# Portfolio Optimization with Machine Learning

Lvcheng Dong[1,4], Guangqi Li[2,4], and Changting Song[3,4]

[1]Shanghai Jiao Tong University
[2]Southernwestern University of Finance and Economics
[3]Zhejiang Sci-Tech University

[4]Team G1, PAFBMA-402002

March 9, 2023

**Abstract**

This project explores the application of machine learning methods in portfolio optimization. It aims to optimize investment portfolios using machine learning techniques, including Long Short-Term Memory (LSTM), Support Vector Regression (SVR), Ledoit-Wolk-Shrinkage, combining with traditional method of Mean-Variance. Also, ten stocks from different market sectors are selected to minimize non-systematic risk. The project findings indicate that the proposed approach outperforms the market benchmark SP500, achieving positive returns and beating bearish market performance. Overall, this study highlights revolutionary impact of machine learning on the quantitative finance.
**Keywords**: Portfolio Optimization, Machine Learning, Shrinkage

# 1 Introduction

## 1.1 Background

Portfolio optimization is a crucial task in finance that involves selecting a combination of assets that maximize returns while restraining the risk within a tolerable range. Over the years, various methods have been developed to optimize portfolios, ranging from traditional methods such as mean-variance analysis to more advanced techniques such as neural networks.

Furthermore, there have been notable events in the financial industry that have highlighted the importance of diversification. For instance, the GameStop short squeeze in early 2021 and the pandemic-induced market crash. They both underscored the importance of diversification and risk management in portfolio construction[2].

That being said, it is only natural that an investor would like to evaluate how the asset manager do. It is very common to use the benchmark assets and

make comparison about the performance. If the portfolio beats the market, then they are doing better than most of the people.

In this project, we present a novel and recently-popular approach to portfolio optimization, by leveraging the power of numerous machine learning techniques, such as Long Short-Term Memory and Support Vector Regression and etc. The proposed method also involves selecting ten stocks from different sectors, with the aim of reducing non-systematic risk. The results of the proposed approach are evaluated and compared to market benchmark (S&P 500). The findings demonstrate that the proposed approach outperforms the market and traditional methods, achieving higher returns while maintaining a lower level of risk.

## 1.2 Asset Selection

For the purpose of the project, we use the python package `yahoo finance` as our data source[3]. Here is the detail of our assets.

- **Time range:** The data covers a period of 100 trading days, from January 2nd, 2019 to May 26th, 2019. The first 70 days of the data were used as the training set, while the remaining 30 days were used as the validation set. We use the P&L of the last 30 days to evaluate the effectiveness.
- **Assets:** The dataset used in this project consists of the daily stock prices of ten different companies, which is shown in Table 1.

| | | | | |
|---|---|---|---|---|
| Energy | Material | Industrial | C. Cyclical | Healthcare |
| `SHEL` | `CBT` | `UPS` | `TSLA` | `PFE` |
| Technology | Real Estate | Communication | C. Defensive | Financial |
| `AAPL` | `EQIX` | `NFLX` | `WMT` | `GS` |

Table 1: Stock tickers for 10 Assets

- **Benchmark asset:** SP 500. As is shown in Figure 1 and Table 2, 2019 is quite a good year for stocks overall with annualized returns of 28.5% and sharpe ratio of 2.07[3]. However, from Figure 1 one could tell that the validation period of our project (i.e. Apr 12 to May 26, 2019) happens to be the downward period SP 500 with 21.9% loss. It raises a challenge for us: can we do better than the market in this bearish period? Our model proves viable and even capable of achieving positive returns in the subsequent analysis.

| Time | 2019 | 30 days |
|---|---|---|
| Returns | 28.5% | −21.9% |
| Volatility | 12.5% | 12.1% |
| Sharpe Ratio | 2.07 | −1.98 |
| Max Drawdown | −6.8% | −3.4% |

