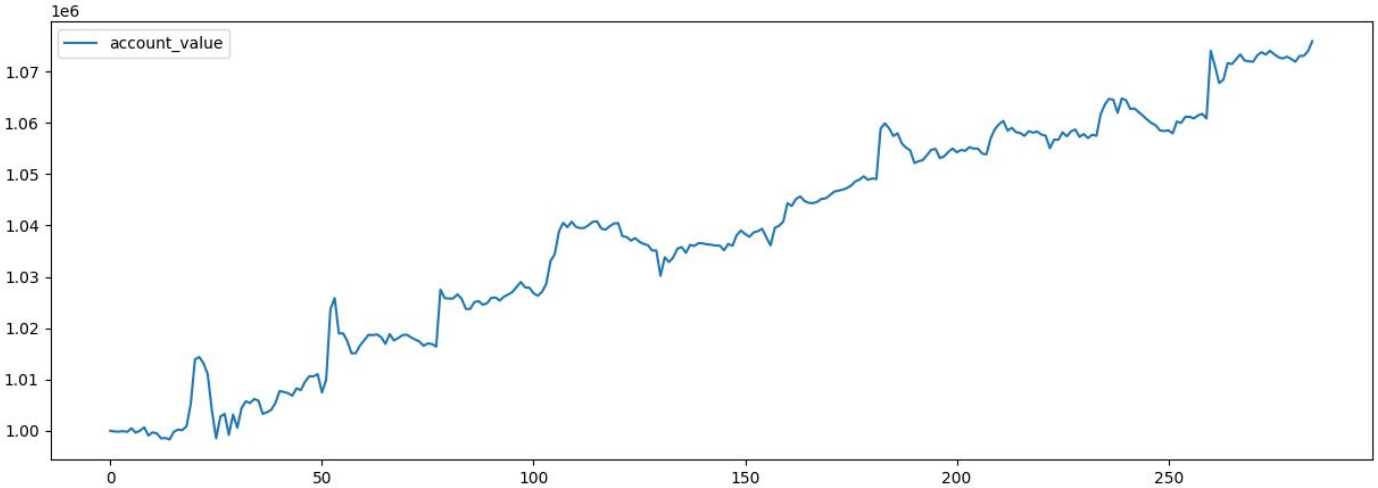


FinRL feature Engineering

- Optimize the gain of returns by utilizing WorldQuant 101 Alphas as factors
- Adjust the time interval of data
- Use the KAN model to identify useless factors
  - Retaining those that are useful for prediction results
  - Eliminating factors that have little impact on the outcomes

