A Machine Learning approach to Stock Price Prediction and Portfolio Optimization

Jorge Barreno, Anna Skarpalezou, Arjun Tisseverasinghe

New York University Shanghai, Shanghai, China

December 12, 2020

Portfolio optimization is a process of allocating funds into financial assets with the goal of maximizing predicted returns over predicted risk. This paper achieves this by forecasting the stock prices into the future using traditional finance and machine learning approaches to later develop an optimized portfolio. We explore 3 models, namely Moving Average Returns, Multiple Linear Regression and Long Short Term Memory Neural Networks to predict the stock prices while using a custom and loss function to optimize our portfolio. We aim to develop a portfolio which sparsifies the weights of our assets in order to minimize the complexity of our investment strategy.

1 Introduction

Quantitative investment strategies have been growing in popularity, due to their data driven nature. According to Markowitz (Markowitz, 1991), portfolio creation is comprised of two main steps; first is the analysis of historical data, which informs our expectations for future behavior of the financial assets, and then capitalizing on these insights is the optimization procedure. The objective of this work is to develop an approach that utilizes Machine Learning techniques for stock price prediction, comparing these against traditional statistical approaches. From these predictions, an optimized and sparcified portfolio is created for the Sharpe Ratio, a measure of returns over risk, acknowleding the investor preferences for simplicity. We begin by gathering stock information, which is used to train our 3 stock price prediction models. These models include a simple Moving Average Prediction, Multiple Linear Regression and Recurrent Neural Network, specifically Long Short Term Memory (LSTM). From the predicted prices, we calculated the daily returns which are then fed to our optimization algorithm. The expected output is a set of weights associated with each of the stocks in the optimal portfolio, maximizing a metric for the returns over risk. A full model schematic follows.

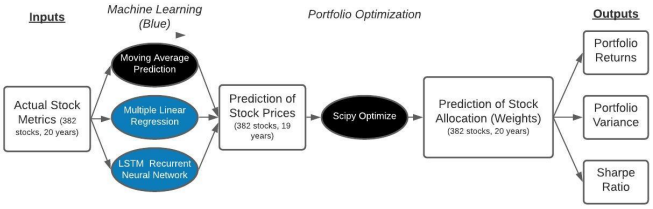

Figure 1: *Model Architecture Schematic*

2 The Dataset and Features

2.1 Choosing Stocks

Through our research, we decided to gather historical data for the constituents of the S&P 500 (as of October 28th, 2020) as there is a rigorous framework in place to ensure that companies within the index are of 'high quality.' While there is certainly bias in only considering these 505 stocks instead of the 3000+ stocks

in the New York Stock Exchange (NYSE), there are two main reasons why we chose to remain with these stocks. Firstly, there are hidden trends in the fundamental characteristics of companies, such as growth in net income, interest expenses, assets and liabilities, that are not always reflected or 'priced-in' to the stock prices that we see. Here, it is important to recognize the difference between the fundamental value and market price of a stock. In a simplistic analogy, fundamental value can be thought of as the price to produce a piece of fancy artwork while its market value would be the price that it is sold for in auction. We can notice that market value can be extraordinarily larger than fundamental value, and this was a major consideration when we chose our stocks. However, by using the stocks in the S&P 500, the rigorous framework ensures that the fundamental value of these stocks is 'high' because of their actual performance rather investor speculation driving up the market value. Thus, because we are attempting to create a portfolio of stocks that we plan to buy and hold for a long period of time, it is essential that we choose companies reputable enough to be included in the S&P 500. Secondly, limitations on memory and computational power would constitute gathering and processing 20 years' worth of data for 3000+ stocks to be nearly impossible for this project. Through further investigation, researchers may be able to make simple adaptations in order test the differences between our portfolios and one that includes all the stocks traded in the NYSE.

2.2 Gathering the Data

Financial data is widely available throughout the web, however we opted to use the QuantMod package in R to gather our data from Yahoo Finance. While other API's are able to gather daily stock data for global equities, QuantMod was able to provide us with various metrics of stock price as well as gather the data for multiple stocks in parallel. These metrics of stock price include the price that the stock opened at 9:30am E.S.T, the price that the stock closed at 4:00pm E.S.T as well as the low and high price of the stock on a specific trading day. Additionally, we were also able to gather the traded volume for each stock that day. However, the data required a bit of additional cleaning. Since all the stocks that are listed on the S&P500 have not been listed on the stock market for 20 years, we decided to drop the stocks with limited history which left us with a total of 382 stocks. Of these stocks, the top 5 best and worst performing stocks can be visualized in figure 2.

