Smart Money Trading Strategies using 13F Data

Yu Zhang, Yijia Gao

Abstract

This project focuses on exploring the application of machine learning and reinforcement learning models in developing effective trading strategies using 13F data. The study involves tracking the holdings of major fund managers through 13F data and utilizing predictive models to generate insights for trading decisions. By analyzing the data through these models, the project aims to provide new insights into the development of successful trading strategies in the financial markets.

Keywords: Trading Strategies, 13F, Machine Learning, Reinforcement Learning

1 Introduction

Form 13F is a quarterly report mandated by the Securities and Exchange Commission (SEC), which institutional investment managers with assets under management of $100 million or more are required to file. It reveals their equity holdings and offers valuable information about the investment decisions of large fund managers, providing insights into the behavior of smart investors in the market. As the information in this report is publicly accessible, individual investors can analyze the holdings of these major players and attempt to replicate their investments, given that those who work in large funds typically possess knowledge and expertise in stock analysis. This allows retail investors to gain valuable insights from the investment decisions made by large fund managers.

Certainly, previous research has shown promising results in utilizing machine learning algorithms to construct trading strategies based on 13F filings data. For instance, Yinan et al[1] have successfully used models like Random Forest, XGBoost, and CatBoost to predict stock movements and generate alpha. However, their findings revealed that these strategies may perform well only in bullish markets and are inconsistent in bearish markets. Building on this, Alexander et al[2] have demonstrated that more advanced models, like reinforcement learning, can be employed to predict the performance of investment managers and generate higher returns than the S&P 500. These studies suggest that machine learning models can help identify insights from 13F data and construct profitable trading strategies, thereby providing valuable tools for investors seeking to optimize their portfolios.

Mikkilä and Kanniainen (2023) explore a pure data-driven approach to hedging strategies in their paper, "Empirical Deep Hedging"[3]. They rely on empirical intra-day data from stock and option markets, negating the need for specific volatility or jump dynamics specifications. The authors use Deep Reinforcement Learning (DRL) with continuous actions for hedging derivative securities. This study uses six years of intra-day option price observations on the S&P500 index for training, with other periods for validation and testing. The results showed that a DRL agent trained with synthetic data from a calibrated stochastic volatility model surpasses the traditional Black-Scholes delta hedging strategy. This suggests that DRL can effectively capture S&P500 dynamics from actual intra-day data and self-learn to efficiently hedge real options. Their methods inspire us to use the DRL method to portfolio construction using 13F and stock price data, too.

1.1 Objectives and Methods

This paper is dedicated to the design and implementation of three distinct trading strategies, each seeking to deliver superior returns by leveraging insights from 13F filings and applying advanced machine learning and reinforcement learning techniques. The significance of this project stems from the prospect of gleaning and capitalizing on the expertise of top-performing fund managers.

The first strategy is straightforward: it proposes to construct a portfolio for the upcoming quarter comprising stocks most frequently held by fund managers in the previous quarter. The second strategy applies machine learning models, including K-means, Ridge regression, and Random Forest, to distill meaningful patterns from 13F data and generate trading signals accordingly.

The third strategy, the most sophisticated of the trio, integrates the gleaned features into a Deep Reinforcement Learning (DRL) model. This model is capable of learning and adapting over time to make optimal trading decisions. These strategies collectively aim to curate a high-performing portfolio of 50 stocks, outpacing the S&P 500 index by capitalizing on the acumen of the most successful fund managers.

2 Data

The two datasets are provided by CEO and chairman of Rebellion Research, Alexander Fleiss. In our study, we used two datasets to build and validate our trading strategies. The first dataset is the 13F filing data, which contains information on the holdings and positions of the investment managers from each quarter. The second dataset is the stock price data from 2013 to 2020. This dataset contains daily stock price information for a variety of companies. The tables 1 and 2 in appendix show the descriptions for each column.

2.1 Data Processing

After filtering the 13F data by time, we only kept records that were filed after June 30, 2013. This is because prior to this date, companies were not required to disclose their holdings[1]. Once we had this subset of data, we merged it with the stock price data using iCUSIP and iPERIOD_END as the merging keys. This allowed us to match the holdings data with the corresponding stock prices. To ensure the accuracy and consistency of the dataset, To summarize, we performed the following data processing steps:

- Truncated the starting period to be June 30, 2016.
- Dropped any missing values and duplicates from the dataset.
- Dropped records with holding QTY equal to 0.
- Dropped records with market value equal to 0.