Table 2: Simple Tear Sheet

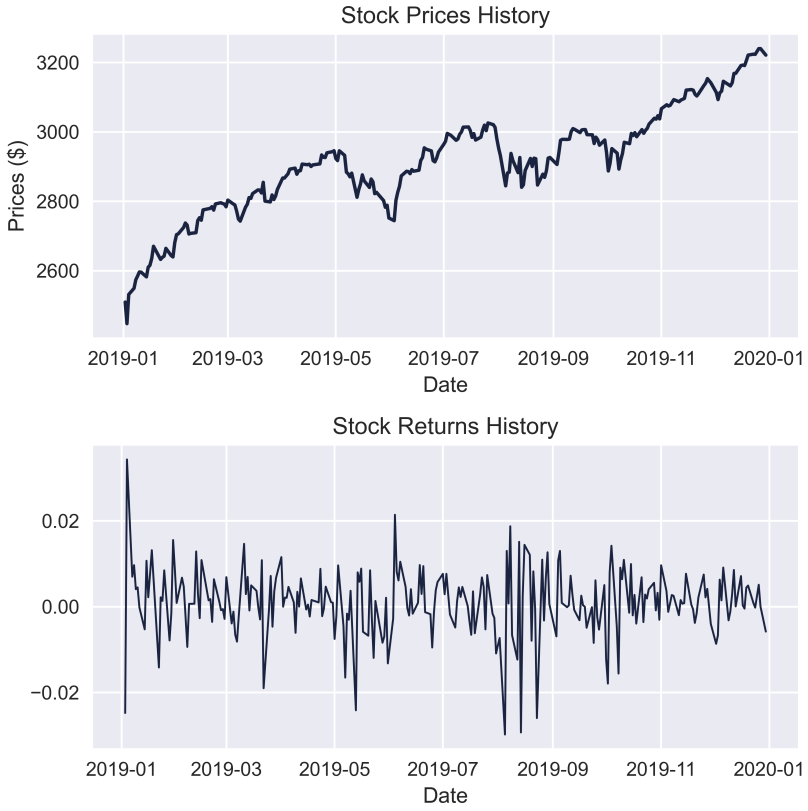



Figure 1: S&P 500

- **Reasons behind stock selection:** We say our portfolio is ideal because it comes from a variety of sectors, allowing for diversification and risk reduction. For instance, we choose Apple and Tesla because they have high market capitalization and popularity. As for Pfizer, it is one of the largest pharmaceutical companies. Netflix, as we both know, is a popular streaming service provider. And last but not the least, the Walmart, is one of the largest retail companies in the world. We believe they are all the strongest representatives of the whole sector's prosperity.

# 2 Methodology

## 2.1 Overall Framework

Basic framework of our project mainly consists of three parts. They are estimation, optimization, validation & evaluation process respectively. Each process is demonstrated in the flowchart in Figure 2, which is explained in detail below.

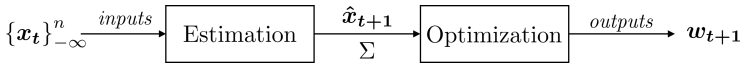



Figure 2: Flowchart of Our Work

- **Estimation Side:** The portfolio is based on estimation results, which is why it is crucial to pay close attention to it. This project aims to explore several prediction methods to achieve better outcomes. The methods used are Support Vector Regression (SVR) and Long Short-Term Memory (LSTM). Additionally, the ARIMA model was considered, but unfortunately, it yielded poorer predictions for seasonal dates. Therefore, the final forecasting methods used were SVR and LSTM, which provided reliable results.

  Based on historical stock data ($\{\boldsymbol{x_t}\}_{-\infty}^{n}$), we select specific training data as input to forecast $\hat{\boldsymbol{x}}_{\boldsymbol{t+1}}$. The prediction results and covariance matrix $\Sigma$ are then provided for optimization in the next step.

- **Optimization Side:** Drawing on the predicted values, we utilize the return and volatility metrics as inputs to construct a mean-variance model, from which we can obtain the optimal portfolio weights $\boldsymbol{w_{t+1}}$. Subsequently, we can easily calculate and compare the portfolio returns, which is precisely what our project aims to accomplish in the following step.

