Enhanced LSTM Financial Time-Series Forecasting

§1 Project Summary

Equity markets develop prices through the cooperation and competition of millions of independent agents placing bids and asks for small stakes in companies. They are a representation of the faith and hope we place, for instance, in the US financial system and on its ability to innovate against the greatest challenges of our time. Something so abstract is inevitably rife with disagreement and financial markets fluctuate as new information becomes available to investors and speculators. Using statistical tools similar to those used by investment strategists, machines can anticipate market activity by studying the factors which leave the greatest impact on investor sentiment. Using Long Short-Term Memory (LSTM) Recurrent Neural Networks (RNNs), time-series forecasts can be developed which can motivate an investors decision making.

§2 Historical Context of Engineers in Finance

The general aim of science is to make accurate predictions: gravity works the same way today as it will tomorrow. Unfortunately, for the earnest theoretician, there is a recognition that most natural systems are dynamical and stochastic. Such systems are difficult to predict because they are highly sensitive to either initial conditions, unquantifiable factors, or both. The motivation to produce reliable models of semi-stochastic systems is clear and it is no less clear when financial incentives become involved.

United in their quest for returns on investment, investors and speculators have looked to quantitative finance and statistics to resolve market uncertainty and to understand where particular equities are likely to move. Perhaps the most revolutionary statistical finding of 20th century finance came from Harry Markowitz in 1952. Markowitz found that there are some portfolios which are more efficient than others, and that the efficient portfolio could help investors minimize risk for maximum returns. [1] After diversified efficient portfolios were constructed, however, investors were left to speculate on whether their work will yield financial returns in the near future and by what margins their hopes would be realized.

In 1973, Fischer Black and Myron Scholes created an efficient pricing formula for corporate options. [2] With the advent of computing power, the options market was created in 1973 which allowed speculators to hedge against risk and in some cases seek extra returns in exchange for betting for or against price movements. This emerging financial product gave investors a glimpse into the future of the underlying market that derivatives described, but neither market was immune to collective misunderstanding and folly. As revolutionary as computers were in the 1970's, they were not powerful enough to run the advanced statistical methods that mathematicians had long since described.

In the late 1980's, firms began developing software to run the Vector Autoregression and Value-at-Risk models needed to forecast local price changes. These tools reduced noisy and

complicated time-series data into generalized trends which could anticipate business cycles and generate expectations. Portfolio theory collided with the computing power of the early 90's to produce the age of the Quant, a highly trained mathematician who specializes in developing trading strategies and complicated financial products. As the demand grew for more data and more rigorous algorithms with which to study it, neural networks emerged as efficient multidimensional curve fitters, capable of crunching large volumes of data into general trends which could be relied upon.

A large volume of the artificial neural network (ANN) research today is in the primordial stage. The power and sophistication of these models is apparent to everyone but the exact configuration of neurons and weights to guarantee the most optimal results is as of yet unexplained. Rather, species of ANNs have been developed which cater to the needs of a diverse range of computing challenges. One such species, invented in 1997 by Sepp Hochreiter, is the Long Short-Term Memory (LSTM) recurrent neural network (RNN) which recalls previous outputs when studying new inputs. [3] This enables the RNN to more easily discover statistical relationships as it is fed new data.

Although the involvement of engineers in the world of corporate finance might seem obscure to the lay person, there is a clear trend among the worlds foremost financial firms to focus on an increasingly quantitative approach to developing investment strategy.

§3 Project Description

Financial data will be gathered and efficient portfolios will be developed to cater to varying risk-tolerance profiles. The capital asset pricing model (CAPM) will reflect how investors ought to behave with respect to the risk free rate of return and the classes of risk tolerance will be developed with respect to the capital market line (CML). Meanwhile, an LSTM network will generate risk profiles for the market as a whole as well as for the specific portfolios under consideration. The network will not only consider the pricing history of large market indexes but will consider how changes in β, price to equity ratio, dividend yield, CPI, and other quantitative factors have motivated investors to adjust their financial positions.

§3.1 Rationale and Significance

When predicting dynamical systems, most time series approaches consider the value-over-time history in a vacuum.[1] The justification for this approach is clear when there are clearly identifiable trends in the data-set that the machine learning (ML) system is studying. Within the financial context, however, the pricing history is only partly motivational. Investors mostly make financial decisions based on industry and stock specific qualities. For instance, investors may judge whether a particular stock is overpriced by using its price to equity ratio. Or they could study whether an investment is riskier than the market by analyzing its β. It is the thesis of this project that these quantitative factors play a dominant role in

[1] https://blog.statsbot.co/time-series-prediction-using-recurrent-neural-networks-lstms-807fa6ca7f

an investors decision making process. Ergo, for an LSTM network to truly garner predictive power, it must develop an intuition about how these ratios and stats affect investor outlook.