These steps resulted in a clean and reliable dataset that we could use for our analyses and models.

2.2 Feature Engineering

The original data lacked the necessary information for building effective models, so we performed feature engineering to generate additional features that could improve the accuracy of predictions. Some of the new features we created are listed below:

- **Return:** Return is the quarterly change in price of a stock between two consecutive IPERIOD_END dates in percentages. It is calculated as follows:

$$\text{Return} = \frac{\text{pSP\_CLOSE} - \text{pSP\_OPEN}}{\text{pSP\_OPEN}} \times 100$$

- **Average Trading Volume:** The average trading volume represents the mean number of shares traded daily within a given IPERIOD_END. It is calculated as follows:

$$\text{Average Trading Volume} = \frac{\text{pSP\_VOLUME}}{\text{Number of Trading Days}}$$

- **Volatility:** Quarterly volatility is a measure of the price fluctuations of a stock within a given IPERIOD_END. It is calculated as follows:

$$\text{Volatility} = \sqrt{\frac{\sum(\text{pSP\_CLOSE} - \text{pSP\_OPEN})^2}{\text{Number of Trading Days}}}$$

- **Turnover:** Quarterly Turnover is a measure of how actively a stock is traded within a given IPERIOD_END. It is calculated as follows:

$$\text{Turnover} = \frac{\text{pSP\_VOLUME}}{\text{QTY}}$$

- **Market Capitalization:** Market Capitalization is the total value of all outstanding shares of a stock at a specific IPERIOD_END. It is calculated as follows:

$$\text{Market Capitalization} = \text{pSP\_CLOSE} \times \text{QTY}$$

2.3 Summary Plots

By making some plots, we can discover some excellent fund managers and the stocks most favored by those investors.

Figures 2, 1, 3, and 4 present informative visualizations that offer valuable insights for investors. Figures 2 and 1 showcase successful fund managers along with their unique identifiers, highlighting their achievements in the industry. Figure 3 reveals the most favored stocks among these managers over the past few years, with a notable emphasis on the tech sector, which has attracted significant investment.

Analyzing these figures, a noteworthy observation emerges regarding the allocation of capital to individual stocks. Despite the managers' substantial control over sizable asset portfolios, their relatively conservative investment approach underscores a keen focus on portfolio liquidity and diversification. This prudent strategy ensures that the investors' interests are safeguarded while optimizing risk and reward.

These figures provide tangible evidence of the meticulous investment strategies employed by these managers, reflecting their commitment to maximizing returns while managing potential risks.

3 Trading Strategies

3.1 Simple Strategy

We implemented a straightforward strategy as our initial approach. In this strategy, stocks are ranked based on their investment amounts in the current quarter, and the top 50 stocks are carefully selected to shape the portfolio for the subsequent quarter. To maintain balance and fairness, equal weights of 2% of the total investment are assigned to each stock. This methodology aims to consistently include the most lucrative and reliable holdings from the previous quarter within the portfolio.

However, after conducting rigorous testing and analysis, we observed that the returns yielded by this strategy were remarkably unstable. While there were instances of significant profits, there were also periods where the portfolio value experienced substantial declines. This volatility in returns highlights the inherent unpredictability and challenges associated with this particular approach.

3.2 Machine Learning Strategy

This section outlines our unique strategy that fuses various machine learning techniques and adopts a quarterly re-balancing period. Given the comprehensive set of features and the massive volume of data at our disposal, our primary objective is to harness the insights of top fund managers. To this end, we implemented a strategy dubbed "the best of the best". This strategy involves identifying the stock with the highest predicted return at the end of each quarter, and then assigning weights to it based on the total investment made by fund managers in the previous quarter. For instance, if our machine learning model predicts that Apple Inc. (AAPL) will be among the top performers in the next quarter, we would incorporate AAPL into our portfolio for the next quarter. The weight assigned to AAPL would be proportional to the total market investment in AAPL during the previous quarter compared to the total market investment in the other 50 stocks in our portfolio.

