Constructing Equity Portfolios With Deep Reinforcement Learning (using SEC 13F Data)

Alexander Fleiss[1], Amrith Kumaar[2], Adam Rida[3], Ang Li[4], Jialiang Chen[5], Vivian Fang[6], Xinying Lai[7] and Junsup Shin[8]

[1] *Chief Executive Officer of Rebellion Research , New York, USA*
[2] *(Statistics) Cornell University, New York, USA*
[3] *(Applied Mathematics and Data Science) CY-Tech, CY Cergy Paris University, France*
[4] *(Applied Mathematics) Columbia University, Department of Applied Math and Applied Physics, New York, USA*
[5] *(Finance) New York University, NYU Stern, New York, USA*
[6] *(Economics) University of Chicago, Chicago, USA*
[7] *(Quantitative Finance) Rutgers University, New Jersey, USA*
[8] *(Financial Engineering) Columbia University,New York, USA*

Abstract— The ambition of this paper is to catch hidden information inside the Securities and Exchange Commission's (SEC)13F public holding data in order to construct an equity portfolio that maximizes returns. The 13F filing data give us the quarterly stock trading decisions of included funds, but we're not given any insight on how they made their decisions or if information has been flowing between funds. To remedy this lack of knowledge, this paper used feature extraction in order to filter out the best performing funds through several criteria. We propose a method employing powerful machine learning techniques (Deep Reinforcement Learning) to try to catch the missing pieces of information behind the decision process and use them as a prediction tool to construct quarterly equity portfolios. This approach reached an annualized return of 21% with a sharpe ratio of 1.8 outperforming the S&P500 both in returns and stability through historical backtesting.

Keywords— Exchanges/markets/clearing- houses, statistical methods, simulations, big data/machine learning, deep reinforcement learning

I. Introduction

The Securities and Exchange Commission's (SEC) 13F form requires all institutional investors managing greater than $100 million worth of assets to disclose their equity holdings at the end of each quarter. By tracking changes in these quarterly filings, retail investors can glean insights on the movements of "smart money," or capital that is managed by Wall Street's top asset management firms. Since fund managers' smart money has the power to move markets, retail investors can use 13F data to inform their own investment decisions and consequently, construct their own alpha-generating portfolios. This process is commonly referred to as alpha cloning.

Beyond stock-picking, retail investors can develop machine-learning models to leverage data from SEC 13F filings in order to build and routinely adjust their investment portfolios on a quarterly basis. Previous successful quantitative approaches to alpha cloning include the identification of overweighted equities in fund managers' portfolios (compared to a predetermined benchmark), as well as calculating portfolio weights by the number of asset managers who hold a given security at a specific filing period (Cohen, Polk, and Silli 2010; DiPietro 2019). Newer scholarship delves into processing 13F data through filtering and feature extraction. By incorporating extracted features into logistic regression and XGBoost machine learning models, stock price movement can be forecasted and from there, portfolios can be constructed. Additional methods border on the more traditional side, in which the holdings of the top-performing funds among the 13F filings are replicated to construct an aggregate portfolio. (Fleiss, Cui, Stoikov, and DiPietro, 2020).

13F data, however, does present a few limitations in its utility when developing quantitative models and may consequently hinder portfolio construction. SEC 13F data is instantaneous in the sense that it only reports fund managers' holdings at a specific point in time; 13F forms can also be filed up to 45 days after the end of the quarter. Thus, there is not only a time lag that results in a four-month gap between trades and their filing, but also a lack of differentiation between short-term and long-term positions. Additionally, portfolios can be stabilized by filler holdings, which dilute the presence of stronger ones that asset managers believe will yield the highest return (Cohen, Polk, and Silli 2010). 13F data also is susceptible to herd behavior, or bandwagoning, in which fund managers borrow investment ideas from each other, resulting in overvalued stocks. These shortcomings can ultimately mislead machine learning model construction, but the filtering of data can reduce the magnitude of these errors (DiPietro 2019).