Results & Analysis

Test Results

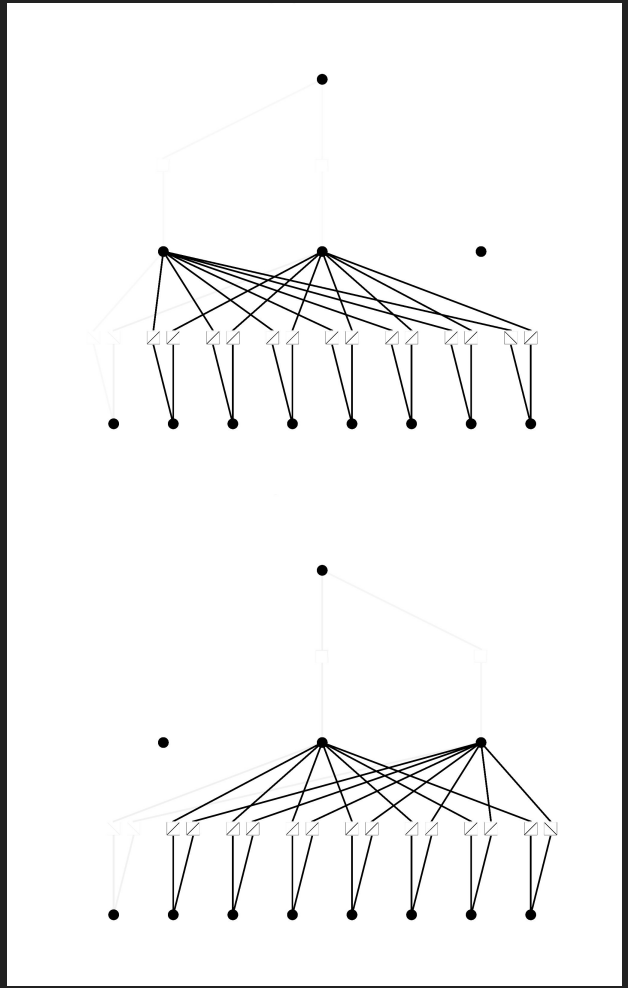


Return Rate in 15 days = 1.08

⇨Annualized Return Rate ≒ 192%

KAN Visualization Result

- GS/MSFT KAN plot result
- Input: ["money", "close", "holding"]+["boll_ub", "boll_lb", "rsi_30", "dx_30", "close_30_sma"]
- Action has a relatively low correlation with money
- Trained function are almost linear
    - can’t fit the input/output well
    - Too much noise in input/output



GS

MSFT

Feature Selection Results

- The short-term results are better than the long-term results

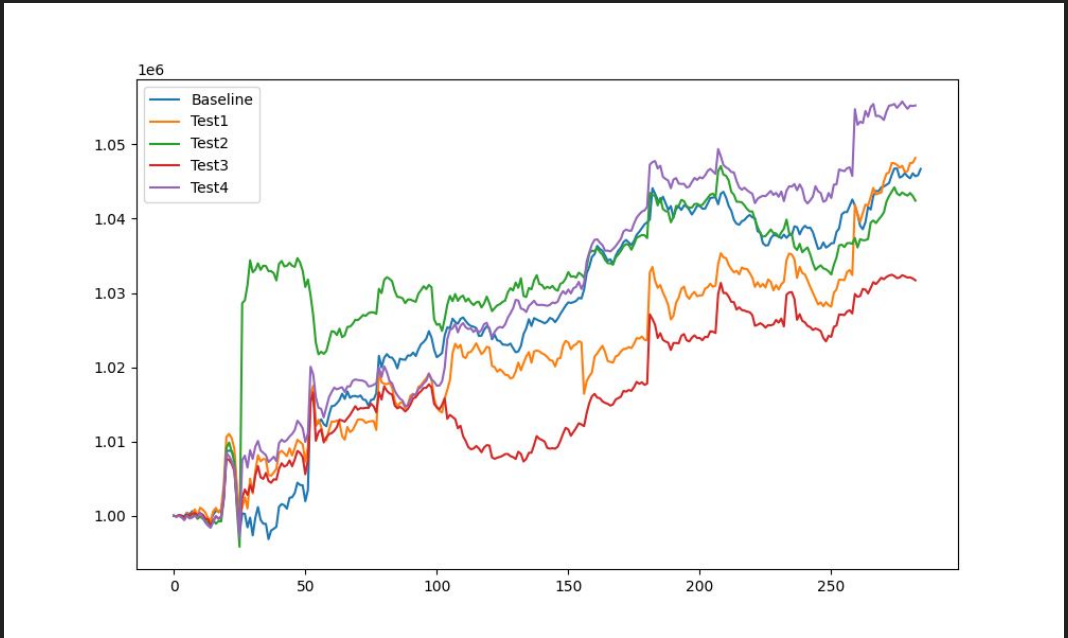
	Long-Term	Short-term
Time Period	2023-05-16~ 2024-05-15	2024-05-01~ 2024-05-15
Return Rate	1.37	1.08
Annualize Return Rate	37%	192%

Analysis:

In long-term trading, the agent's trading frequency is lower, and its strategy tends to be buy and hold.