- **Validation and Evaluation:** In this part, we conducts backtesting on the final portfolio to evaluate the returns under known weights and daily returns. Fortunately, the results show that our model is practical and reliable compared to the S&P 500 over the same time frame.

## 2.2 Techniques Involved

### 2.2.1 Support Vector Regression

SVM, as a machine learning algorithm, is successful for forecasting future value of individual stocks and stock market index[4]. SVM builds a hyperplane or a set of hyperplanes in a high-dimensional or infinite-dimensional space for regression. See Figure 3, the hyperplane with the largest distance to the nearest data points of any class (known as functional margin) is considered as the optimal separator, as it typically results in lower generalization error for the classifier[5]. Despite the limitation, previous studies have successfully demonstrated the potential of ML and ANN methods for addressing both large-scale and small-scale problems [6]. To that end, we need to perform parameter tuning to ascertain the best kernel function and other parameters to do the regression.

### 2.2.2 Long Short-Term Memory

LSTM is a type of recurrent neural network that is commonly used in deep learning for time-series forecasting and sequence prediction. It is designed to

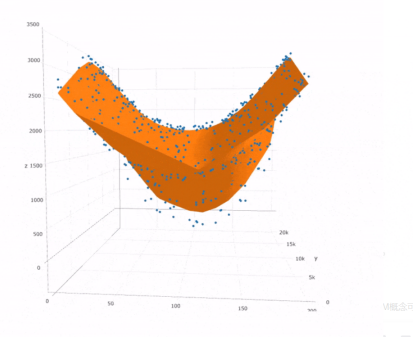


Figure 3: SVR in 3D Space

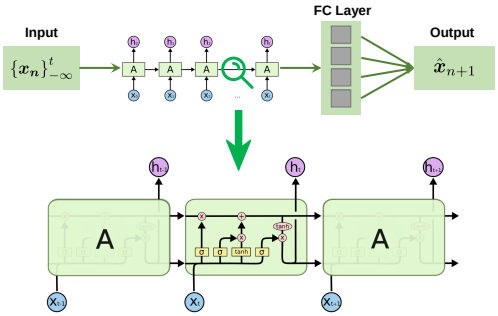



Figure 4: Structure of LSTM

overcome the vanishing gradient problem, which is a common issue in RNNs that makes it difficult to learn long-term dependencies[7]. The key innovation of LSTM is the use of memory cells and gates (the lower diagram of Figure 4), which allow the network to selectively remember or forget information over time. The memory cells can store information for long periods of time, and the gates control the flow of information into and out of the cells[8]. And in our project, we use `tensorflow` to construct a neural network with 2 LSTM layers and 2 fully-connected layers.

### 2.2.3 Mean-Variance Optimization

Mean-Variance Optimization introduced by Economist Harry Markowitz in a 1952 essay[9], is a mathematical framework for assembling a portfolio of assets such that the expected return is maximized for a given level of risk. It is a formalization and extension of diversification in investing. Its key insight is that an asset's risk and return should be assessed by how it contributes to a portfolio's overall risk and return. Mean-Variance optimization adopts the variance of asset prices as a proxy for risk[10]. Under the model, portfolio return $R_p$ is the proportion-weighted combination of the constituent assets' returns $R_i$. Portfolio return volatility $\sigma_p$ is a function of the correlations $\rho_{ij}$ of the component assets, for all asset pairs $(i, j)$.

- Returns

$$\begin{aligned}\sigma_p^2 &= \sum_i \sum_j w_i w_j \sigma_{ij} \mathrm{E}\left(R_p\right) \\ &= \sum_i w_i \mathrm{E}\left(R_i\right)\end{aligned}$$

- Covariance

$$\sigma_p^2 = \sum_i \sum_j w_i w_j \sigma_{ij}$$

The efficient frontier in Figure 5 is the set of optimal portfolios that offer the highest expected return for a defined level of risk or the lowest risk for a given level of expected return[10]. Portfolios that lie below the efficient frontier are sub-optimal because they do not provide enough return for the level of risk.