In one particularly interesting approach to time-series forecasting, Ryo Akita et. al. develop an LSTM to process news articles about particular companies and to estimate investor sentiment using the frequency of positive and negative mentions. [4] This approach contrasts sharply against the strict time-series value approach documented above. My strategy is similar: enhance an LSTM time-series analysis by including external factors.

§3.2 Plan of Work

In the following sections, I will describe the field of study for this project along with the methodologies employed and the milestones which must be completed in a timely manner for this research to be completed.

§3.2.1 Scope

I have higher ambitions for creating a trading platform which automatically balances portfolios and estimates market risk. However, I must deconstruct this project into its constituent parts before I proceed. First I would like to test whether market trends can be forecasted using what statistics are known about individual stock or portfolio. This project is not attempting to fundamentally improve quantitative financial maths nor is it proposing to improve the architecture of LSTM networks generally. Rather, it is looking to apply both techniques together in a way that hasn't yet been well documented.

§3.2.2 Methods

Essentially, the financial component to this project involves the formation of efficient portfolios and quantifying the relationship between acceptable risk and return. For this, Markowitz's efficient portfolio theory and the CAPM will allow us to quickly generate the portfolios we are looking for. Additionally, we can calculate the normalized moving window β to motivate the LSTM by $\Delta\beta$ and not by the explicit values of β itself. These quantifiable components have already been developed over the last 6 months of independent research and they simply have to be applied to the project.

The most challenging component in this project will be the development of the neural network. The architecture of the LSTM-RNN is particularly challenging and most scholars are unsure how to optimize them. To this end, the attributed of the LSTM could be fed to a genetic algorithm to determine the most optimal configuration. Such a genetic approach should be considered an absolute last resort as it would take an enormous amount of time and research to develop. However, should I exhaust my alternative approaches to LSTM optimization, the genetic algorithm might be the only remaining option.

After the system is developed fully, it can be served from a raspberry pi which is connected to the internet. This will allow for real time evaluation of financial data. The Twitter APIs offer a promising opportunity for feedback but the best results might come from serving a

small website from the device which can only be accessed securely. This security is less for privacy reasons, as the device has no access to the users financial data, just a simulation, but mostly to handle against buffer overflow. A blacklist of data-requests could mitigate against a server crash.

§3.2.3 Task Breakdown

This research project relies on the fusion of several distinct programming functions. Primarily, the successful implementation of LSTM software to judge market conditions and the financial front-end to support the LSTM analysis. Additionally, there are several improvements to the Raspbian-Linux OS which could make this software more efficient but I will mostly rely upon CHRON tables and reduced software packages to ensure efficient operation.

The first stage of research ought to be the compilation of all financial analysis software that I have developed thus far. The LSTM system cannot take input data before the synthesis of financial data is reliable and salable. This stage of the project ought not take longer than a few days.

After the financial base of the project is reliable developed, the research into LSTM can begin. I have found several promising leads for financial time series analysis LSTM systems and I can begin following their advice immediately to develop example projects before starting on my own.

Once the LSTM network is working properly it can be served within a larger ecosystem of software packages on the Raspberry Pi. Alternatively, a lower level system such as the MicroPython could be employed considering that the native language for this project will be Python, but this would present more challenges as the packages and software are updated.

Below, Figure 1 represents the work-flow which is to completed by this research project. Each item represents a function of subsystem which must be completed before the system can expected to produce reliable results.