For example: Let W_{AAPL} denote the weight of AAPL in our portfolio, I_{AAPL} denote the total market investment in AAPL in the last quarter, and I_{Total} denote the total market investment in all 50 stocks in our portfolio in the last quarter. The weight of AAPL in our portfolio can be calculated as follows:

$$W_{AAPL} = \frac{I_{AAPL}}{I_{Total}}$$

This weight represents the ratio of the last quarter's total market investment in AAPL to the last quarter's total market investment in all 50 stocks in our portfolio. This method simply allows us to leverage market view to the stock and no need to assign a new weights based on other techniques like Mean-Variance optimization.

To delve deeper into the specifics, after normalizing the data, we initially cluster the stocks into distinct groups. Subsequently, we employ Principal Component Regression with an l_2 regularizer (Ridge regularization) to capture the linear relationship between factors and stock prices, thereby avoiding the issue of overfitting. In addition, we utilize a Random Forest model with a maximum depth of 4 and 50 trees to encapsulate the non-linear relationships. After obtaining predictions from both models, we amalgamate the two sets of predictions and select the top 50 stocks to form our portfolio for the subsequent quarter.

3.2.1 ML-Based Strategy Pseudo-code

Below is the Psedo-code for strategies' implementation, see the my Github repo for more details:

Algorithm 1 Portfolio Algorithm using Machine Learning Techniques

```
Sort quarters in ascending order
for each quarter in sorted quarters do
    Extract data for the current quarter
    if there is previous quarter data then
        Calculate the returns for the selected stocks from the next quarter
        Calculate the weights based on the market values of the stocks
        Compute the portfolio return
    end if
    Scale the current quarter data using StandardScaler
    Apply KMeans clustering to the scaled data
    for each cluster in unique clusters do
        if number of stocks in the cluster > 10 then
            Perform PCA and Ridge Regression
            Train a Random Forest model
            Combine predictions from Ridge and Random Forest
            Select the top 50 performers
        end if
    end for
    Store the current quarter data as previous quarter data for the next iteration
end for
```

3.2.2 Results and Discussions

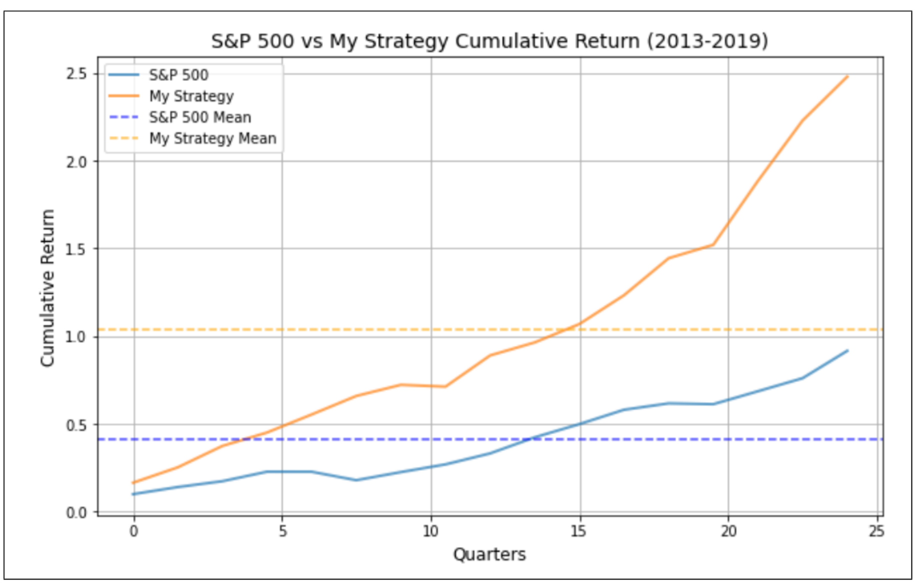

Fig. 1: This plot depicts the returns of the our machine learning strategy over the test period. The x-axis represents quarters, and the y-axis represents the cumulative returns. The orange line indicates the performance of the strategy.

