DRAWL Trading: Final Project Report

Combining Machine Learning models with proven economic principles (i.e., Modern Portfolio Theory) to construct a realistic stock market environment and provide profitable trading suggestions to investors.

Drew Peterson, **R**yan Middlemiss, **A**drian Lopez, **W**ill Long, **L**ucas Josephy

Duke University: ECE 496

April 25, 2022

Table of Contents

I. Problem Overview

A. Problem Background

1. Stock market analysis is a very lucrative industry that can be broken down into two main schools of thought: fundamental analysis and technical analysis. Fundamental analysis utilizes a company's underlying financial performance and intrinsic value in order to predict stock price movements for that company. Technical analysis utilizes historical data in order to recognize stock market price signals and patterns and predict future prices and volume. Inherently, stock market volatility is largely based on consumer sentiment. Thus, it is very difficult to implement regression models capable of making accurate predictions in a volatile, emotion-driven environment. It is also increasingly difficult to determine a direct correlation between volatility and the movement of stock prices (i.e., high volatility may indicate rampant up and/or down swings). There have been many attempts at accurately predicting movements in stock prices, which have made extensive use of recurrent neural networks and historical stock price data. Our goal is to employ powerful predictive models, with accurate variance estimates, to construct an optimal portfolio capable of responding accurately and profitably to movements in stock price.

B. Problem Solution

1. The solution to our stock portfolio allocation problem is the creation of an optimized portfolio tool, invoking Market Portfolio Theory, and capable of allocating funds within a portfolio based on an investor's capital restraints. The solution includes a frontend, user interface that will allow the investor to seamlessly construct their own personalized portfolio and determine their potential output/payoff. We optimized 2 LSTMs to predict both the future stock price and the variance associated with our prediction to be able to power our Optimized Portfolio Allocation Tool to make more accurate predictions. Specifically, our Optimized Portfolio Allocation tool uses these predictions to allocate capital in such a way such that the Sharpe Ratio, a measure of a portfolio's return relative to a risk-free investment adjusted for volatility, is maximized. We generate fresh results after the passing of each trading day to ensure that our users have the most up to date information by querying Alpha Vantage and rerunning the predictions for our LSTMs and then calculating the optimal portfolio using our Optimal Portfolio Allocation Tool.

II. Project Components/Features

A. Stock Price Prediction Tool

Training Procedure: We meticulously trained 30 LSTMs, one for each asset in the DOW Jones, using 2 different training procedures. The first training procedure consisted of using 475 days of exclusive hourly data provided by AlphaVantage with a hyperparameter grid search that took over 2 weeks to complete. Our models received as input a 7 dimensional tensor corresponding to the 7 hours in a trading day, and as an output, we had it predict the stock prices at those same 7 hours for the following day. We aimed to maximize these models' predictive performance in predicting the following day's stock price, and then with this, aimed to use a recursive approach to predict the stock price for any desired amount of days in the future by inputting the prediction of the following day back as input to the model. The validation dataset consisted of 45 days worth of data to prevent overfitting, and the weights that performed best on this dataset were chosen for our final models. For the second training procedure, we ignored the use of the validation dataset, and instead trained the model on 520 days worth of hourly stock market data, using the same grid search, and selecting the weights which minimized the training loss.

The results of the validation MSE corresponding to the best weights of the 30 LSTMs trained with a validation dataset are shown in Figure 1 below:

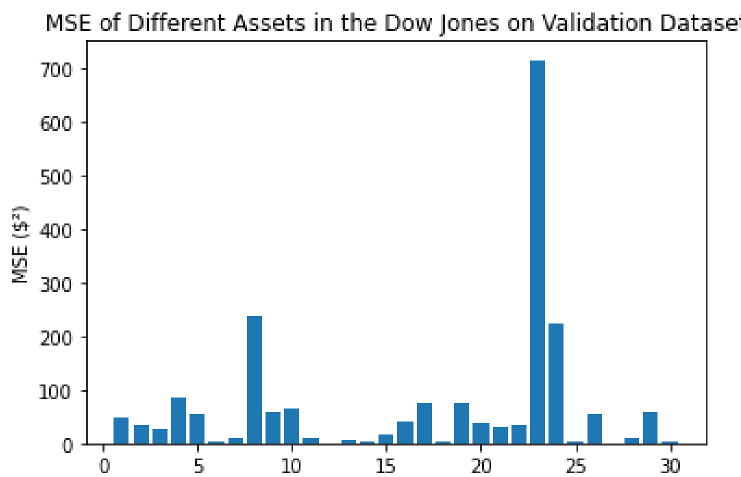


Figure 1: MSE of the 30 Securities in the Dow Jones on LSTMs trained with a validation dataset

As can be observed in Figure 1, there is a huge outlier in the validation MSE for the 23rd security in the DOW Jones referring to UnitedHealth Group Inc (UNH). We aimed to understand the source of this variance in more depth and looked at the stock price evolution of (UNH) during the dates of (Feb 1, 2022 - Mar 16, 2002), which is shown in Figure 2 below:

Figure 2: UnitedHealth Group, Inc (UNH) Stock Price Movement

As one can see from Figure 2, the variance of the stock price was extraordinarily high during this period. This realization sheds light as to why our model struggled to accurately generate predictions since choosing optimal weights for this time must have been extremely difficult and price movements likely included forces outside of simply previous stock price (i.e., forces not captured in our training dataset). To be able to quantify risk for our portfolio allocation tool for more investors, we aimed to develop a way to accurately measure volatility. To do so, we trained 30 additional LSTMs, again one for each asset with a similar training procedure, which instead aimed to learn the mean squared error (MSE) between the prediction of the LSTMs and the actual stock prices. For an unbiased estimator, the MSE is equal to the variance, and as such our goal was to use this MSE to learn the "variance" of our predictions. To show the predictive power in being able to accurately guess the MSE, we displayed the square root of the MSE of the model which aims to predict the MSE of the first model in Figure 3 below:

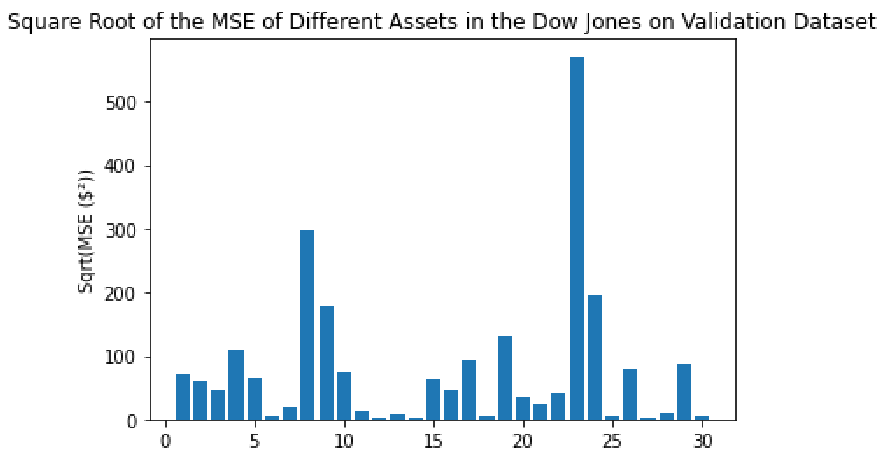


Figure 3: Square root of the MSE of the MSE estimator of the price predictor LSTMs (w/ validation)

As expected, UNH shows a high amount of volatility. However, both stock price predictions as well as variance predictions looked unusually high. We conclude that this is due to the extreme volatile movement of the stock price in the validation dataset for some assets, leading to non-optimal weights being chosen and early stopping in the training procedure. We found that some of the models stopped training after just 80 epochs with the validation dataset, whereas for the models trained without the validation dataset, almost all went to the full 7k epoch training procedure. We hypothesize that having a larger validation dataset may improve results since the deviation in price in the long run would be smoother than as shown in Figure 2, for example. We then decided to test LSTMs trying to minimize the MSE using the entire training dataset. The MSE of the stock price predictor and the root mean squared error (RMSE) of the variance predictor is shown in Figures 4 and 5 below.