Here, we can see that some companies such as AIG (ticker: AIG) and Citigroup (ticker: C) have declined quite drastically while companies such as Apple (ticker: AAPL) and Monster Beverages (ticker: MNST) have performed very well. Additionally, we can clearly see the effect of the 2008 financial crisis on some of

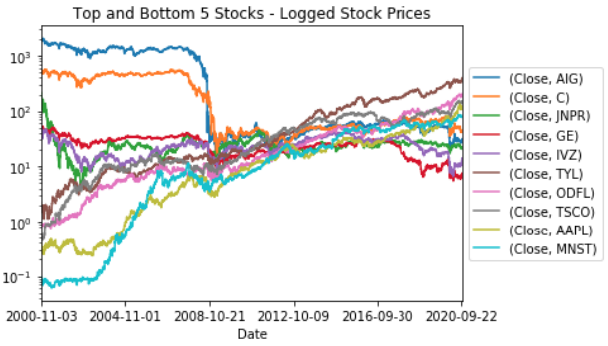

Figure 2: *Logged Stock Prices*

these stocks, and how certain companies were never able to recover to their previous levels. Yet, these lines do not portray the full story of what is going on. More interestingly, we can look at the yearly mean returns in figure 3 to see how much money we would have made if we invested in these companies 20 years ago.

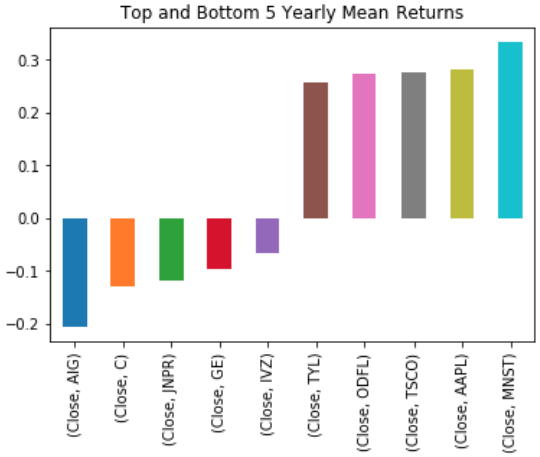

Figure 3: *YoY Mean Returns*

Now, we can more clearly see how much each of the respective companies grow on an annual basis. While certain stocks may have performed better or worse during certain years, we can see that if we invested in Monster Beverages (a subsidiary of Coca-Cola) 20 years ago and held the stock until 2020 we would have averaged growth of nearly 30%, or if we invested in AIG that we would have lost nearly 20% per year.

2.3 Feature Generation

In addition to the information for each stock in our initial dataset, which included opening, closing, low, high, and adjusted prices, as well as trading volume, we decided to calculate additional financial metrics. These figures were added to the original dataset to increase the number of features in our model, and therefore reducing the level of overall bias. These metrics include Daily Returns, 20 Day Returns, Rolling 20 day volatility, Normalized 20 Day Returns and Normalized

20 Day Volatility.

2.4 Dimensionality Reduction

In order to avoid the "Curse of Dimensionality" that a model with too many features would inevitably face, we decided to perform dimensionality reduction using Principal Component Analysis. The process compressed the initial 4207 features into 382 whilst maintaining almost all of the initial variance.

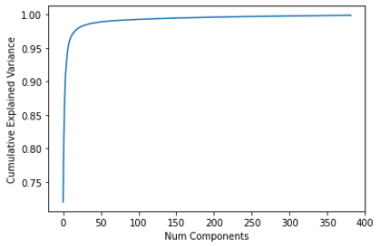

Figure 4: *Pct. of variance explained after PCA*

3 Predictive Models

3.1 Moving Average Prediction

This is a naive forecasting technique, that uses then mean of the prices in our lookback window as the prediction for the future prices. In financial data, simple moving average techniques have always worked surprisingly well due to the serial dependence that time series data tend to exhibit. Additionally, we know that stock prices follow a mean reversion pattern. Since sudden and dramatic changes in pricing are unlikely, the closing price of a stock on day 2 is likely to be relatively close to the price of the same stock on day 1 as can be visualized in figure 5. Therefore, we use the mean of the returns from time *t-n* to *t* as a predictor for how a particular stock may perform in the near future. Yet, the inherent limitation of this model is that it doesn't exploit other features such as it's own volatility or it's covariance with other stocks. The naive moving average prediction is our baseline model.