The orange line in the plot represents our strategy, which, as is clearly visible, outperforms the S&P 500 in the long run. However, there are a few critical nuances to consider:

- While both lines demonstrate an upward trend on a quarterly basis, this does not necessarily imply that the strategy yields positive results daily. The backtest results are predicated on the quarterly closing prices of the holdings, which may mask intra-quarter fluctuations.
- The strategy fundamentally thrives in a bullish market. The S&P 500 has largely demonstrated a bullish trend over the past decades, and our strategy leverages this by selecting top-performing stocks that have already been identified and invested in by top asset managers. This approach allows us to tap into the insights of these highly skilled and successful investors.

3.3 Twin Delayed Deep Deterministic Policy Gradient (TD3)

TD3 is an enhanced version of the Deep Deterministic Policy Gradient (DDPG) method for addressing continuous action reinforcement learning problems. It refines DDPG by tackling issues such as overestimation bias and hyperparameter sensitivity. The three key enhancements in TD3 include Clipped Double-Q Learning (utilizing the smaller of two Q-values), Delayed Policy Updates (updating the policy less frequently than the Q-function), and Target Policy Smoothing (adding noise to the target action for the Q-function).
Here are Pseudo-code of TD3 implementation:

Algorithm 2 Twin Delayed Deep Deterministic (TD3) Algorithm

Initialize actor network π and critic networks Q_1, Q_2
Initialize target networks π', Q'_1, Q'_2 with same weights as π, Q_1, Q_2
Initialize replay buffer B
for each episode **do**
    Obtain initial observation state s_1
    for each step **do**
        Select action a_t using policy π with exploration noise added
        Execute action a_t and observe reward r_t and new state s_{t+1}
        Store transition tuple (s_t, a_t, r_t, s_{t+1}) in B
        Sample a batch of transitions from B
        Update Q_1, Q_2 by minimizing the loss: $(Q_i(s_t, a_t) - y_t)^2$, $i = 1, 2$, where $y_t = r_t + \gamma \min_{i=1,2} Q'_i(s_{t+1}, \pi'(s_{t+1}))$
        if step is update step **then**
            Update π by maximizing $Q_1(s_t, \pi(s_t))$
            Soft-update π', Q'_1, Q'_2 with τ towards π, Q_1, Q_2
        end if
    end for
end for

3.3.1 TD3 Agent Configuration and Training

The TD3 agent, consisting of an actor and two critic networks with fully connected layers, is initialized with the state and action dimensions. The actor network selects actions, while the critic networks estimate Q-values of state-action pairs. A decaying exploration noise is used for exploration, and the Adam optimizer adjusts the parameters of the networks. Hyperparameters control the learning process, and a 0.1% transaction cost is applied.

The agent is trained over roughly 4 episodes for each stock, equivalent to two years of data. During training, the agent interacts with the environment, executing actions based on its policy and storing transitions in a replay buffer. The critic networks are updated by minimizing the MSE loss between current and target Q-values, while the actor network is updated 2 times less frequently to maximize Q-value. The target networks are softly updated towards the actual networks' parameters for stability, and the exploration noise decays 20% over time. The training process continues until a pre-defined stopping criterion is met.