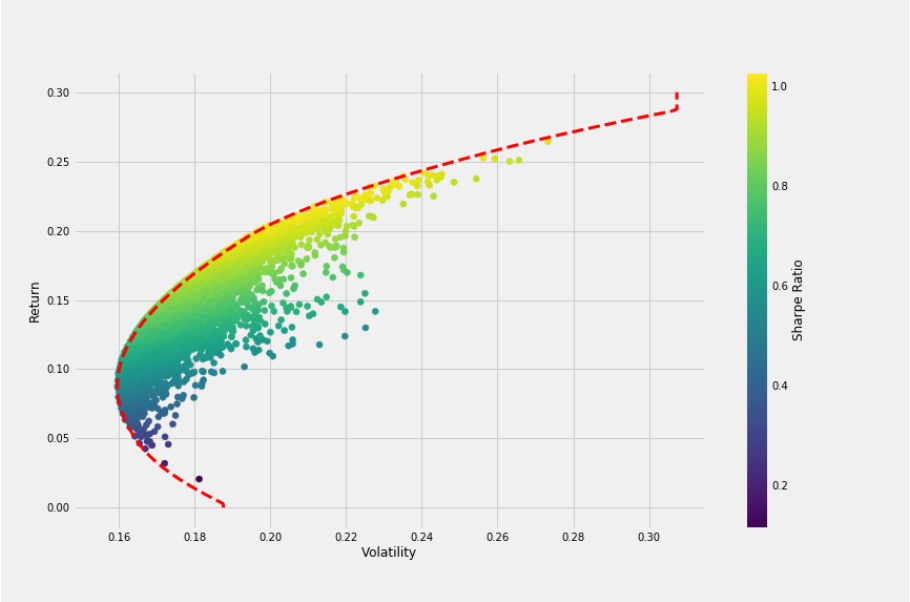



Figure 5: Efficient Frontier

### 2.2.4 Covariance Shrinkage

In the optimization process, the covariance matrix is a key input that measures the pairwise correlation between asset returns. However, estimating the covariance matrix from historical data can be challenging due to limited sample size, noisy data, and non-stationarity of the underlying process[11]. There are two popular methods for improving covariance estimation, Ledoit-Wolf shrinkage and Oracle Approximating shrinkage, one of which basically impose regularization and the other one is more of a data-driven method[11, 12]. For the benefit of low computational cost, we choose Ledoit-Wolf shrinkage . This method basically convert the simple covariance $\Sigma_s$ with a parameter $\delta$ to mix with a target matrix[13]. The method constructs the new covariance matrix as

$$\hat{\Sigma}_{LW} = \delta\hat{\Sigma}_o + (1-\delta)\hat{\Sigma}_S.$$

The optimal value of $\delta$ can be obtained through cross-validation. In that way, we balance the trade-off between estimation bias and variance. This shrinkage method provided in Python packages like `sklearn` or `PyPortfolioOpt`.

# 3 Implementation

## 3.1 Portfolio Optimization

The portfolio allocation is based on Mean-Variance Optimization(MVO) with several adaptions. The purpose is to improve the performance of MVO by introducing new methods on expected returns and covariance matrix other than the sample mean or sample covariance. The final solution is to combine Support Vector Regression (SVR), Long Short-Term Memory(LSTM), and Covariance Shrinkage with MVO.