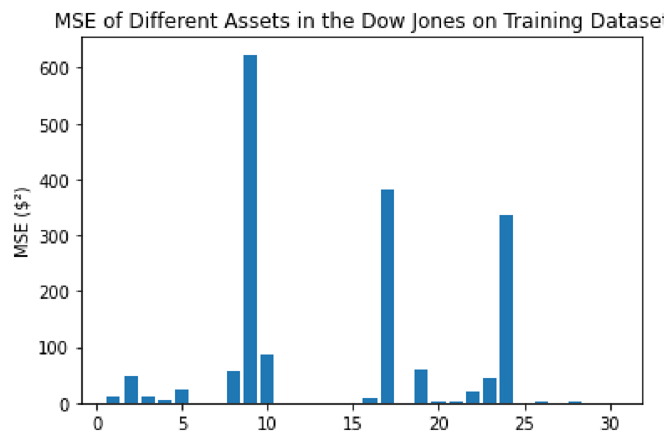


Figure 4: MSE of the 30 Securities in the Dow Jones on LSTMs trained without validation dataset

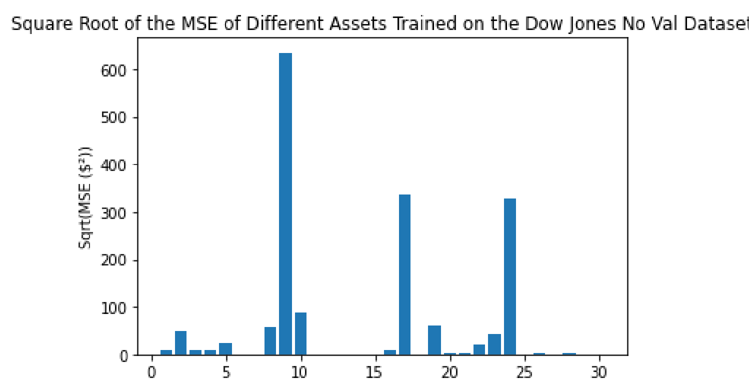


Figure 5: Square root of the MSE of the MSE estimator of the price predictor LSTMs (w/o validation)

As one can see on Figure 4, the model had difficulty predicting the 9th security, which corresponded to Home Depot, Inc. We then analyzed the stock price of Home Depot over the past year on Figure 6 and as expected, it had tremendous variance, which likely led to the high MSE value.

Figure 6: Home Depot, Inc (HD) Stock Price Movement

Note that the MSE values in Figures 4 and 5 in general are better than those of LSTMs trained with a validation dataset. We aimed to compare the predictions made in the following 30 days to get a visual sense of their difference and then to test them on a testing dataset to evaluate their performance and to choose a model.

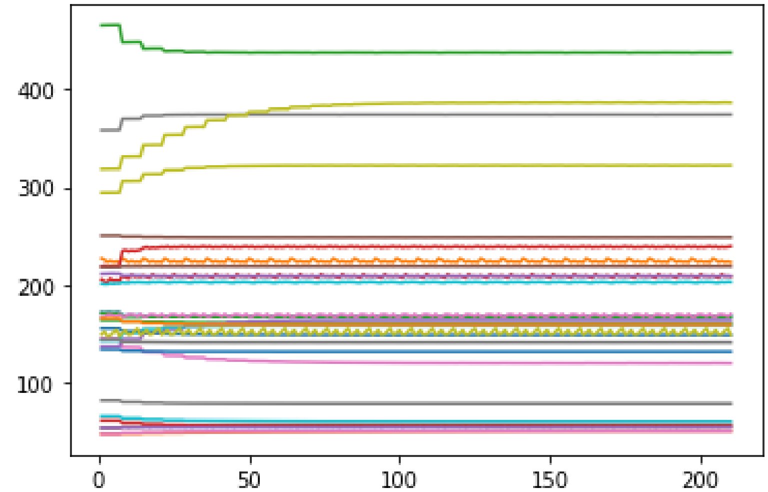


Figure 7: Dow Jones Stock Price Predictions for Next 30 Days for LSTMs trained with a validation dataset (210 trading hours).

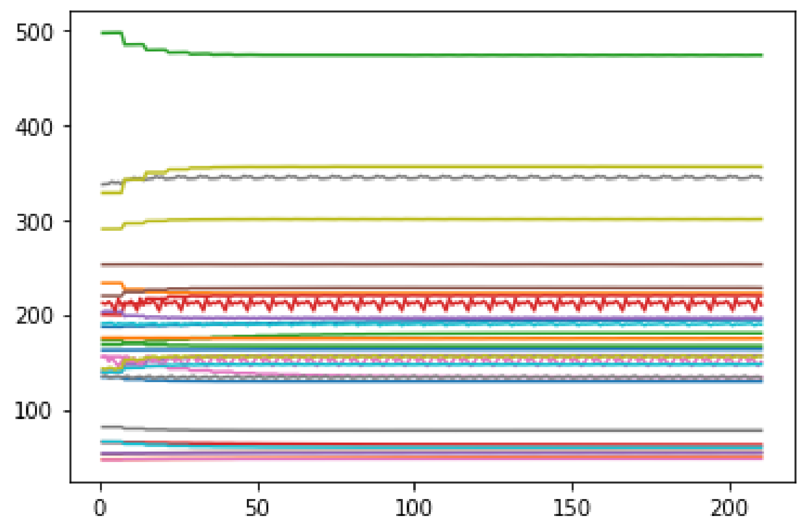


Figure 8: Dow Jones Stock Price Predictions for Next 30 Days for LSTMs trained with no validation dataset (210 trading hours).

Testing the Models: Our training dataset covered a 2 year span which ended on March 16, 2022. To test the model, we gathered data for all 30 stocks on the DOW Jones index from the first trading day immediately following March 16, 2022 to the most recent trading day (April 22, 2022) which corresponded to 26 total trading days. We then evaluated the predictive metrics of the networks trained above on this dataset. Our stock price predictor LSTM trained using a validation dataset had the following error (predicted - actual) for the following 30 securities shown below over the 26 trading days.