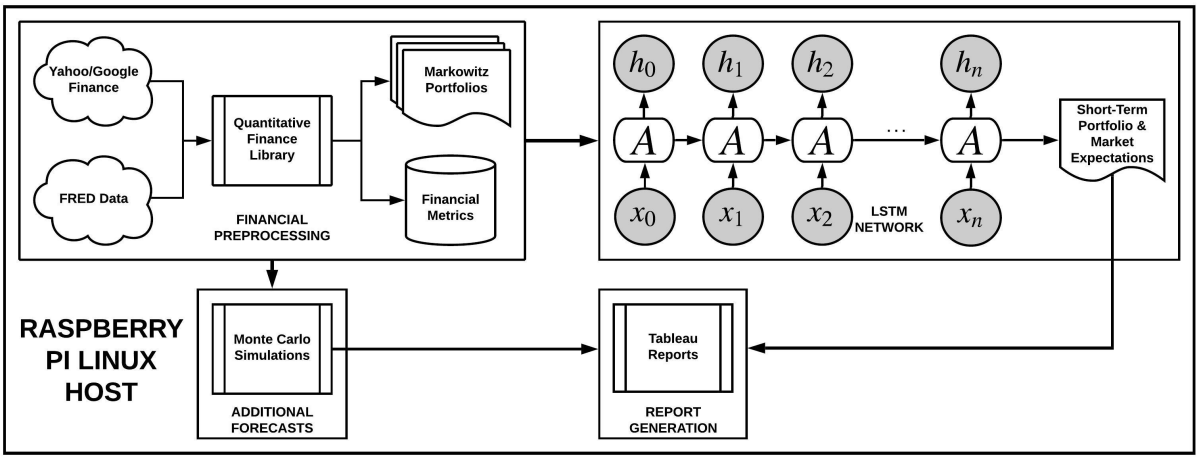

Figure 1: Diagram of system tasks.

In terms of the time-table for this research, the priority must be the development of the LSTM RNNs. With respect to these realities, this research task has been given the greatest

share of the time. Figure 1 details the processes which must be completed, by which date and what must have be accomplished before the task can be considered completed.

Process	**Completion Date**	**Success Criteria**
Financial Preprocessing	4/7/2019	When, in a single file or function, a set of stock tickers can be sent in and the efficient portfolios and risk analytics are returned automatically.
Additional Forecasts	4/10/2019	Automatically generates the Monte-Carlo spread plots given a portfolio or market-index input.
LSTM Network	5/01/2019	The time-series analysis is working to the best of its ability and strategies to optimize it further have been exhausted.
Report Generation	5/05/2019	Automatic reports are generated from a database of portfolios, risks, and LSTM guidance.
Linux Host Machine	5/10/2019	The ecosystem of python functions works without supervision and can be configured remotely.

Table 1: Schedule of tasks.

These time-frames are realistic if the software approaches are direct and without IT incident. In other words, if work proceeds linearly then these tasks can be achieved with 5 hours per week over the course of this semester. However, projects rarely proceed in a linear fashion and I expect delays which may exceed the time allotted for this semester.

§4 Conclusion

The knowledge and experience gained from this project will vastly improve upon my ability to both apply neural networks to complicated problems as well as to understand how financial risk is estimated and how its conclusions are acted upon. This is a unique approach to financial time-series analysis, possibly offering future developers the opportunity to program more accurate and reliable forecasting tools.

§5 Personal Statement

It is important for me to personally express both my excitement to develop a time-series approach as well as my general disdain for the financial speculation proposed by this project. When it comes to managing wealth, it is wise to focus on risk mitigation and fee minimization. To these ends, investors are wise to avoid unnecessary complexities in their investment strategy as well as to reduce their active management role so as to lower their fee structure. The methods and sciences which this project glorifies ought only to be explored by corporate wealth managers and market makers. As for the lay person, I subscribe to Benjamin Graham's *Intelligent Investor* which outlines a strategy of index funds, dollar cost averaging, and the strict avoidance of market timing. Avoiding local sell-offs and corrections by liquidating your investments, moving to more secure investments for a time, and then finding an appropriate time to reenter the market, is foolish. If one were to genuinely follow the speculative approach outlined in this paper, I would suggest that they avoid speculating with anything they are not completely willing to lose. Neural networks have captured our imagination since emerging from obscurity in 2006, but they cannot predict the future. I earnestly implore you to keep an open mind but not to believe the hype.

References

[1] H. Markowitz, "Portfolio selection," *The Journal of Finance*, vol. 7, pp. 77–91, 1952.

[2] F. Black, "The pricing of options and corporate liabilities," *Journal of Political Economy*, vol. 81, pp. 637–654, 1973.

[3] S. Hochreiter, "Long short-term memory," *Neural Computation*, vol. 9, pp. 1735–1780, 11 1997.

[4] R. Akita et al., "Deep learning for stock prediction using numerical and textual information," 2016.

To: Professor Mohammad Imtiaz, PhD.
From: Austin Dial
Subject: ANN Financial Analysis
Date: May 11, 2019

Enclosed within this document lie the results of a thorough literature review and empirical analysis of utilizing ANNs for financial research. In particular, this report summarizes existing approaches to financial time series analysis including applying LSTM structures for time series prediction. However, neural networks have not historically been applied to equity valuation. The motivation is clear to develop an approach for predicting broad market price movements. Not only does this report detail such an approach, an additional series of financial analysis tools have been developed which are less speculative.