$$M_t = \frac{X_t + X_{t-1}...X_{t-n}}{n} \tag{1}$$

where M is the prediction, X is the price of stock at time t and n is the lookback window. In matrix form, this becomes:

$$M_t = \frac{X}{n} \tag{2}$$

3.2 Multiple Linear Regression

In an effort to exploit our original and generated features, we modeled multiple linear regressions to predict 30 days of future stock prices by using 30 days of

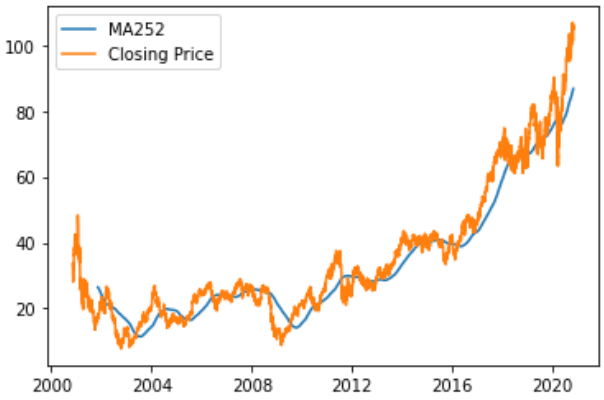

Figure 5: *Moving Average: Predicted prices against actual ones*

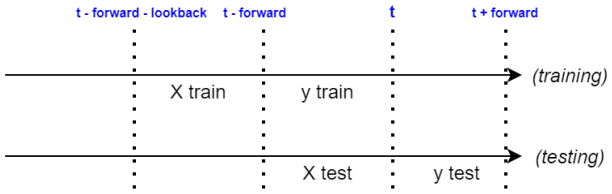

Figure 6: *Linear Regression Training Testing*

historical data from our full features dataset. In comparison to our moving average prediction, the linear regression still attempts to predict the future by determining a trend in the stock on a limited history of data, however it now has access to other information such as the performance of other stocks and the direction of the entire stock market. As we use PCA to reduce the dimensionality of the dataset, the line can be modeled as follows:

$$\underset{\text{(Lookback, Principal Components)}}{PC} \quad \bar{w} = SP \tag{3}$$

$$PC_1 w_1 + PC_2 w_2 ... PC_n w_n = StockPrice_{t+n} \tag{4}$$

The input matrix required to obtain *forward* days of predicted stock prices requires a matrix of shape m = *lookback (LB)* by n = *principal components*. Our training and test set split at each point, is visualized in figure 6. The model is fed data from a time window in the past (Xs), and maps those to y values in a later time window. The problem has now become of supervised nature. When learning the weights, we had to iterate over the length of the entire dataset in 30 day intervals for each stock to train and test the data. Additionally, in order to prevent data leakage, we performed PCA independantly on our training and test sets. Though this model is still quite naive, it serves as an example to see whether the added complexity of including additional features improves returns in our portfolio optimization.

Below follow some plots of predictions versus actual returns for stock "A", Aggilian technology. unnamed (1)