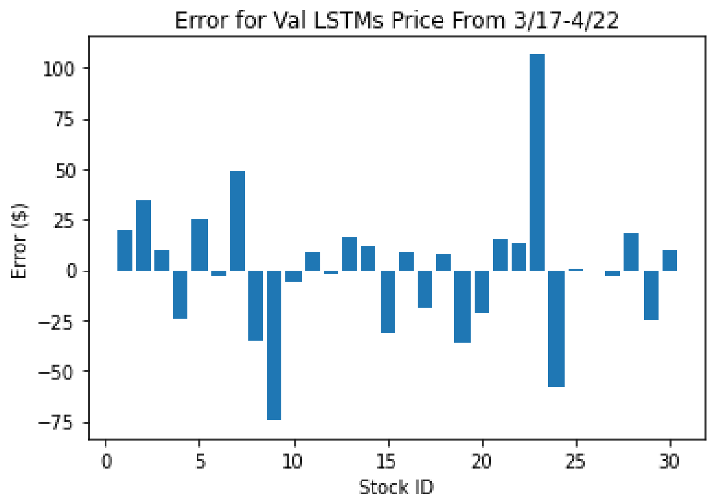


Figure 9: Error for LSTMs trained with a Validation Set

The mean squared error across the 30 stock predictions was 1070.46 $\$^2$. As you can see, the model performed very poorly, especially for securities such as UnitedHealthGroup, Inc. for which the validation data had extremely high volatility which in turn severely impacted the model's performance since nonoptimal weights were chosen.

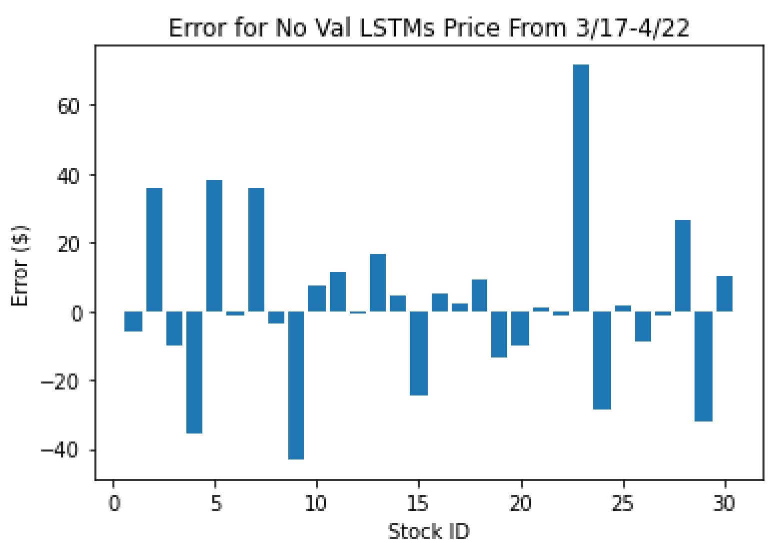


Figure 10: Error for LSTMs trained without a Validation Set

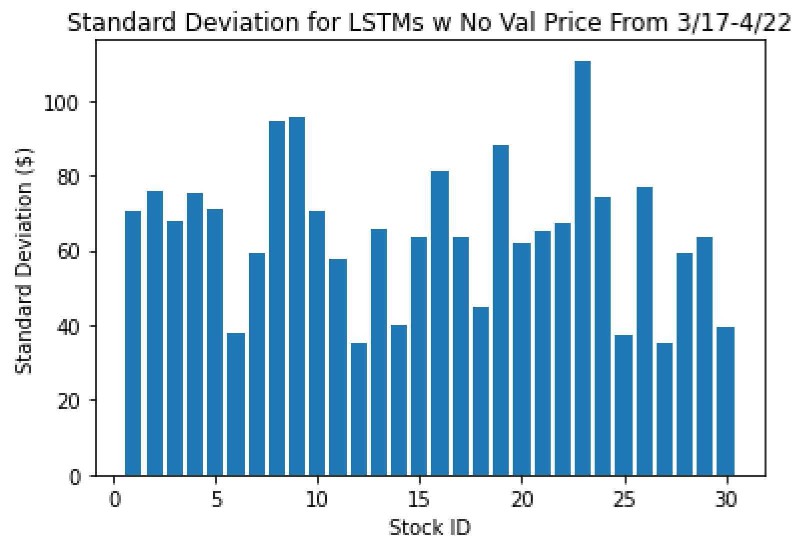


Figure 11: Predicted Standard Deviation for the 30 assets by our Variance Predictor Network

The mean squared error across the 30 stock predictions was 555.16 $\$^2$ which is about half of that for LSTMs trained with validation datasets and as such we decided to use these networks going forward. As shown in Figures 10 and 11, we found that the error always stayed under 1 standard deviation away with our variance predictor network and thus were pleased with its performance.

B. Optimized Portfolio Allocation Tool - Using Modern Portfolio Theory (MPT), we took data relating to the 30 stocks on the Dow Jones Industrial Average (hereafter referred to as the Dow) and built a tool to construct a portfolio which maximizes investor's expected returns. First, let us go over at a high level how MPT allocates wealth. Given a set of securities, MPT takes the expected return and variance of each security and outputs a portfolio which maximizes expected return for any given level of risk.

Our wealth allocation tool utilizes this theory by first calculating the daily expected return of each security in the Dow by taking in price data and then dividing each individual price by the subsequent price. This produces an array of percent changes in prices. Next we take the average over this array in order to calculate the average daily return for each stock, which is organized into a vector v where v[i] is the expected daily return for stock i.

Variance for the entire portfolio is calculated by first finding the covariance between every pair of stocks in the Dow. These values are organized into a covariance matrix M where M[i, j] is defined to be the covariance between stock i and stock j. Then a vector of weights w where w[i] is the percentage of wealth allocated to stock i is multiplied on both the left and right side of M. That is,

$$\sigma^2 = wMw^T$$

where σ is variance. Notice that the variance of the entire portfolio cannot be known until the weights, or the percentage of wealth allocated to each security, are known in full. This is where we must begin the optimization process.

As stated previously, our tool aims to create a portfolio which maximizes expected return given a certain level of volatility. Thus, we can even allow the user of such a tool to input their own desired level of risk. This is done my minimizing the following equation[1]:

$$\sigma^2 - r\mu$$

Where r is the user-inputted risk tolerance and μ is the expected return of the entire portfolio, calculated as follows:

$$\mu = wv$$

Below, a graph is shown of a number of different portfolios given a level of risk generated by our optimization tool.

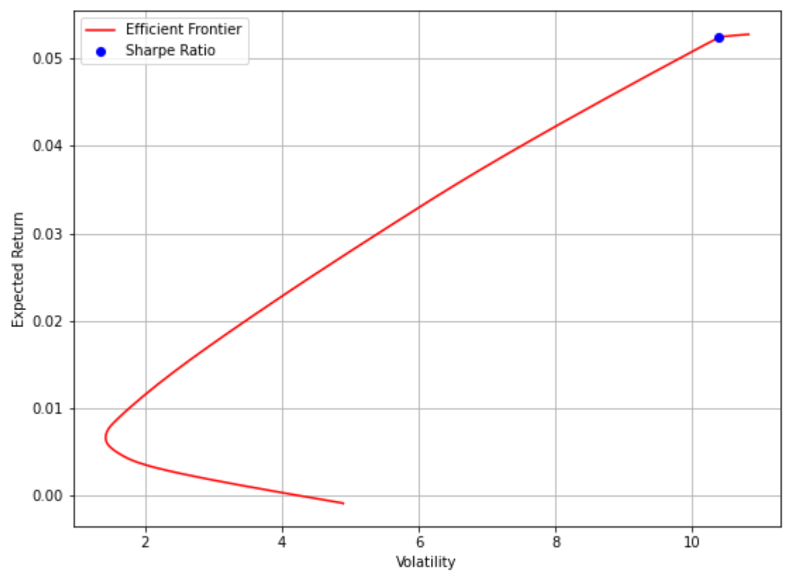


Figure 11: Efficient Frontier generated by our Optimization Tool

To generate this graph, past data spanning roughly a year and a half and ending in mid-March of 2022 was used in order to create many different optimized portfolios for a number of different volatility measures. This figure is known in financial Economics as the Markowitz Bullet, named for Harry Markowitz, the founder of MPT.[2]

[1] Lysenko, Ivan. “Modern Portfolio Theory Model Implementation in Python.” Medium. Analytics Vidhya, March 24, 2021. https://medium.com/analytics-vidhya/modern-portfolio-theory-model-implementation-in-python-e416facabf46.
[2] Kenton, Will. “Who Is Harry Markowitz?” Investopedia. Investopedia, March 14, 2022. https://www.investopedia.com/terms/h/harrymarkowitz.asp.