This paper seeks to build on the work of Fleiss, Cui, Stoikov, and DiPietro through a deeper dive into the uses of both data filtering and extraction of features from 13F data in order to build an alpha-generating 50 stock portfolio. The first approach incorporates extracted features into a deep reinforcement learning model; the under-construction portfolio is the agent, which undergoes the action of buying and selling stocks to achieve different states, or stock pools with their respective features. If the state accomplishes the reward of excess positive returns, the model "learns" which trades to execute given the existing portfolio to maximize total return. The second approach employs a random forest machine learning model, which employs decision tree algorithms to predict the probability of positive returns for stock pools and from there, constructing a portfolio with the 50 stocks that will yield the highest return.

After filtering data and extracting features, both the deep reinforcement learning and random forest machine learning models will be backtested with historical stock market data and evaluated by their rate of return (i.e. the parameter that determines the model's degree of success in alpha cloning). Note that model training and backtesting will only go as far as 2013, since 13F data was not digitized before; the scope of the 13F dataset considered in this paper only consists of data thereafter. The models will rebalance their respective portfolios accordingly 46 days after each quarter, following the disclosure of 13F data.

II. Data

For this study, two main databases are used: historical 13F filings and historical prices. These data sets are provided by Rebellion Research in the form of excel csv tables. Each column type and description is shown below. Both historical 13F filings and historical prices contain significant gap, so columns containing missing values have been eliminated. Extra filtering has been applied to increase the accuracy of the training set, detailed in the "Further Filtering" section. Also, the scope of expanding window, which uses all available historical data to train the model, has been limited to 2013-2020 to balance out the noise and over fitting.

Data Processing :

Price data were parsed to ensure that there were no significant gaps, holes, or inaccuracies. The 13F filing data underwent substantial processing. First, individual records were filtered out based on the following criteria:

1. Records that are duplicates.
2. Records wherein QTY is zero.
3. Records wherein MARKET_VALUE is zero.
4. Records that were filed before June 30, 2013.

(Before this date, electronic form filings were not required, leading to many records being unavailable.) Next, entire funds were removed through a series of filters. To remain, a fund had to meet all filtering criteria for at least one year; such funds are labeled survival funds. Filters and their rationale are described in Table 1. Before the application of filters, there were 5,903 unique funds from 2013 to 2018. After application, 1,331 survival funds remained. Further filtering of funds took place after their classification and analysis, detailed in the "features" section.

Filter	Rationale
Volatility: The quarterly volatility for each fund must be less than the median of quarterly volatility for all funds	This allows for the construction of more risk-averse quantitative models
Survival time: The fund must survive for at least two consecutive quarters & Quarterly	There must be at least two consecutive quarters to calculate returns.

Table 1: Fund Filters

Feature	Description
x_1	This allows for the construction of more risk-averse quantitative models
x_2	Percentage of good sub-funds holding the stock as defined by industry cluster
$x_3/x_4/x_5$	30-/60-/90-day historical returns
x_6	Idiosyncratic risk of funds that are currently holding the stock
x_7	Total change in market value for all shares held by filing funds from the past quarter to current quarter
x_8	Total change in quantity of shares being held by all filing funds from the past quarter to current quarter

Table 2: Extracted Features and Description

Column Name	Type	Description
RECORD_ID	Int	identifier for each record
CIK	Char	identifier for each company's fund
CUSIP	Char	identifier for each security
PERIOD_END	Date	period end
FILING_DATE	Date	actual filing date
QTY	Int	number of shares
MARKET_VALUE	Float	total amount of money invested

Column Name	Type	Description
CUSIP	Char	identifier for each security
EXCHANGE	Char	exchange on which the security is traded
TICKER	Char	identifier for each security
DATE	Date	date which the price data corresponds
VOLUME	Int	volume of the security
OPEN	Float	opening price of the security
HIGH	Float	high price of the security
LOW	Float	low price of the security
CLOSE	Float	closing price of the security

III. Fund Classification and Feature Extraction

By classifying funds, we can extract valuable features to be the environment in Reinforcement Learning models. Here are the useful metrics for fund classification.

a. Industry Sector

Primary industry sector focus is on one of the most straightforward ways to categorize a fund. However, funds tend to diversify their holdings, making it difficult to label a fund by one particular sector. Nevertheless, funds can by broken down into sub-funds, each of which contain that funds holdings in an industry sector. This sub grouping method enables us to compare how well a funds sector group performs relative to other funds sector groups.