## 3.2 Estimation

### 3.2.1 SVR method


Figure 6: Correlation of variables

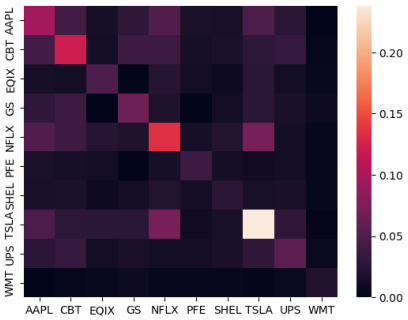



Figure 7: Covariance Heatmap

After selecting ten stocks, the first step in SVR is to divide the data into training and testing sets. It is necessary to normalize the stock prices. The next step is to create a correlation plot of the open, close, high and low prices. The plot in Figure 6 reveals a high degree of correlation between these variables. Since they both contain certain information about the movement of price, we input them as a bundle to forecast the closing price. During model building, selecting the appropriate kernel function and parameters is vital. However, overfitting and underfitting are inevitable challenges in providing accurate predictions, see Figure 8 and Figure 9. Grid and randomized search algorithms are effective techniques to tackle these problems. Finally, we compare the mean squared error (MSE), root mean squared error (RMSE) and R-squared to identify the best model and plot a line graph to visualize the performance.

- **Overfitting & Underfitting:** Figure 8 and 9 reveals potential problems in our model. When we input training data to predict the close price of training data, the model suffers from overfitting, where predicted spots overlap with the real ones. Despite a good fit, the model performs poorly

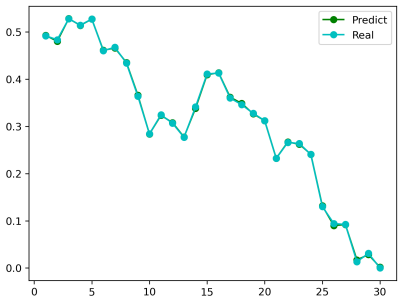



Figure 8: Overfitting

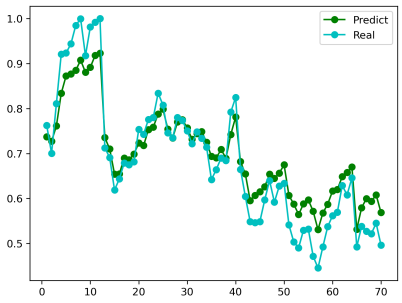



Figure 9: Underfitting

when tested with new data, indicating low universality. To solve this problem, we can use grid and randomized search to try out possible parameters and select the best one. It is possible to obtain smaller prediction errors in the test set than in the training set when using a linear kernel, which is similar to Bruno's conclusion [14].The data used for this experiment is the stock price of TSLA over 30 days, in Figure 10. The final model shows universality in test data, although some offsets are still present.

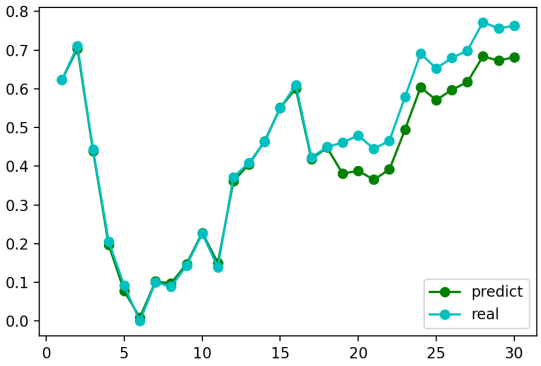



Figure 10: Prediction of FPE

### 3.2.2 LSTM method

We list the steps we take with LSTM model as below:

- First, we split the dataset into training set and test set in a 80-to-20 ratio, like we do in the SVR case.
- Then, we input training data into the model and fit it. Since we believe the most recent information should have the most impact on tomorrow's stock price, we feed the model with a horizon of 7 days for each data

point. So the inputs $\{[x_{t-6} : x_t] \in \mathbb{R}^7 \mid t \in \text{training set}\}$ are in a shape of 7-dimension vector, and the output $\{x_{t+1} \mid t \in \text{training set}\}$ is tomorrow's stock price.

- In this way we build up the model with python packages `tensorflow` and its `Sequential()` object. Set number of epochs to be 30.
- Finally, we evaluate the efficacy of the model with metrics like RMSE, MAE, and make some plots to see it more apparently.

We shall show some plots to demonstrate how our process works. Take Apple for example. We acquire the history data from yahoo finance, as is shown in Figure 11.