Another feature of MPT is the ability to find a portfolio which maximizes the Sharpe Ratio, which indicates the return of the portfolio as compared to the risk-free rate adjusted for the portfolio's volatility. In our project, this risk-free rate was taken to be the return of US Treasury-issued bonds, which we measured at 1.67%. The Sharpe ratio is therefore calculated using the following equation:

$$(\mu - 0.0167) \div \sigma$$

In order to find the portfolio which maximizes this ratio, we simply minimized the negative of this equation. The portfolios which minimize this ratio are the ones displayed in this report and in our final product. It is also worth mentioning that all optimizations were done using Scipy's minimize function.

In order to test the functionality of our tool, we decided to find the portfolio which maximizes the Sharpe Ratio using the same data set as mentioned previously to create the Markowitz bullet graph. Then, we tested how this portfolio would do over the next month, with trading ending in mid-April. Below are our results.

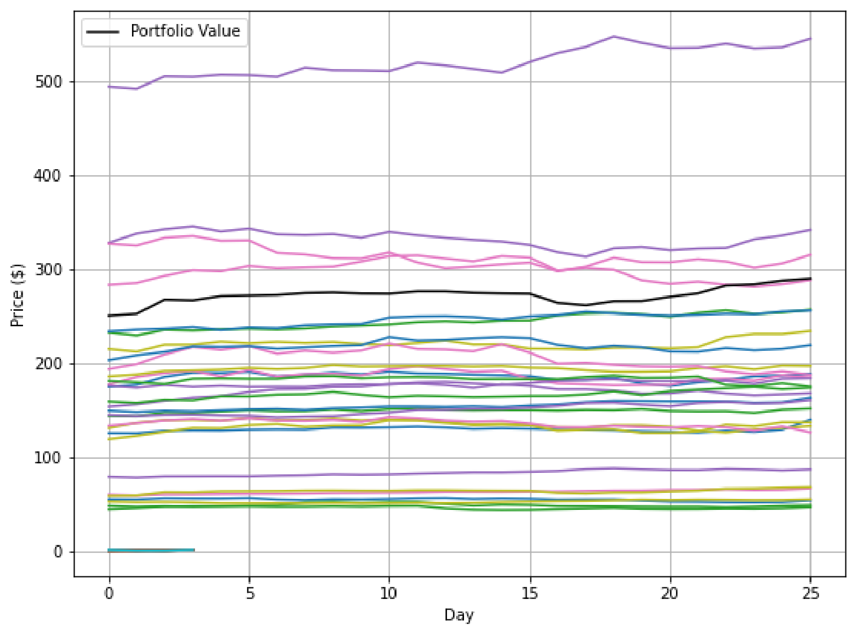


Figure 12: Predicted Stock Price Movement and Sample Portfolio Allocation (Initial investor capital of $250)

In this example, we used a portfolio constructed of $250 of wealth allocated to all 30 stocks comprising the Dow. Depicted are the trends of each security superimposed with the trend of our portfolio value (in black). One can see that our portfolio generates a positive return mirroring that of the most positively trending securities.

Our final goal with this portfolio optimization tool was to test the return of a portfolio constructed using predicted values over the entire trading period (the period of time over which the above graph spans) generated from the LSTMs described in part A. As stated previously, the portfolio depicted above was constructed using past data for future trading (using over a year's worth of data ending in March 2022 to trade during a period spanning March and April of 2022). That is, past data was used to construct the expected returns for each security and the covariance between every pair of securities (v and M as depicted in the above equations). We aim to test the portfolio which maximizes the Sharpe Ratio constructed from LSTM-generated price data spanning the trading period. That is, in this experimental portfolio, the expected returns for each security and the covariance between every pair is calculated using generated future price data. The portfolio using past data will serve as a good benchmark with which to compare this experimental portfolio. The two trends of each portfolio are depicted below (each portfolio was constructed using an initial $250):

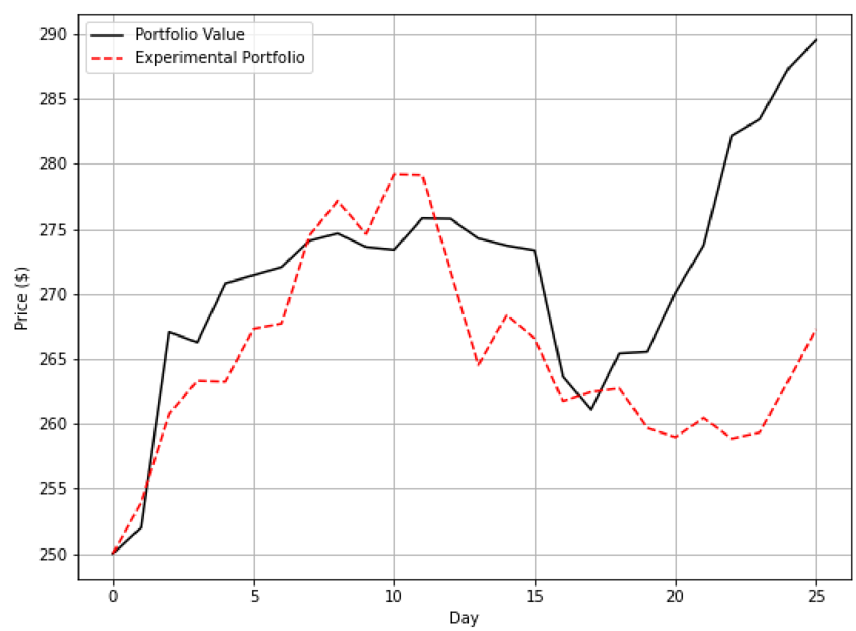


Figure 13: Portfolios constructed from historical stock data (black) and predicted stock data (red)

As depicted, both portfolios follow the same general trends, except at the last few trading days where the portfolio based on past data climbs much more drastically than that based on future data. This makes sense however, considering much less volatility was seen when the LSTMs started generating prices further and further from the present as seen in part A. However, it is very apparent that both make good investments.

Finally, the price trends of each security over both the past and future trading period are depcited. Keep in mind that the trading period corresponds to a small window centered around the 500 day mark.