Each stocks industry classification was identified by web-scraping Yahoo finance. After classifying every stock found in the 13F holdings, the funds were then classified into sub-groups which allowed us to isolate that funds best perform sector groups.

b. Performance

Historical performance is the most direct standard to measure a fund or stock. Funds with higher and more stable performance in the past have shown an ability to analyse market supply and demand, and have shown keen insight in both bull and bear markets. It is useful to classify and identify.

For each fund, the return was recorded from June 30, 2013 to December 31, 2017. We calculated the quarterly yield of each fund during this time period. Then, obtaining the mean performance of all funds by calculating the mean of all the means. Later, We divided all funds into three categories based on the calculated mean performance. Funds whose performance was 20% above the mean performance were named high-performance funds. Funds whose performance was 20% blew the mean performance were names low-performance funds. All the other funds were medium-performance funds.

c. Survival Time

Generally speaking, the longer a fund exists, the more stable its past performance is and the more popular it is with investors. Stable performance does not necessarily mean good performance, but relatively small fluctuations can make investors trust this stock or fund more. What's more, fund liquidation is unpopular. Survival analysis can identify funds with good long-term and stable performance as well as longevity.

The duration of the fund shall be calculated as the time difference between the last filing date and the first filing date. And we also calculate the mean survival time after calculating the survival time of each fund. Later, we defined that funds whose survival time was twice the mean survival time of all funds were high-survival funds. Funds whose survival time was 20% lower the mean were low-survival funds. All the other funds were medium-survival funds.

d. Volatility

Volatility measures the fluctuation of returns for a given stock around the mean price. In general cases, high-risk stocks have high volatility whereas low-risk stocks have low volatility. Since the potential profits of high volatility stocks often surpass that of other asset classes, they are considered an attractive security among investors.

Because the period end of the dataset is in quarterly, the quarterly volatility for each fund was calculated based on its return, and then we took a mean across all funds to set up a standard to categorize each fund as high, medium or low volatility funds. Funds with 20% above the mean were labeled high-volatility funds, and funds with 20% below the mean were labeled low-volatility funds, and all others were labeled medium-volatility funds.

e. features

Based on the two features and classifications, funds were filtered. Shown by Table 1 above.

The Reinforcement Learning strategy uses features extracted from cleaned and filtered 13F and price data to predict the direction of stock prices. Stocks that predict good performance are used to build portfolio.

The features, which are calculated for each stock, are listed and described by Table 2 and further elaborate upon below.

x_1: Computed as the total number of funds that held the stock on a given filing date

x_2: The percentage of funds that are labeled good among all the funds that hold the stock on a given filing date

x_3: The historical return of the stock computed on the time period that is 1 month from today.

x_4: The historical return of the stock computed on the time period that is 2 month from today.

x_5: The historical return of the stock computed on the time period that is 3 month from today.

x_6: Previous works show that funds trading in high-idiosyncratic-risk stocks earn significantly higher abnormal returns than funds trading in low-idiosyncratic- risk stocks. The idea behind is that funds trading high-idiosyncratic- risk stocks often exposed to valuable and secretive information about the stocks. The idiosyncratic risk is measured by the following steps:

1. Run the Fama–French three factor model for each stock in each quarter using a rolling window of 24 months. Take $1 - R^2$ as the idiosyncratic risk measure for each stock, $StockIdio_{stock_i, quarter_j}$
2. Obtain the fund-level idiosyncratic risk by aggregating stock-level risks. Let m indicate the index of a fund, fund m holds N stocks where the portfolio weight of each stock is $w_{stock_i, quarter_j}$, then

$$FundIdio_{fund_m, quarter_j} = \sum_{i=0}^{N} StockIdio_{stock_i, quarter_j} \times w_{stock_i, quarter_j}$$

3. For each stock, average the fund-level idiosyncratic risks of the funds that are currently holding the stock with equal weights on each of the funds. This results in the average idiosyncratic risk of all funds holding a given stock for a specified quarter.

x_7: Computed by the change in market value of the given stock that is held by all the filing funds on the current period and the previous period

x_8: Computed by the change in quantity of shares of the given stock that is held by all the filing funds on the current period and the previous period

Now that the features for each stocks are built, outliers can be corrected by assigning values that are not in
[mean+3 * std, mean−3 * std]
The importance of each feature was then assessed via a linear regression between the future quarterly returns of a stock and each feature. That said, they should be included in the final model because other features are derived from sectors. Coefficients and P-values for each feature are listed in Fig .
As per Fig, features 2, 3, 5, 6 all have P-values smaller than 0.01, indicating that they are statistically significant in predicting future returns.

IV. Modeling Methodology

This problem has been modelled using Deep Reinforcement Learning techniques. The principal benefits of reinforcement learning is in it's ability to catch hidden patterns in the data. To link it with our rationale, if some funds have been exchanging information between them, it would be hard to spot it. This modeling method is supposed to detect this kind of behaviour in the 13F holding data.
The proposed strategy contains 2 alterable parameters. The most significant of which being the data sampling rate (quarterly sampling frequency vs daily sampling frequency). We also alter the testing and training time frames to identify periods of maximum performance. We've also chosen to omit 2020 from our prediction time frame as Covid-19 related pressures caused irregular market activity not ideal in testing our model efficacy.

We develop and evaluate our strategy using two different popular deep reinforcement learning models, A2C and PPO. For each model we've adjusted our 2 alterable parameters until an optimal alpha generating return rate is produced.
In order for the deep reinforcement learning models to be effective, the right reinforcement learning environment has to be created. This paper proposes the following modelling framework:

- An Agent that will take decisions. In our case, it will be our portfolio.
- A State that is reflecting the environment. This is our stock pool with all respective features for each quarter.
- An Action space, to allow our Agent to change it's State. Our State is the buying or selling of some stock.
- A Reward, in order to give a goal to our Agent. The reward is the excess positive returns in our case.

a. Summary of Reinforcement Learning

Reinforcement learning (RL), also known as reinforcement learning, evaluation learning or reinforcement learning, is one of the paradigms and methodologies of machine learning. It is used to describe and solve the problem that agents maximize returns or achieve specific goals through learning strategies in the process of interaction with the environment.

The common model of reinforcement learning is the standard Markov decision process (MDP). According to the given conditions, reinforcement learning can be divided into model-based reinforcement learning (model-based RL) and model free reinforcement learning (model free RL), active reinforcement learning (active RL) and passive reinforcement learning (passive RL). The variants of reinforcement learning include reverse reinforcement learning, hierarchical reinforcement learning and reinforcement learning of some observable systems. The algorithms used to solve reinforcement learning problems can be divided into strategy search algorithms and value function algorithms. Deep learning model can be used in reinforcement learning to form deep reinforcement learning.

Inspired by behaviorist psychology, reinforcement learning theory focuses on online learning and tries to maintain a balance between exploration and exploitation. Unlike supervised learning and unsupervised learning, reinforcement learning does not require any data to be given in advance, but obtains learning information and updates model parameters by receiving the reward (feedback) from the environment.

b. Quick summary of A2C

A2C (advantage actor critical) is a typical actor-critical algorithm that we use as a component in the ensemble method. A2C is introduced to improve policy gradient updates. A2C uses a benefit function to reduce the variance of the policy gradient. Instead of estimating only the value function, the critical network estimates the advantage function . Thus, the evaluation of an action does not only depend on the quality of the action, but also considers how much better it can be. To reduce the high variance of political networks and make the model more robust.