3.3.2 TD3 Algorithm Performance and Diagonose

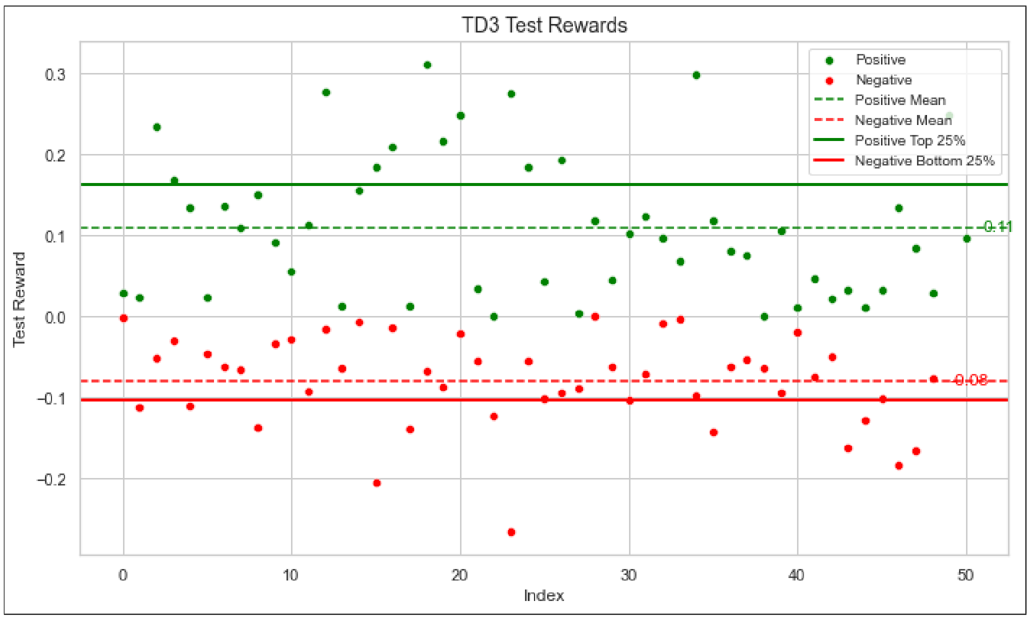

Fig. 2: This graph illustrates the average annualized return of the TD3 algorithm over 100 test runs, with green dots representing positive returns and red dots indicating negative returns.

The TD3 algorithm underperformed compared to our machine learning strategy, often failing to identify promising stocks and holding positions. However, its losses were typically smaller than its gains, signaling potential risk management benefits. Two main issues contributed to this underperformance:

- Limited training steps per episode due to the use of quarterly data.
- Non-stationary nature of the stock return distribution.

4 Further Discussions

4.1 Algorithm Comparison and Enhancement Strategies

When leveraging quarterly 13F data from leading fund managers, our machine learning methodology surpassed the TD3 reinforcement learning approach. This performance can be attributed to two key factors: the advantageous preselection of stocks by the fund managers and our ability to mimic the allocations of top investors, eliminating the need for precise future performance predictions.

For future improvements, we could enrich our machine learning model by incorporating additional factors such as macroeconomic indicators, risk metrics, fundamental analyses, and technical analyses. This could potentially enhance prediction accuracy. On the other hand, for the reinforcement learning model, employing a dataset with greater granularity could provide the agent with ample data for each stock, thus improving its training efficiency and overall performance.

4.2 Reliability of 13F Filings

An analysis from Columbia University revealed that hedge funds could amend their initially reported 13F quarterly holdings through restatements[4]. These restatements, associated with significant abnormal returns, suggest strategic misreporting of original holdings to conceal trading intentions. Therefore, a critical future challenge involves developing methods to identify and accurately interpret these holdings.

4.3 Potential for Weapons of Math Destruction

"Weapons of Math Destruction" (WMDs), as defined by data scientist Cathy O'Neil, are mathematical models or algorithms that have the potential to cause detrimental societal impacts. These models, if opaque, scalable, and damaging, can unintentionally perpetuate social inequality and bias.In the context of our machine learning and deep reinforcement learning strategies, the potential for creating a WMD is contingent on their transparency, scalability, and potential for harm. However, in finance, algorithms primarily serve as decision-making tools, with the ultimate responsibility for these decisions resting on the users. Emphasizing responsible use, ensuring transparency, and establishing proper safeguards can minimize the risk of our algorithms becoming a WMD. Nevertheless, we must persistently monitor their impacts to ensure fair operation and avoid unintentional harm or instability.

4.4 Data, Code, and Contributions

The data used in this research is proprietary to Rebellion Research and cannot be publicly disclosed. However, the code, methods, and procedures used in this project are available on Yu Zhang's GitHub repository: SmartMoneyStrategy.

The work was divided among team members as follows: Yijia Gao was responsible for data processing, statistical analysis, generating plots, devising simple strategies, and writing the paper and formatting it in LaTeX. Yu Zhang is responsible for development of the Machine Learning and Reinforcement Learning strategies and also contributed to the paper's writing and formatting.