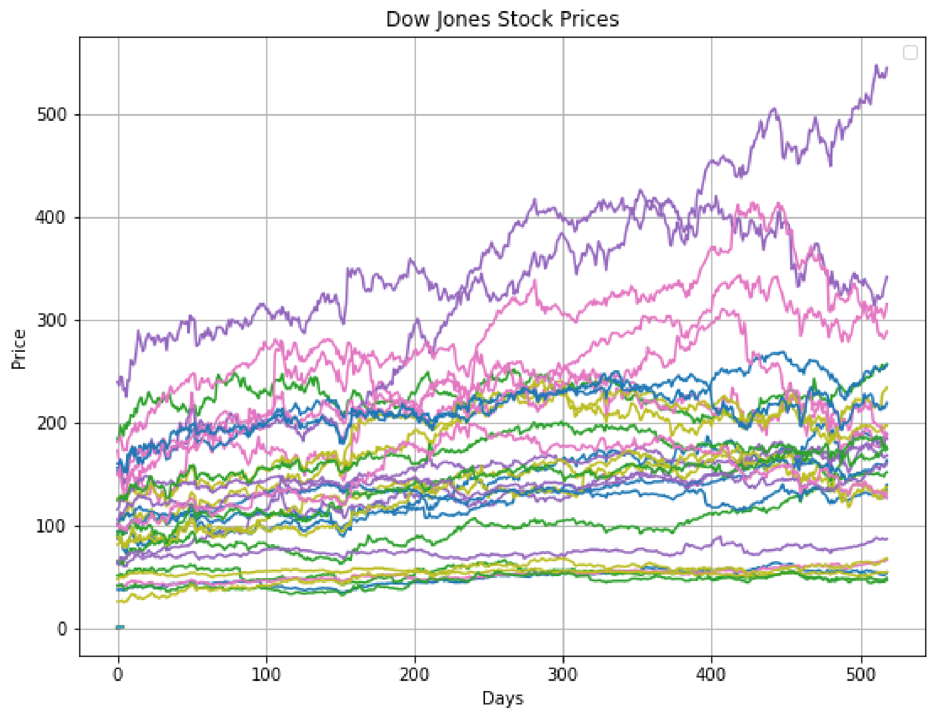
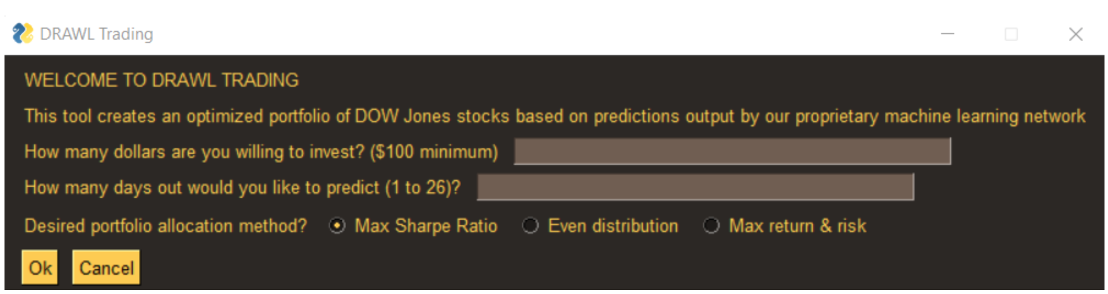


Figure 14: Actual Dow Jones stock price data pulled from Alpha Vantage

C. Application Frontend - The purpose of the frontend graphical interface is to provide the user a way to interact with the output of the other two components of this project. The script initially downloads the output from our generative stock price prediction tool, as well as the data used to train and test it. The user interface itself is implemented using the PySimpleGUI package and it queries the user for their desired investment amount, timeline, and allocation method as seen below.

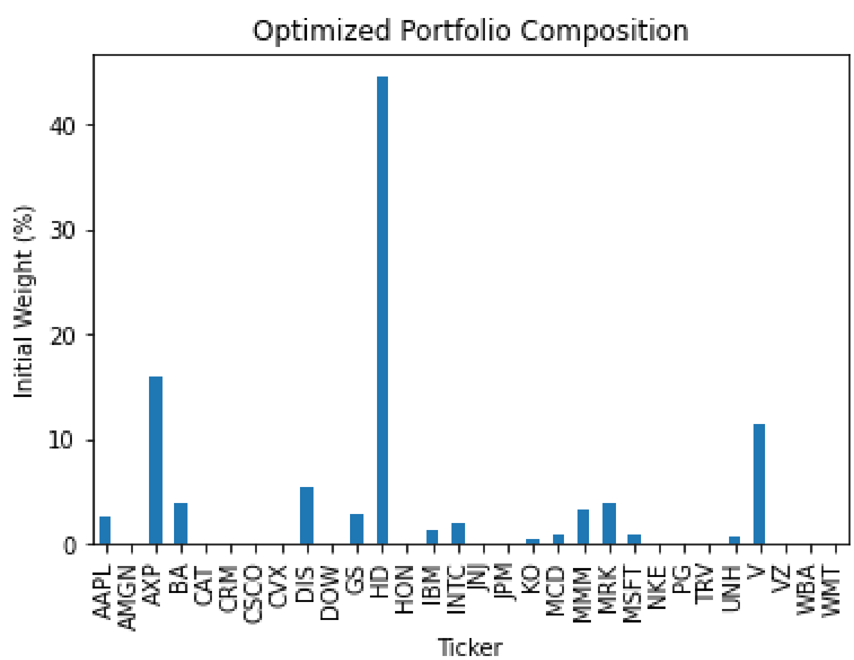


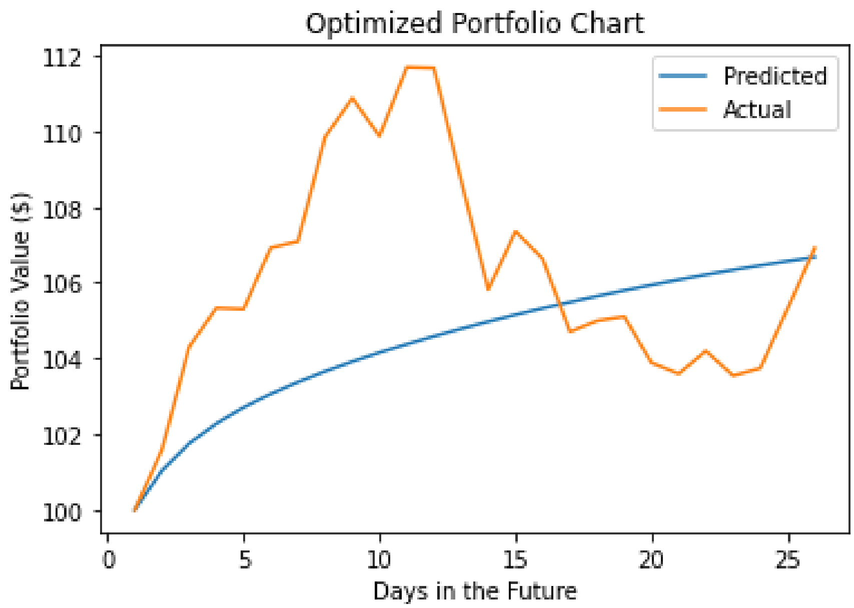
The dollar amount is simply used as the superficial starting value for the portfolio. The number of days out is used to determine over what period of time the portfolio should be optimized, and it is limited to 26 days as the generative network was trained on data ending on March 16th, 2022. At the time of this demo, there were exactly 26 trading days since that date. Finally, the allocation method gives the user a further customization option: "Max Sharpe Ratio" tells the optimizer to output the portfolio with the greatest Sharpe Ratio, "Even distribution" results in a portfolio where each of the 30 stocks in the DOW are given equal weights, and "Max return & risk" simply outputs a portfolio where 100% of the funds are allocated to the stock predicted to have the greatest return over the given period. These parameters are passed directly to an object of the portfolio optimizer class, and the resulting output is then displayed. This output is described below in the final demonstration section.