A2C uses copies of the same agent working in parallel to update gradients with different data samples. Each agent works independently to interact with the same environment. After all the parallel agents have finished calculating their gradients, A2C uses a coordinator to transmit the average gradients of all the agents to a global network . So that the global network can update the actor and the critical network. The presence of a global network increases the diversity of training data. Updating the synchronized gradient is more cost effective, faster, and works best with large batches. A2C is a great model for stock trading due to its stability.(see Fig. 1)

Feature	Coefficient	P-Value
x_1	0.000	0.3111
x_2	0.096	0.0053
x_3	0.201	0.0003
x_4	-0.066	0.0838
x_5	0.134	0.0013
x_6	0.144	0.0053
x_7	0.000	0.5713
x_8	0.000	0.4306

Table 3: Coefficients and P-Values of Each Feature

Algorithm 1 Q Actor Critic

Initialize parameters s, θ, w and learning rates α_θ, α_w; sample $a \sim \pi_\theta(a|s)$.
for $t = 1 \dots T$: **do**
  Sample reward $r_t \sim R(s,a)$ and next state $s' \sim P(s'|s,a)$
  Then sample the next action $a' \sim \pi_\theta(a'|s')$
  Update the policy parameters: $\theta \leftarrow \theta + \alpha_\theta Q_w(s,a)\nabla_\theta \log \pi_\theta(a|s)$; Compute the correction (TD error) for action-value at time t:
    $\delta_t = r_t + \gamma Q_w(s',a') - Q_w(s,a)$
  and use it to update the parameters of Q function:
    $w \leftarrow w + \alpha_w \delta_t \nabla_w Q_w(s,a)$
  Move to a $\leftarrow a'$ and s $\leftarrow s'$
end for

Fig. 1: A2C Algorithm

c. Quick summary of PPO

The proposal of proximal policy optimization (PPO) aims to learn from TRPO algorithm and use first-order optimization to achieve a new balance between sampling efficiency, algorithm performance, implementation and debugging complexity. This is because PPO will try to calculate a new strategy in each iteration to minimize the loss function, and ensure that the newly calculated strategy can be no different from the original strategy. At present, openAI has taken PPO as the preferred algorithm in its RL research.

It defines the odds ratio between the new policy and the old policy and we can call it $r(\theta)$.(see Fig. 2)

$$r(\theta) = \frac{\pi_\theta(a|s)}{\pi_{\theta old}(a|s)}$$

Fig. 2: Odd ratio

Now we can modify the objective function of TRPO (see Fig. 3)

$$J(\theta)^{TRPO} = E[r(\theta)\hat{A}_{\theta old}(s,a)]$$

Fig. 3: Changed TRPO

Without adding constraints, this objective function may lead to instability or slow convergence rate due to updating of large and small rungs respectively. Instead of adding a complicated KL constraint, PPO imposes a policy ratio, $r(\theta)$ to stay in a small interval around 1. This is the interval between $1-\epsilon$ and $1+\epsilon$. ϵ is a hyperparameter and in the original PPO paper it was set to 0.2. We can now write the objective function of PPO.(see Fig. 4, Fig. 5)

$$J^{\text{CLIP}}(\theta) = \mathbb{E}[\min(r(\theta)\hat{A}_{\theta_{\text{old}}}(s,a), \text{clip}(r(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_{\theta_{\text{old}}}(s,a))]$$

Fig. 4: PPO Objective

V. Statistical analysis and model performance

In order to compare our model with the S&P500 benchmark we will consider two main metrics. First will obviously be the returns because if buying and holding is more profitable then the interest of having a model become nonexistent.
The second metric that we will use is the Sharpe Ratio. The Sharpe ratio measures the performance of an investment (e.g., a security or portfolio) compared to a risk-free asset, after adjusting for its risk. It is defined as the difference between the returns of the investment and the risk-free return, divided by the standard deviation of the investment (i.e., its volatility). It represents the additional amount of return that an investor receives per unit of increase in risk.