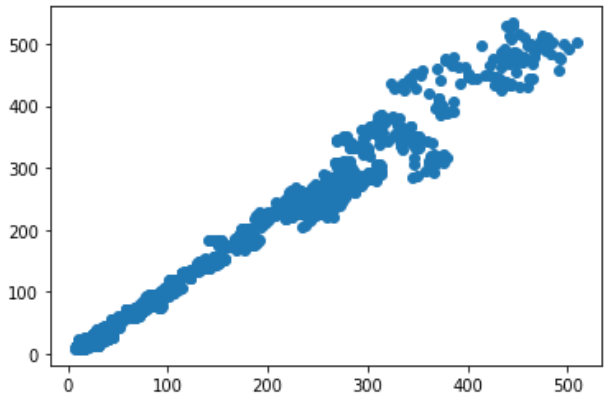

Figure 7: *QQ plot: Prediction on the Y-axis, actual prices on the X-axis*

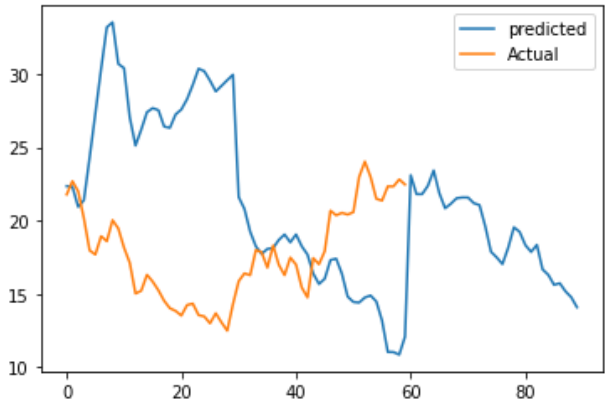

Figure 8: *Predictions against actual prices (short time frame)*

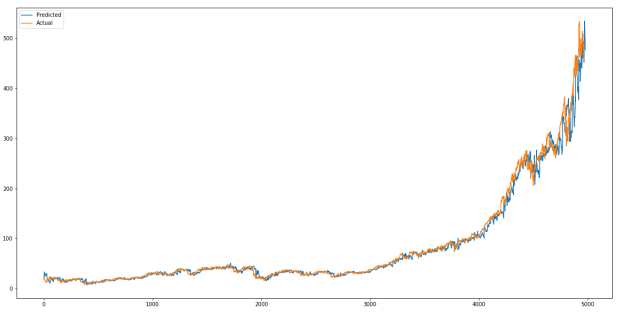

Figure 9: *Predictions against actual prices, (long time frame)*

3.3 Long Short Term Memory (LSTM)

The LSTM network is by far the most complex model that we utilize. Similar to the linear regression, this model also uses the full features dataset that was compressed by PCA to predict future stock prices. However, the key benefit to this model is that it is able to apply patterns recognized in the past to future predictions. The specific type of RNN is a improvement to vanilla RNN, adressing the problem of diminishing gradient. It passes on new information as it propagates forward. Each cell in the LSTM model has internal gates that regulate the flow of information, determining which of pieces of information are of importance to the predictions and hence which to keep. The hidden state acts as the NN's memory. The forget gate is fed the previous hidden state, combined with the current input (X) and passes these through a sigmoid function. The output, always in range [0,1] determines how much information to forget and how much to keep respectively. The input gate, also fed the hidden state and current input, determines how much information is passed on to the new cell state. Lastly, the new cell state passes through a tanh function and is combined with the current input and the previous hidden state (which have already been passed through a sigmoid function). This last output becomes the next hidden state to be used as an input for the next cell (Markowitz, 1991). A schematic of the internal cell architecture follows.

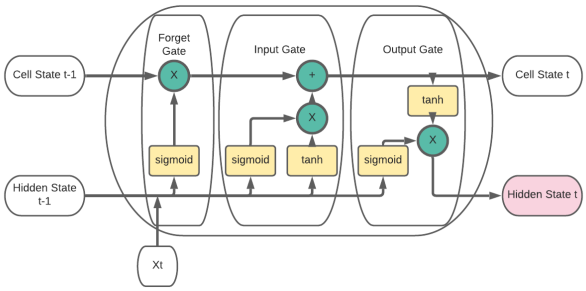

Figure 10: *Internal LSTM cell architecture*

We used a "stateful" approach to formulating our X and y labels in which there was no overlap in the dates (Arnold et al., 1998). In terms of the network architecture, we found three potential formulations with regards to number of layers to use in the network. (Zou and Qu, 2020) pose that two LSTM layers are too complex for stock price predictions, however we experimented with the depth of the network and concluded that a double layer LSTM performs significantly better. A schematic of the final model follows

Min-Max scaling was used as it does not assume normallity, scaling the data between a range of [0,1] (Zou and Qu, 2020). The final model inputs a window of historical stock prices and outputs predictions for future stock prices, scaled. We then inverse scaled the output in order to return the data into its original scale

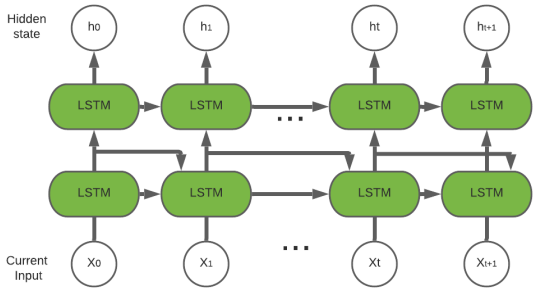

Figure 11: *Double layer LSTM RNN architecture*

and prepare it for the next section of the model. The final MSE reached for the model was 1.9044e-04. A graph of the prediction achieved by the LSTM against the actual prices for the Apple's stock "AAPL" follows.