It is my firm belief that speculating on financial markets is generally poor financial practice. However, an investigation into the methodologies involved in such speculation will make for interesting research at the very least.

Embedded Linux Financial Analysis Platform

Report By: Austin Dial
Submitted To: Professor Mohammad Imtiaz PhD.
Date: May 16, 2019

Abstract:

Equity markets develop prices through the cooperation and competition of millions of independent agents placing bids and asks for small stakes in companies. They are a representation of the faith and hope we place, for instance, in the US financial system and on its ability to innovate against the greatest challenges of our time. Something so abstract is inevitably rife with disagreement and financial markets fluctuate as new information becomes available to investors and speculators. Using statistical tools similar to those used by investment strategists, machines can anticipate market activity by studying the factors which leave the greatest impact on investor sentiment. Using Long Short-Term Memory (LSTM) Recurrent Neural Networks (RNNs), time-series forecast studies can be trained to predict financial patterns.

Contents

1 Introduction

According to the efficient market theory, a market correction occurs when speculative pressures evaporate and rational investors dominate the market, allowing for the price of a security to once again reflect the underlying value of the system or business it represents. For instance, if the public becomes irrationally exuberant[1] about a particular stock, then the sudden demand for the equity product will send prices soaring beyond what any reasonable investor would be willing to pay. Then, investors who already possess a long-position in the company will sell their over-valued shared, causing prices to lower again and *correct* for the exuberant bullishness that once dominated the market's conception of price.

Luckily for investors, there are always signs that a stock has become over valued. For instance, sudden high volume purchases of an equity from multiple sources over a short period of time could indicate irrational exuberance, an obnoxiously high price to earnings (P/E) ratio could suggest that the price does not reflect a fair share of the business, or prices for a stock that are close to their 52 week high could mean that the stock is statistically likely to fall in price. All of these indicators, however, are only part of the bigger picture. Ultimately, there is never a guarantee that a stock is overvalued and only by taking many of these factors into account can a model become accurate and reliable. Multi-variate long short term memory networks are offer many exciting approaches to this problem.

2 Literature Review

Before discussing a machine learning approach to quantitative finance, it is important to understand where my research fits into the corpus intellectualis. Particularly, a historical review of engineering approaches to financial problems will help eliminate the stigma that separates the two fields. Additionally, an exploration into the research of finance risk assessment will expose the direction that financial analysts are taking in their work. Lastly, to avoid discussing the history of neural networks would be intellectually impoverishing, so the rise of ANNs and RNNs are discussed.

2.1 A Brief History of Financial Engineering

The general aim of science is to make accurate predictions: gravity works the same way today as it will tomorrow. Unfortunately, for the earnest theoretician, there is a recognition that most natural systems are dynamical and stochastic. Such systems are difficult to predict because they are highly sensitive to either initial conditions, unquantifiable factors, or both. The motivation to produce reliable models of semi-stochastic systems is clear and it is no less clear when financial incentives become involved.

United in their quest for returns on investment, investors and speculators have looked to quantitative finance and statistics to resolve market uncertainty and to understand where

[1] A term coined by Alan Greenspan.

particular equities are likely to move. Perhaps the most revolutionary statistical finding of 20th century finance came from Harry Markowitz in 1952. Markowitz found that there are some portfolios which are more efficient than others, and that the efficient portfolio could help investors minimize risk for maximum returns [1]. After diversified efficient portfolios were constructed, however, investors were left to speculate on whether their work will yield financial returns in the near future and by what margins their hopes would be realized.

In 1973, Fischer Black and Myron Scholes created an efficient pricing formula for corporate options [2]. With the advent of computing power, the options market was created in 1973 which allowed speculators to hedge against risk and in some cases seek extra returns in exchange for betting for or against price movements. This emerging financial product gave investors a glimpse into the future of the underlying market that derivatives described, but neither market was immune to collective misunderstanding and folly. As revolutionary as computers were in the 1970's, they were not powerful enough to run the advanced statistical methods that mathematicians had long since described.

In the late 1980's, firms began developing software to run the Vector Autoregression and Value-at-Risk models needed to forecast local price changes [3]. These tools reduced noisy and complicated time-series data into generalized trends which could anticipate business cycles and generate expectations. Portfolio theory collided with the computing power of the early 90's to produce the age of the Quant, a highly trained mathematician who specializes in developing trading strategies and complicated financial products. As the demand grew for more data and more rigorous algorithms with which to study it, neural networks emerged as efficient multidimensional curve fitters, capable of crunching large volumes of data into general trends which could be relied upon.