A Appendix: tables

Table 1: Asset Manager Holdings

Field	Type	Description
RECORD ID	Integer(11)	Unique identifier for each record
iCIK	Char(10)	Unique identifier for each asset manager
iCUSIP	Char(9)	Unique identifier for each security
iPERIOD END	Date	Quarter to which filing corresponds (YYYY-MM-DD)
iFILING DATE	Date	Actual filing date (YYYY-MM-DD)
iQTY	Integer	Number of shares held
iMARKET VALUE	Float	Total amount of capital invested in this security

Table 2: Stock Price Data

Field	Type	Description
pSP_CUSIP	Char(9)	Unique identifier for each security
pSP_EXCHANGE	Char(10)	Unique identifier for the exchange
pSP_TICKER	Char(10)	Ticker symbol of the stock
pSP_DATE	Date	Date of the stock price
pSP_VOLUME	Integer	Volume of the stock traded
pSP_OPEN	Float	Opening price of the stock
pSP_HIGH	Float	Highest price of the stock
pSP_LOW	Float	Lowest price of the stock
pSP_CLOSE	Float	Closing price of the stock

B Appendix: plots

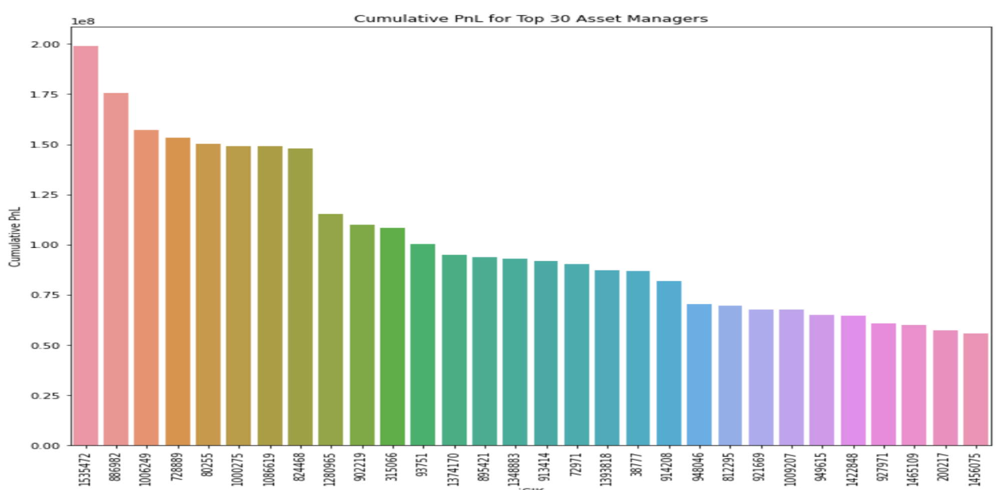

Fig. 1: Top Managers by Cumulative Returns