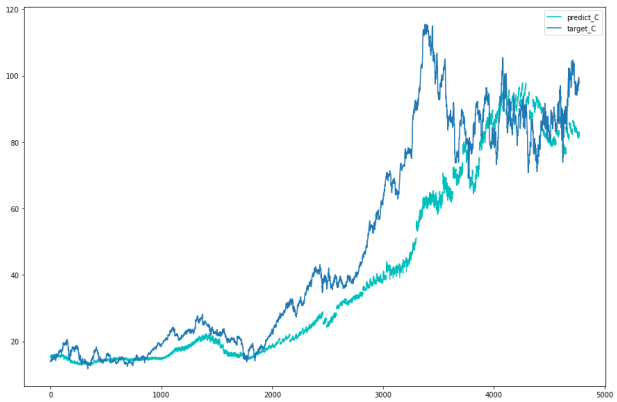

Figure 12: *LSTM predicted prices against actual ones*

4 Portfolio Optimization

4.1 Predicted Stock Returns

Next, we calculated the log returns for both the actual and predicted stock prices. While we could have calculated the simple returns, which is also financial measure of how much a stock increases or decreases as a percentage, log returns have some beneficial features. First, as stock returns are log normally distributed, and generally experience a greater frequency of positive returns rather than negative returns, the distribution is heavily right-skewed. Thus, by logging the returns, the distribution becomes normally distributed and properly scaled for the model. Second, cumulative returns for a specified time period are additive which makes it computationally simpler to present the returns for different periods of time.

4.2 Loss Function Derivation

The tradeoff between portfolio optimality and simplicity is a problem of scarcity. Well performing yet simple portfolios are preferred, as investors consider trading and transaction costs as well as the time-consuming nature of these procedures. Therefore penalizing the model for increased complexity seemed to be a proper approach in generating optimality beyond pure theory. We decided to use LASSO (L1), aiding in variable selection and parameter shrinkage, while sparcifying the outputted weights. Hence, the investor is encouraged to invest more in smaller number of stocks (David Puelz and Carvalho, 2016). The optimization equation can be expressed as:

$$\begin{aligned} &\max_{w} \mu^T \mathbf{w}^T - \lambda_1(\mathbf{w}^T C \mathbf{w}) + \lambda_2 ||\mathbf{w}||_1 \\ &\text{s.t.} \quad w1^T = 1 \\ &\qquad w\mu^T = Rp \end{aligned} \tag{5}$$

In order to maximize this portfolio, the input will be the predicted logarithmic returns for the next 22 tradng days (aprox. one calendar motnh). The model will then output weights for our portfolio to be held for the same time window. After the 22 trading days have passed, we repeat the process and rebalance the portfolio with the new outputted weights. We will be ignoring transaction costs.

5 Results

5.1 Stock Price Prediction

When training our programs, we expected the LSTM to perform the best in predicting the stock price; followed by the linear regression. However, in figure 16, we can see that the MSE for our predictions on the entire dataset for the LSTM was significantly larger than that of the linear regression. While this may seem like a large issue, what is of interest is not the predicted stock prices are from the actual stock prices, but the direction of the returns. Thus, in the following portfolio optimization section, we are able to use our stock price predictions to calculate relevant metrics, such as the portfolio returns and Sharpe ratio, to see the accuracy of the machine learning models.

5.2 Portfolio Optimization

Due to computational power constraints, we decided to only train our portfolio for a period of 5 years, starting from September 5th, 2014 to August 14th, 2019.