III. Final Demonstration

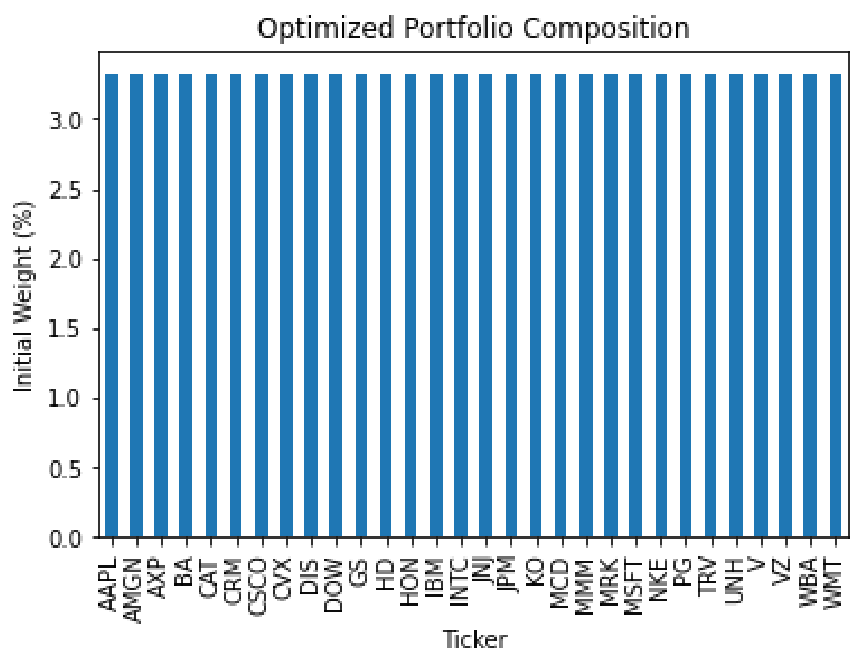
A. Description - Our final demonstration consists of a walkthrough of our stock portfolio allocation tool. The demonstration starts by querying the user for desired investment amount, timeline, and allocation method before displaying a customized portfolio based on our machine learning model predictions and optimization algorithms. The first part of the display is a plot of our optimized portfolio's predicted performance overlaid with our optimized portfolio's actual performance over the specified time period. The second part is a bar chart that depicts the initial weights, or percentage of the specified investment amount, that are allocated to each stock in the optimized portfolio. Finally, the exact weights as well as expected and actual return are output in the console in case the user wants to see concrete numbers. Three possible scenarios are shown below:

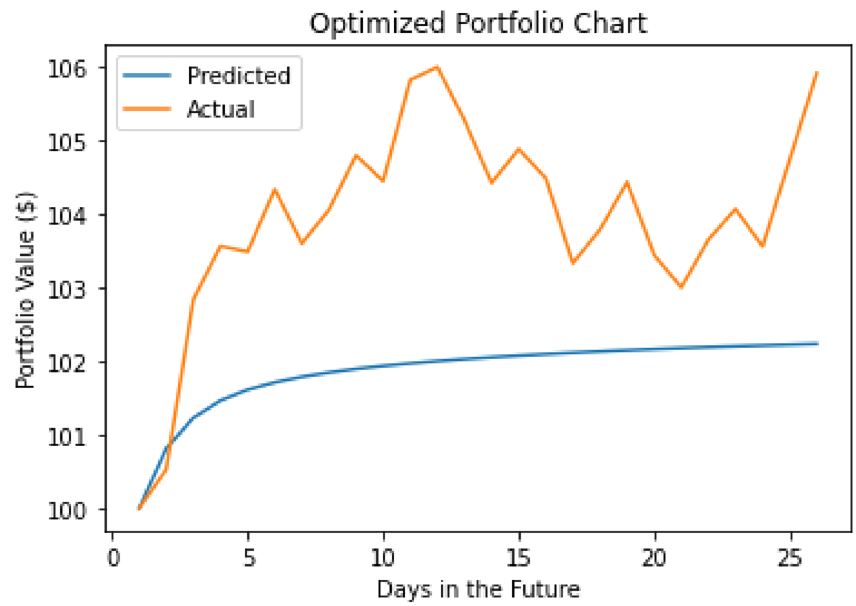
   1. *Max Sharpe Ratio, 26 days, $100*



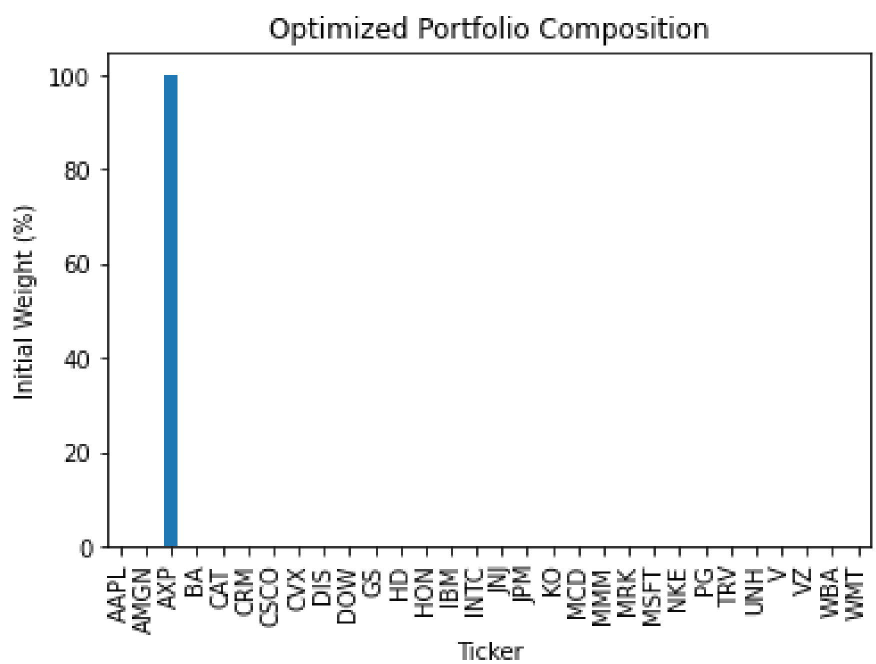


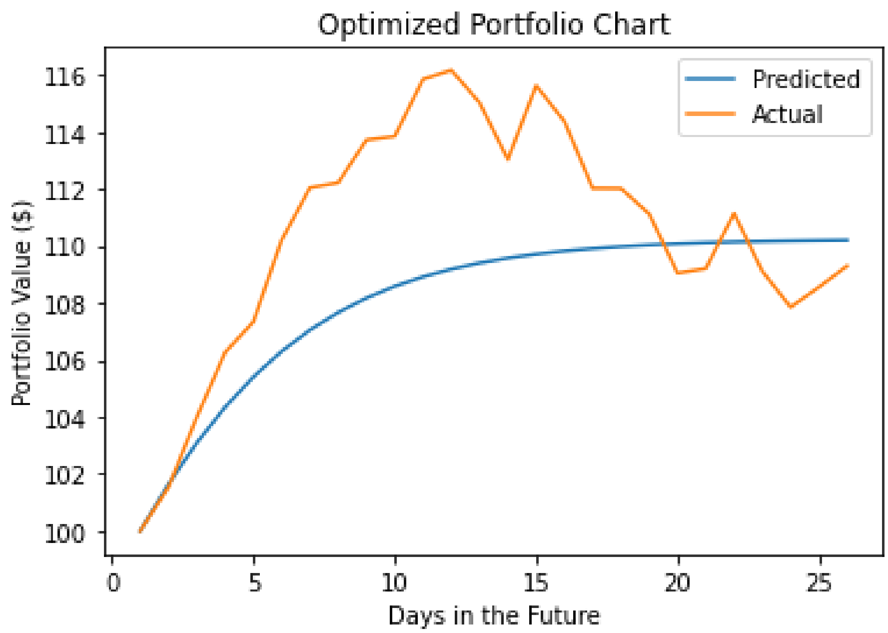
2. *Even distribution, 26 days, $100*





3. *Max return & risk, 26 days, $100*





As can be seen in the plots above, over the specified time period, the max Sharpe Ratio portfolio performed better than the evenly distributed portfolio but worse than the more risky all-in portfolio. This should be expected, and demonstrates the effectiveness of our optimization algorithm.

IV. Modern Design Considerations

A. Societal Impact

1. The primary societal impact from our project comes in the form of the economic well-being of our potential investors. By utilizing our portfolio allocation tool, these investors are entrusting us with sound recommendations for their capital in hopes of realizing a positive gain. Although these investors are afforded the opportunity to indicate their risk appetite by means of our friendly user interface, there is inherent "systematic" risk in the market that is not always considered. Systematic risk, also known as "undiversifiable risk", "volatility" or plainly "market risk", is unpredictable and difficult to avoid. On the other hand, there is unsystematic risk which is specific to a company or industry. Utilizing our optimized portfolio allocation tool for stock market price predictions, we are confident in our abilities to utilize diversification to mitigate this risk exposure for our investors.

B. Ethical Questions

1. In regards to ethical questions, FINRA (Financial Industry Regulatory Authority) mandates that all investment recommendations be suitable for investors. Suitability obligations are critical to ensuring investor protection and promoting fair dealing with customers and ethical sales practices. The three main suitability obligations include reasonable basis suitability (i.e., the potential risks and rewards of the recommended security must be sufficiently understood), customer

specific suitability (i.e., the recommendation must come after a thorough analysis of the customer-specific investment factors) and quantitative suitability (i.e., the recommended transactions are not excessive for the customer when taken together based on the customer's investment profile).

V. Extensions

A. Next Steps

1. To increase accuracy, we could have optimized for sampling frequency, as our grid search only considered several learning rates, hidden unit sizes, and number of dense layers. Sampling finer points in theory should lead to higher accuracy since we're providing the model with more information, but we found that doing so required a lot of GPU memory which we found to be very limited and much more computationally expensive. In addition, just optimizing for learning rate, hidden unit sizes, and number of dense layers took weeks since we had to do so for all 30 securities. Adding to this, restricting our time frame to intraday trading would greatly reduce the variance in the stock price and thus lead to less risk. More importantly, this would also allow us to train more informed models since using a higher sampling rate is more justified since we're trying to predict a shorter time frame out in the future. This in turn would greatly increase the predictive performance of our model, potentially leading to more profitable strategies for our investors.
2. Some of the primary extensions we envisioned including in our final product, but have not been able to implement due to time constraints, include the following:
    a) Cryptocurrencies → Modeling cryptocurrencies may have been an interesting extension for our project given the trading nature of these volatile assets. Although cryptocurrencies have been around for a little over a decade, they are still classified as a riskier asset and as a result often trade with heightened volatility. In the future, as the cryptocurrency market becomes more liquid, and investing in cryptos becomes more lucrative, it will be advantageous to model these trading patterns and generate accurate predictions.
    b) Bonds → As a debt instrument, bonds have historically been classified as a safer investment than the stock market. Thus, there have not been many attempts to model these trading patterns because it is evident that in times of turbulent markets bond prices rise and bond yields fall. However, an interesting project would encompass both stocks and bonds and focus on predicting the downfalls in the stock market, and concurrent upticks in the bond market, to turn a profit and stay ahead of the market.
    c) ETFs → An ETF is a type of pooled investment security that operates much like a mutual fund. These ETFs usually track a particular industry, sector, commodity or other asset and simultaneously trade on a stock exchange similar to a stock. Given its nature as a marketable security, it would be a relatively straightforward transition to modeling ETF trends, but there would have to be considerations taken into account, such as the volume, expenses, performance, holdings and commissions of the ETF.

VI. Appendix

A. Past features

1. At the inception of our project, we planned on implementing one, or multiple, GAN architectures. The most promising GAN architecture that we considered was the MarketGAN which trained using a 3-layer dense network as the generator, a 3-layer convolutional neural network as the discriminator and a boosted-tree classifier using XGBoost. We transitioned away from this classifier model, and instead utilized this model as a baseline, because we wanted a more accurate depiction of the projected stock market prices for the securities in the Dow Jones. In order to create a portfolio, we could not simply classify whether or not a certain security would go up or down whilst not including a quantitative depiction of the movement. Instead, we decided to implement an additional step, with a more advanced training protocol, to model and project the stock price movement and in turn make profitable trading suggestions. Thus, we settled upon training two sets of LSTMs, one which aims to predict the estimated stock price, and another which aims to predict the estimated mean squared error (MSE) which we used as a metric of volatility.
2. Another feature that we no longer incorporate in our final deliverable is the means by which an investor indicates their risk tolerance. Our initial GUI included a slider feature that allowed an investor to indicate their risk appetite on a scale from 1-10. However, based on our implementation of SciPy, and its limitations regarding its linear solver package and its shortcomings in integrating our personalized variance calculations, we transitioned to a simpler risk metric in which the investor either indicates that he/she would prefer a less risky portfolio allocation (i.e., the portfolio is completely diversified and hence equally weighted across the 30 securities in the Dow Jones), a more risky portfolio allocation (i.e., the portfolio recommends investing in a singular, or small group of, well performing securities), or a portfolio allocation that utilizes our Sharpe Ratio (i.e., calculated utilizing a distribution of a set of possible portfolio allocations and adjusts for overall risk-return characteristics).
3. Former Predictive Model Implementations - At the onset of our project, we anticipated utilizing a set of baseline models to determine how our portfolio allocation tool was performing. However, we decided that the best baseline for our model would be the implementation of a portfolio with the same weights as our optimized portfolio, but utilizing actual historical stock prices as opposed to our predicted prices. This comparison affords us the opportunity to have a concrete comparison of how our portfolio would have performed in an actual stock market environment.
    a) Monte Carlo Simulation - Utilizing a periodic daily return function, or the natural log ratio of the current day's price to the previous day's price, and incorporating the average, standard deviation and variance values, we are able to assess the probability that a security in the Dow Jones will follow a given trajectory. The price for the following day for each security in the Dow Jones is calculated using the current day's price multiplied by the exponential function taken to the power of the sum of a drift value, which is the average daily return subtracted by half of the variance value, and a random value, calculated by multiplying the

standard deviation by the inverse normal function with a random variable input. The result of these calculations, after repeating a desired number of times, is a simulation of future price movement.