$$S_a = \frac{E[R_a - R_b]}{\sigma_a} = \frac{E[R_a - R_b]}{\sqrt{\text{var}[R_a - R_b]}}$$

where R_aR_a is the asset return, R_bR_b is the risk-free return (such as a U.S. Treasury security). $E[R_a - R_b]E[R_a - R_b]$ is the expected value of the excess of the asset return over the benchmark return, and σ_a is the standard deviation of the asset excess return.

Algorithm 4 PPO with Adaptive KL Penalty

Input: initial policy parameters θ_0, initial KL penalty β_0, target KL-divergence δ
for $k = 0, 1, 2, \ldots$ **do**
  Collect set of partial trajectories $\mathcal{D}_k$ on policy $\pi_k = \pi(\theta_k)$
  Estimate advantages $\hat{A}_t^{\pi_k}$ using any advantage estimation algorithm
  Compute policy update

$$\theta_{k+1} = \arg\max_{\theta} \mathcal{L}_{\theta_k}(\theta) - \beta_k \bar{D}_{KL}(\theta||\theta_k)$$

  by taking K steps of minibatch SGD (via Adam)
  if $\bar{D}_{KL}(\theta_{k+1}||\theta_k) \geq 1.5\delta$ **then**
    $\beta_{k+1} = 2\beta_k$
  else if $\bar{D}_{KL}(\theta_{k+1}||\theta_k) \leq \delta/1.5$ **then**
    $\beta_{k+1} = \beta_k/2$
  end if
end for

Fig. 5: PPO Algorithm

After some tuning, parameters kept for the A2C models where the following :
- 10 steps
- 0.005 ent-coef
- 0.0004 learning rate
- 40000 timesteps

Investment strategy	Sampling granularity	Annual returns	Sharpe ratio
A2C Deep RL	Quarterly	8%	0.9
A2C Deep RL	Daily	21.5%	1.8
PPO Deep RL	Quarterly	8%	0.9
PPO Deep RL	Daily	21.6%	1.82
S&P500 Benchmark	–	20.8%	1.7

Final results and comparison of A2C, PPO and SP500 benchmark for Jan-Dec 2019

Using quarterly 13F data has been shown to be very inefficient. Stocks momentum could not be ignored and caused our model to under-performed the benchmark by more than 6% and gave a very low sharpe ratio indicating a high volatility.

VI. Conclusion

This paper aimed to identify a functional 13F alpha cloning strategy using reinforcement learning. Previous literature such as (Fleiss, Cui, Stoikov, and DiPietro, 2020) suggests that traditional ML techniques such as logistic regression and XGBoost have the capabilities to generate alpha using 13F data. As such, we aimed to expand upon previous research and evaluate the efficacy of more complex techniques such as RL. Using the deep RL models A2C and PPO, we were able to generate alpha, outperforming an SP500 benchmark.

Our results suggest that reinforcement learning can be a valuable asset in quantitative stock trading but requires a large number of time-steps in the training process to be able to generate actionable insights. Furthermore, due to our models limited training time frame, 2013 to 2017, it's uncertain how well the model would perform in times of severe recession. For example, when predicting through 2020 and thus the severe COVID-19 related recessions, our model performed poorly.

However, in times of stable markets our model is proven to outperform the SP 500's annual returns while keeping a lower volatility level which signifies the validity of using machine learning techniques with 13F data.

a. Further Research

We believe the use of Inverse Reinforcement Learning, which derives the models reward function from observing professional trading data, would be an compelling avenue for further study. Among the *S&P*500 stocks, we would able to pick the 50 stocks associate with the highest scores calculated by the extract weights from the model. Then, we would apply our models on those 50 stocks which would provoke a higher sharp ratio. It would also be interesting to see how further up-sampling our data to an hourly rate or some even more granular scale would impact our model results.