We will now compare our portfolios with common financial metrics like the Annualized Sharpe Ratio, Annualized Volatility, and Final Equity Value. These values are listed in Table 1 and give a summary of the risk and return tradeoff of our optimized portfolios. As we can see, volatility (defined as standard deviation of portfolio returns) is close to zero for all three different portfolios, showing a low risk profile for all these asset allocations. With such a low volatility, finance theory tells us we should expect low returns and this

is shown in the Sharpe Ratio and Equity Value. In the 5 year holding period, our top performing portfolio achieves a cummulative return of 16.5%. Typically, a Sharpe Ratio over 1 is a sign of a strong and rewarding trading statregy, but our trained portfolios failed to come close to such figure. With a volatility as low as the one present in our portfolios, we expected a higher Sharpe Ratio that was not attained. Comparing our strategy to a similar risk profile asset, like the US T-Bills, our returns are high for the given risk level. A US T-Bill at the time averaged a return close to 1% while pur best models significantly outperformed this simple benchmark. Next, we plotted an Equity graph to allows us to see the performance of our portfolios across time. With an equity graph, one is able to see how the portfolio value changes assuming an initial investment of $100. We conviently choose 100, as it allows percentage-wise interpretation. As we can see from Figure 13, the equity value of our LSTM portfolio is significantly greater than the base models for most of the holding period. For the last year, this pattern breaks and the Moving Average model ends up on top. Overall, for the 5 year holding period, our top performing portfolio is the Moving Average Portfolio, the second best is the LSTM based portfolio, and the worst performer by a wide margin is Linear Regression.

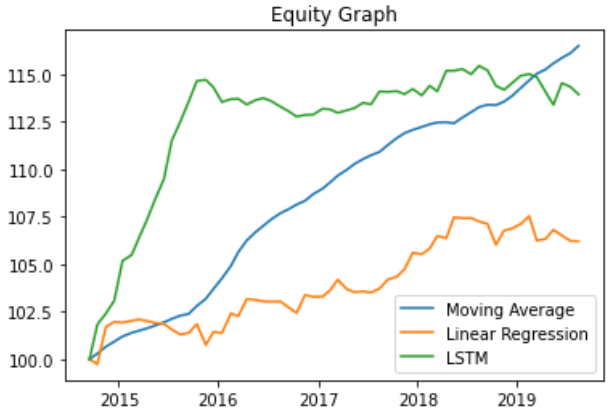

Figure 13: *Equity plot for each of the 3 models*

6 Conclusions and Future Work

The most significant improvement is expected to come from cross validating the dataset. The model depends on a plethora of hyperparameters, which were tuned using heuristics, but it is expected that through cross validation we can improve these. We noticed our LSTM model for predicting returns faced a significant bias. Hence, one approach to improve on the current model would be applying further feature engineering techniques such as Sentiment Analysis on stock reporting using Natural Language Processing or trend approximations using Fourier Transformations. Additionally, we would consider other techniques for dimensionality reduction. Non-linear techniques such as tSNE, UMAP and Autoencoders are expected to outperform PCA. Furthermore, with more computational power available, we could have not only optimized our portfolio for the entire 20 years of data, but also added more stocks to the dataset. Our portfolio is currently optimized with monthly logarithmic returns, and a daily returns approach would yiled more values and amplify our ability to understand the model and its limitations. Finally, in order to improve the comparability of the results, we could optimize for the two fundamental Markowitz portfolios: Maximum Sharpe Ratio (tangency portfolio), and Minimum Variance Portfolio. These last two portfolios are quadratic programming problems and solving them with the number of variables we had was computationally unfeasible.

Effectiveness of Different Models				
Model	Annualized Sharpe Ratio	Annualized Volatility	Ending Equity	Stock Prediction MSE
Moving Average	0.116	8.82E-05	116.5	NA
Linear Regression	0.014	3.19E-04	106.2	187.4
LSTM	0.024	3.68E-04	113.5	8952.9

Figure 14: *Model Effectiveness*

Bibliography

Arnold, A. S. et al. (1998). “A Simple Extended-Cavity Diode Laser”. In: *Review of Scientific Instruments* 69.3, pp. 1236–1239. URL: http://link.aip.org/link/?RSI/69/1236/1.

David Puelz, P. Richard Hahn and Carlos M. Carvalho (2016). “SPARSE MEAN-VARIANCE PORTFOLIOS: A PENALIZED UTILITY APPROACH”. In: *Annals of Applied Statistics*. URL: https://arxiv.org/pdf/1512.02310.pdf.

Markowitz, Harry M (1991). “Foundations of Portfolio Theory.” In: *The Journal of Finance, vol. 46, no. 2,, pp. 469477*. URL: www.jstor.org/stable/2328831.

Zou, Zichao and Zihao Qu (2020). “Using LSTM in Stock prediction and Quantitative Trading”. In: *CS230: Deep Learning*. URL: http://cs230.stanford.edu/projects_winter_2020/reports/32066186.pdf.