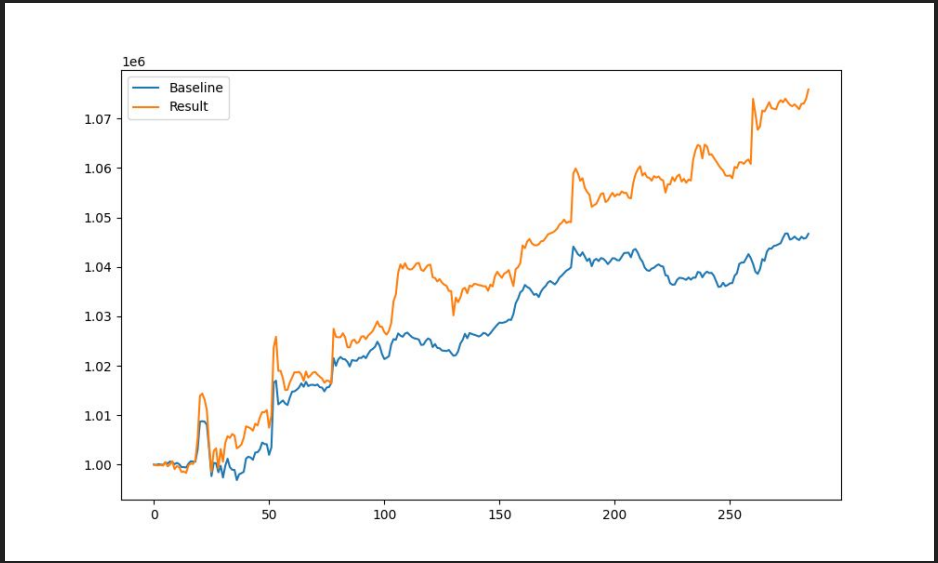
Feature Selection Results

- We tried adding some factors, such as WorldQuant 101 Alphas, to improve the result, but there are no obvious improvement



Feature Selection Results

- After that, we tried to remove some of the indicators, and finally, we get a better result.

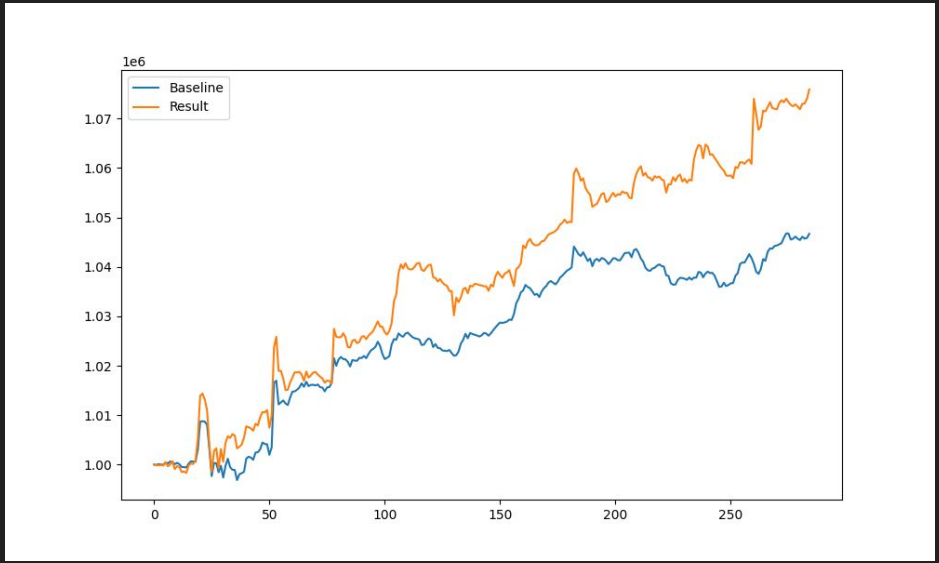


Analysis:

Due to the removal of the indicators, the model can focus on those useful factors. This makes the result better.

Comparison

- Our annualized return rate is twice that of the baseline model.



	Baseline	Result
Time Period	2024-05-01~2024-05-15	
Return Rate	1.04	1.08
Annualize Return Rate	96%	192%

Limitations

- KAN Limitations
  - KAN's potential may not have been fully realized
  - We only trained KAN using the input/output data from the trained PPO model
  - The data was processed in multiple parts
  - The neglect of inter-stock relationships
  - More hyperparameter selection may be needed for better results
  - We could try replacing the MLP in PPO with KAN for testing
    - But much longer training time

Appendix

Appendix

- Github link: https://github.com/freddy645645/NYCU_2024_AI_Final
- Script: Script
- Reference
  - **FinRL**
  - KAN
    - **pykan**
  - **ElegantRL**
  - **kanrl**
  - **FinRL_Contest**

Appendix - Contribution of each member

黃銘宇	KAN visualization: code, slide	35%
鄒東祐	Feature Selection: code, slide	35%
王睿宇	Feature Selection, Presentation	30%
黃致皓		0%