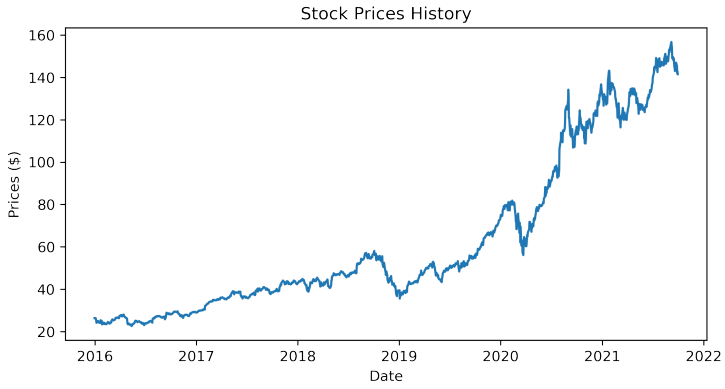



Figure 11: Adj Close of Apple

The next graph is about the choice of epochs, see Figure 12. We plot number of epochs against the loss and discover that after 10 rounds of fitting, the loss is basically stable, so we choose 10 epochs for further prediction. Figure 13 is

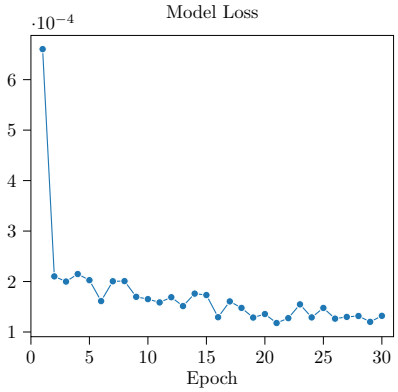



Figure 12: Epochs versus Loss

a demo of how LSTM performs on the estimation of Apple stock. As it turns out, LSTM proves to be quite accurate and robust, since the prediction line and validation line overlap almost everywhere.

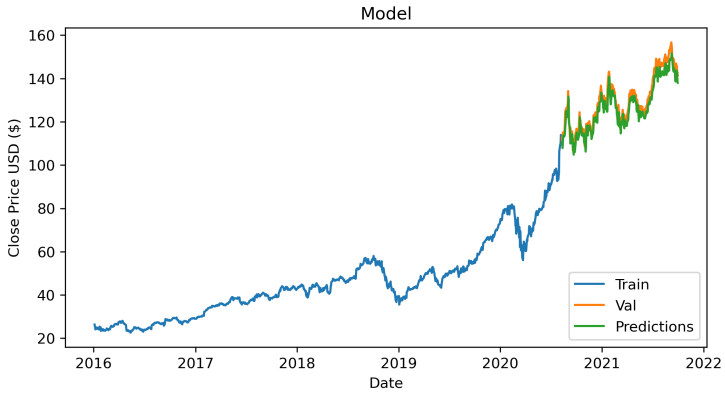



Figure 13: Demo Prediction

Also, we estimate covariance matrix $\Sigma$ in Figure 7, using shrinkage method. The results will be used in the portfolio allocation process.

# 4 Evaluation of Effectiveness

## 4.1 Efficient Frontier

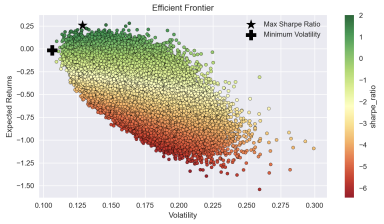



Figure 14: Efficient Frontier(SVR)


Figure 15: Efficient Frontier(LSTM)

We replaced sample mean and sample covariance with the estimation return from Apr. 12th, 2019 to May. 26th, 2019. The covariance matrix was calculated by covariance shrinkage. We draw the efficient frontier by Monte Carlo simulation and generate two types of portfolio weights in Figure 14 and Figure 15, one maximizes sharp ratio, and another minimizes volatility. Here we select a maximized sharp ratio.