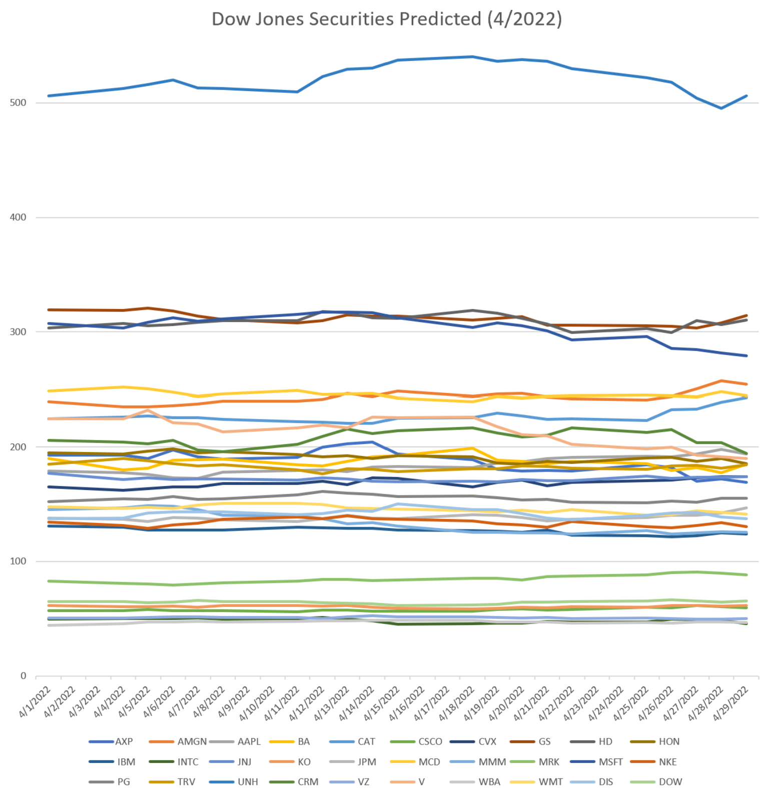


Figure 12: Monte Carlo price prediction simulation model

b) XGBoost Baseline - Using XGBoost, an algorithm which has been known to perform well on financial data, to predict the prices of a given security anywhere from 1-30 days in advance, given the last 30 days' prices. First, the data, which includes an indefinite amount of points and 4 categories, opening price [for a day], closing price, high price, and low price, is reformatted into Hankel matrices, one for each category. Each row in these matrices is an array of time-sorted values. These matrices are then combined to form one big matrix. This data preprocessing technique is used for both training and testing. Finally, using an autoregressive method, the XGBoost model is used to predict the values of the given security from 1-30 days out.

c) Stock Market Classifier Baseline- This GAN uses a 3-layer dense network as its generator and a 3-layer convolutional neural network as its discriminator. It is currently highly accurate when the stock goes down in the future, and relatively accurate when the stock goes up in the future. We made the decision to use the former predictor in our final deliverable over this, as it tended to be slightly more accurate, and its depth allowed for more parameter alterations for the user in the front end application.

B. Contributions

1. Drew Peterson - My main contributions revolved around user interface development as well as the integration between the three main software components of our project (generative network, optimization algorithm and application front-end). I spent a lot of time working with my fellow teammates to construct efficient and functional APIs that made integration between all the parts easy towards the end. My experience in team-based computer science classes here at Duke was invaluable during this phase. I also worked hard to design and implement an easy-to-use front-end interface that was capable of processing/error-checking user input before passing it to the portfolio optimization tool through the previously-mentioned APIs. Finally, I tried my best to be there for my teammates and act as a sounding board when they ran into trouble. As part of this, I worked with Will to troubleshoot our optimization algorithm at multiple points throughout the project and learned a lot about Modern Portfolio Theory in the process.
2. Ryan Middlemiss - Throughout this project I have taken on the role of coordinating the reports and presentations and confirming that our group is prepared to meet our deadlines. As a very organized and diligent person, I enjoyed drafting up presentations and documents in Google Docs or assigning slides for group members to present. Moreover, given my experience in the fields of Economics and Finance, I have been able to converse with group members tasked with more technical projects to ensure that their results are accurate and logical. I have thoroughly enjoyed working on this project with my group both on account of the content, which greatly interests me, and the willingness and excitement amongst our group members to work hard in an attempt to produce a viable product.
3. Adrian Lopez - I gathered and cleaned all necessary data needed for the entire project from Alpha Vantage. I designed, implemented, and tested the Stock Price Prediction tool, as well as helped Will incorporate the tool with his optimization code through an easy to use function where the input is the desired time into the future they want to predict which his code then uses as input. The function queries AlphaVantage for previous training day's prediction everytime a prediction is made to have the frontend app give updated results.
4. Will Long - Throughout this project I have focused primarily on generating appropriate baselines and building our optimization tool. Coming into this class, I had a decent knowledge of intermediate finance and a good background in machine learning, having taken courses in both during my time here at Duke. I have been very interested in the intersection of the two fields for a while, especially given the secretive nature of the quantitative finance industry. Using different models to generate future data in order to make informed decisions in the present proved to be a creative and mentally stimulating exercise using the knowledge I have gained in past courses. Furthermore, it was very exciting to be able to actually apply the knowledge I had gained in finance courses in order to build a functioning tool.

5. Lucas Josephy - Most of my contributions in this project came on the software development side. I came into this project with minimal knowledge on the economics / finance front, so I have strived to use my computer science and machine learning abilities wherever needed. I worked as one of our main model developers, largely working on our Stock Market Classifier, the GAN which performed well in predicting short-term stock movements. I supplemented others models when software help was needed / they were looking for advice as to how to proceed.

VII. References

A. Sources

1. Kenton, Will. "Who Is Harry Markowitz?" Investopedia. Investopedia, March 14, 2022. https://www.investopedia.com/terms/h/harrymarkowitz.asp.
2. Lysenko, Ivan. "Modern Portfolio Theory Model Implementation in Python." Medium. Analytics Vidhya, March 24, 2021. https://medium.com/analytics-vidhya/modern-portfolio-theory-model-implementation-in-python-e416facabf46.
3. https://www.finra.org/rules-guidance/key-topics/suitability.

B. Images

1. https://emerj.com/ai-future-outlook/machine-learning-finance-interviews-podcasts/.
2. https://www.google.com/search?q=united+health+stock&rlz=1C1DKCZ_enUS802US803&oq=united+health+stock&aqs=chrome..69i57j0i512l9.6558j0j7&sourceid=chrome&ie=UTF-8.
3. https://www.google.com/search?q=home+depot+stock&rlz=1C1DKCZ_enUS802US803&oq=home+d&aqs=chrome.0.69i59j69i57j46i20i131i199i263i291i433i512j0i512j0i433i512j69i60l3.879j1j9&sourceid=chrome&ie=UTF-8.