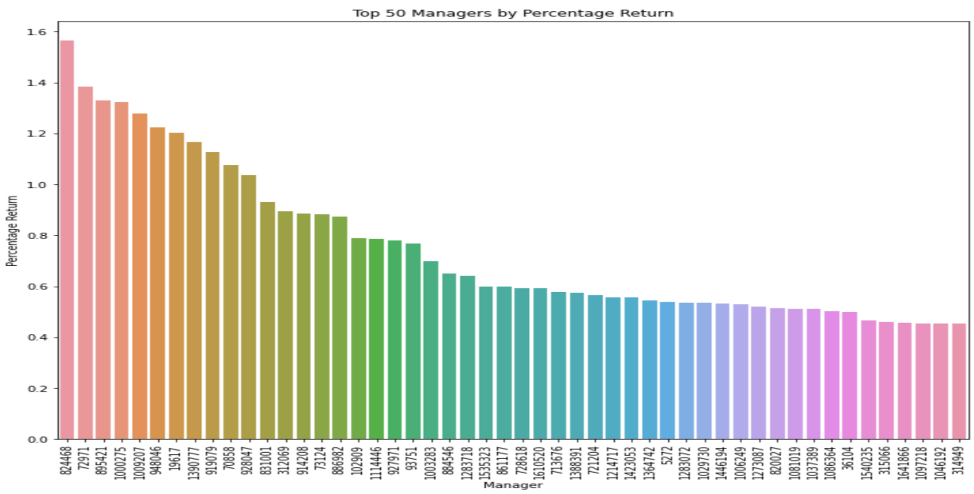

Fig. 2: Top Investors by Return Percentage

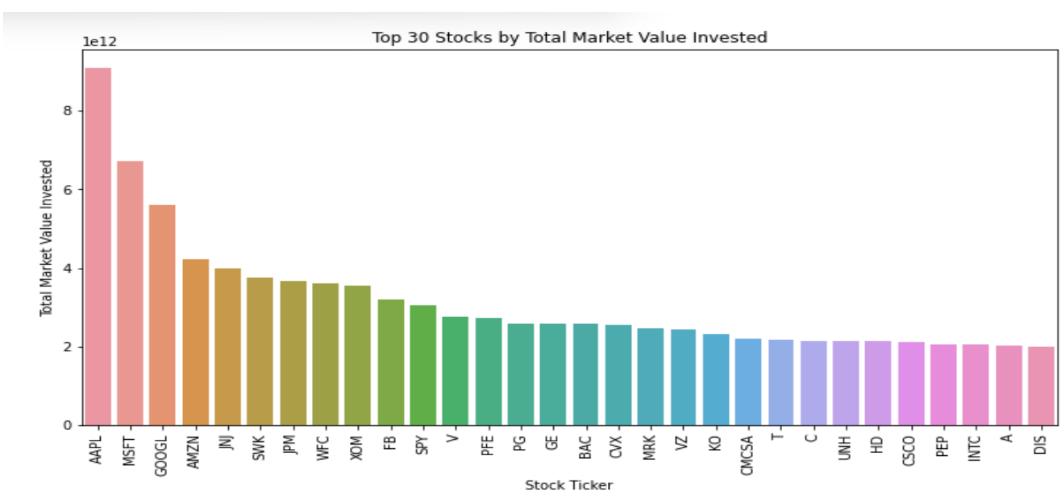

Fig. 3: Most Invested Stocks

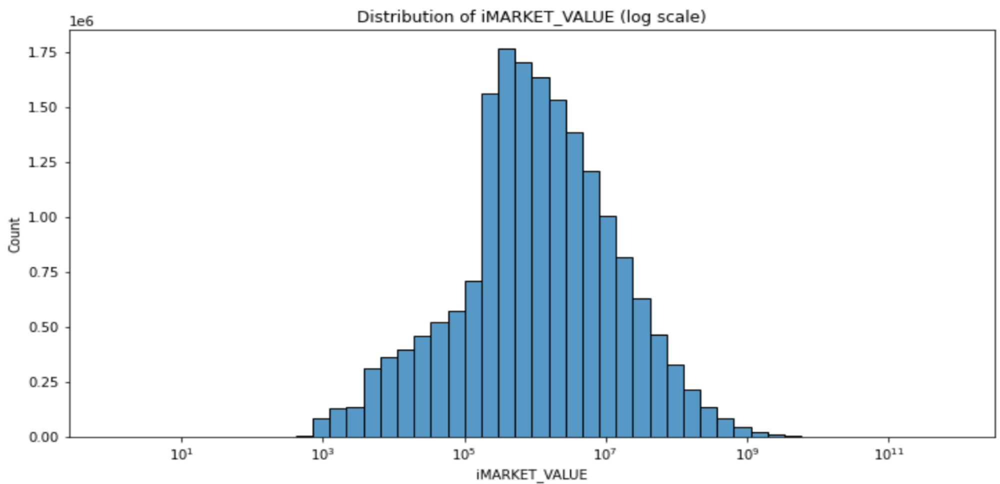

Fig. 4: Distribution of Investment Amount

References

[1] Chen, Y., Song, Y., Qian, S., Fleiss, A.: Applying machine learning to sec 13f investment manager filings for portfolio construction and rebalancing. Available at SSRN 4352145 (2023)

[2] Fleiss, A., Kumaar, A., Rida, A., Shin, J., Lai, X., Fang, V., Chen, J., Li, A.: Deep reinforcement learning & feature extraction for constructing alpha generating equity portfolios. Available at SSRN 3958478 (2021)

[3] Mikkilä, O., Kanniainen, J.: Empirical deep hedging. Quantitative Finance **23**(1), 111–122 (2023) https://doi.org/10.1080/14697688.2022.2136037

[4] Cao, S.S., Da, Z., Jiang, X., Yang, B.: Do hedge funds strategically misreport their holdings? evidence from 13f restatements. Available at SSRN (2022) https://doi.org/10.2139/ssrn.3907560