### 4.2 Results

We apply each portfolio weight to the closing price from Apr. 12th, 2019 to May. 26th to evaluate their performance. As we can see, the benchmark $S\&P$ 500 return is $-21.9\%$, which means the general market is not in a good condition, that is a test of our methods. And our even distributed 1/n weight portfolio proves this by showing $-42.8\%$ return. But our portfolio weight generated by SVR and LSTM + Covariance Shrinkage has a fairly good performance. At the same level of volatility, they can achieve much higher returns, respectively 16.8% and 19.1% with a high sharp ratio and lower max drawdown.

| | S&P 500 | 1/n | average | SVR | LSTM |
|---|---|---|---|---|---|
| Returns | $-21.9\%$ | $-42.8\%$ | 7.4% | 16.8% | 19.1% |
| Volatility | 12.1% | 16% | 12.6% | 12.5% | 13.1% |
| Sharpe Ratio | $-1.98$ | $-3.42$ | 0.63 | 1.30 | 1.40 |
| Max Drawdown | $-4.5\%$ | $-7.2\%$ | $-2.8\%$ | $-2.7\%$ | $-3.4\%$ |

Table 3: Annualized Tear Sheet for *Apr 14 to May 24, 2019*

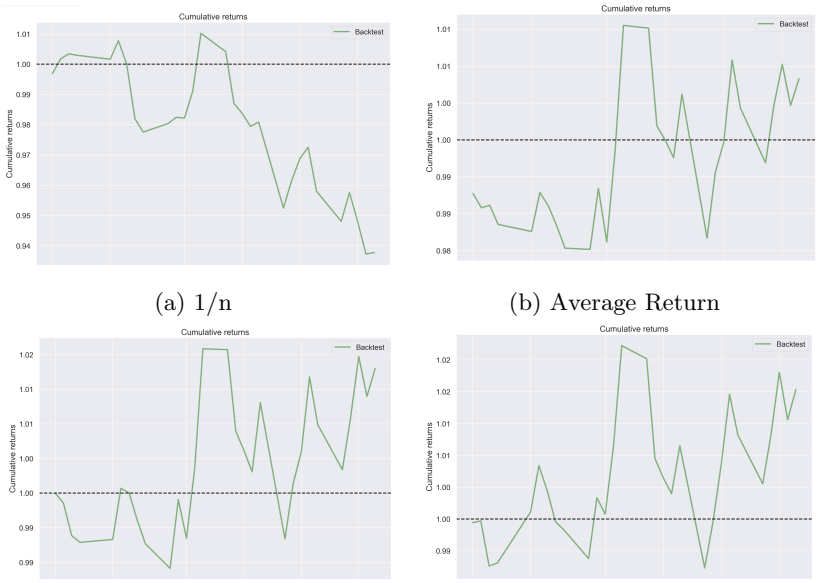
### 4.3 Accumulated Returns

Subfigure 16a to 16d show the cumulative returns of each method. We can tell the difference in their performance, and how the cumulative returns change. By applying SVR/LSTM method we can provide a more accurate estimation of expected returns, thus generating a portfolio with a higher capability to diversify risk.

## 5 Conclusion

Based on prediction and optimization, the portfolio has demonstrated a significant out-performance against the market, delivering an annual return of approximately 19.1% that overwhelmingly exceeds the return of -21.9% of the S&P. It is worth noting that this positive growth occurred during a bearish period, which highlights the portfolio's resilience. The project showcases the deep integration of advanced forecasting technology and portfolio selection through the adoption of various econometrics and machine learning techniques.

However, due to time constraints, there is still further work to be done. The project did not incorporate macroeconomic factors, which can significantly impact the stock market at times. Therefore, a viable solution would be to establish a corpus that contains macroeconomic and policy news to analyze the influence of exogenous shocks. Additionally, the model used in the project is oversimplified, and exploring constraints such as transaction costs, short positions, leverage, and others is necessary to make it more sophisticated and better equipped to tackle potential problems.




(a) 1/n

(b) Average Return

(c) Support Vector Regression

(d) Long Short Term Memory

Figure 16: Accumulated Returns

# A Member Contributions

| Lvcheng Dong | Guangqi Li | Changting Song |
|---|---|---|
| Introduction | Optimization | ARIMA |
| LSTM Model | Tear Sheet Reports | SVR Model |
| Conclusion | Evaluation | Conclusion |
| Slides, LaTeX Support | Slides | Slides |

Table 4: Group Member Contribution Sheet

- **Lvcheng Dong**: My name is Lvcheng Dong. As the group leader of the research team, I organized and facilitated group discussions, fostering a collaborative environment that enabled us to generate innovative ideas. In addition, I took charge of the development of the introduction section, selection of assets, LSTM modeling, and programming. With a background in programming and data analysis, I was able to skillfully use various packages to design and implement machine learning models. Moreover, I utilized my proficiency in LaTeX to create visually appealing and professional-looking slides and reports. Finally, I also led the team in rehearsing and polishing our final presentation, ensuring that it was delivered flawlessly. Through these efforts, I believe I have made significant contributions to the success of the project.

- **Guangqi Li**: My name is Guangqi Li(Aiden). As a team member, I fully participated in the discussion, chose target stocks and form the overall framework together with my team leader. I took charge in the implementation of MVO. I combined the ML/DL estimation to portfolio optimization, generated portfolio weights, and evaluated the performance between different methods. In the mean time, I shared some idea with my teammates, which were beneficial to implementation.
- **Changting Song**: I am Changting Song, a responsible member in our group. We make a deep corporation with each other and determine framework of our project together. At first, I am responsible for ARIMA model in order to predict by time-series analysis. Unfortunately, it shows bad fitting to seasonal data and I choose machine learning method (SVR) to solve problem. During working period, I cope with some overfitting and underfitting problems and provide the prediction from SVR to my partner finally. Also, I offer some personal suggestions and assistant in discussion. It is meaningful and precious for me to attend this project. I am honored to make some teamwork with my partners.

# B Codes

All of the relevant codes of our project are available on Github depository. All mistake is ours. We look forward to your contributions and ideas.

# References


[1] A. Meucci, "Review of Discrete and Continuous Processes in Finance: Theory and Applications," *SSRN Electronic Journal*, 2009. [Online]. Available: http://www.ssrn.com/abstract=1373102

[2] Z. Bodie, A. Kane, and A. J. Marcus, *Investments*. McGraw-Hill Education, 2018.

[3] Yahoo! Inc., "Yahoo finance," https://finance.yahoo.com/, 2023, accessed in February, 2023.

[4] S. Mishra and S. Padhy, "An efficient portfolio construction model using stock price predicted by support vector regression," *The North American Journal of Economics and Finance*, vol. 50, p. 101027, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1062940818302481

[5] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, 2004.

[6] J. Chen, Y. Wen, Y. Nanehkaran, M. Suzauddola, W. Chen, and D. Zhang, "Machine learning techniques for stock price prediction and graphic signal recognition," *Engineering Applications of Artificial Intelligence*, vol. 121, p. 106038, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0952197623002221

[7] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[8] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[9] H. Markowitz, "Portfolio selection*," *The Journal of Finance*, vol. 7, no. 1, pp. 77–91, 1952.

[10] E. J. Elton and M. J. Gruber, *Investments and portfolio performance*. World Scientific, 2011.

[11] O. Ledoit and M. Wolf, "Honey, i shrunk the sample covariance matrix," *The Journal of Portfolio Management*, vol. 30, no. 4, pp. 110–119, 2003.

[12] Y. Chen, Y. Zhang, and M. G. Amin, "Shrinkage algorithms for mmse covariance estimation," *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 5016–5029, 2010.

[13] O. Ledoit and M. Wolf, "A well-conditioned estimator for large-dimensional covariance matrices," *Journal of multivariate analysis*, vol. 88, no. 2, pp. 365–411, 2004.

[14] B. M. Henrique, V. A. Sobreiro, and H. Kimura, "Stock price prediction using support vector regression on daily and up to the minute prices," *The Journal of Finance and Data Science*, vol. 4, no. 3, pp. 183–201, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405918818300060