A large volume of the artificial neural network (ANN) research today is in the primordial stage. The power and sophistication of these models is apparent to everyone but the exact configuration of neurons and weights to guarantee the most optimal results is as of yet unexplained. Rather, species of ANNs have been developed which cater to the needs of a diverse range of computing challenges. One such species, invented in 1997 by Sepp Hochreiter, is the Long Short-Term Memory (LSTM) Recurrent Neural Network (RNN) which recalls previous outputs when studying new inputs. [4] This enables the RNN to more easily discover statistical relationships as it is fed new data.

Although the involvement of engineers in the world of corporate finance might seem obscure to the lay person, there is a clear trend among the worlds foremost financial firms to focus on an increasingly quantitative approach to developing investment strategy.

2.2 Portfolio Management Theory

The contemporary approach to portfolio development involves the minimization of idiosyncratic risk. Specifically, broad portfolios that span multiple industries and businesses avoid taking unnecessary risks and begin to approximate the market. By investing in portfolios

that are comprised of many uncorrelated financial instruments, the investment is protected against the risks associated with the mismanagement of specific companies or industries.

Harry Markowitz went one step further by proving that some portfolios were more efficient than others and that the most efficient portfolio could be determined using basic statistical methods [1]. All portfolios are composed of assets of investments which make up a percentage of the total investments. To *balance* a portfolio is to optimize the percentage of the total portfolio which each investment contributes, known as the portfolio weights. Markowitz plotted the historical returns of a given set of portfolios against their historical volatility, see Figure 1. Some portfolio weights resulted in higher returns for the same volatility, suggesting that some portfolios are more efficient at generating returns for the same quantity of acceptable risk. These portfolios are termed *efficient*, because they optimize returns for risk.

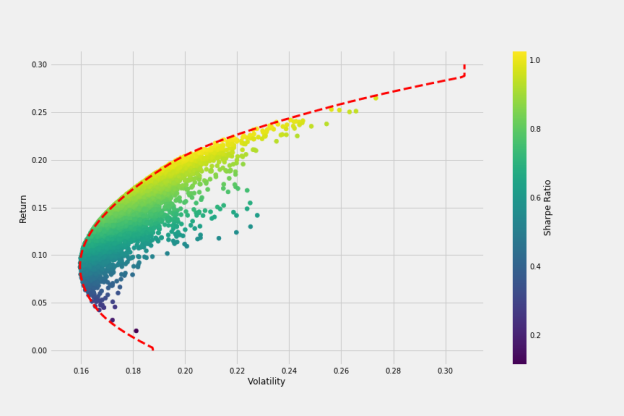

Figure 1: Markowitz's *Efficient Frontier*.

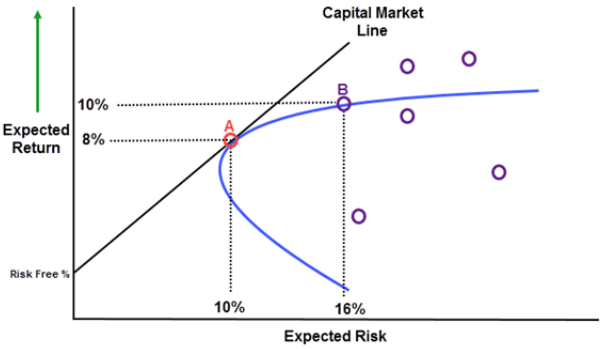

Figure 2: Sharpe's graphical *CAPM*.

Markowitz ended his research with the advent of the efficient frontier in 1954. However, ten years later, William Sharpe took Markowitz's work one step further to suggest a pricing model that factors the risk of an investment scheme into its resultant price [5]. Sharpe named his approach to risk assessment the *Capital Asset Pricing Model* (CAPM) and it is still employed to this day. The CAPM looks at the risk-free rate of return, a theoretical investment with a finite return and no risk, see Figure 2. Although no such investment realistically exists, the U.S. Federal Reserve has the capacity to set interest rates and then work with the Treasury to print money to pay for the interest and principal payments on its loans, known as Open Market Operations. Consequently, the U.S. can always raise its debt ceiling to avoid default at the cost of inflation. The risk-free rate of return is standardized world-wide as the 30 year U.S. government bond due to this low probability of default, allowing for CAPM to function with its risk-free rate of return. CAPM compares the risk of an investment to the value of avoiding investments in equities by buying government bonds.

As time has progressed, more advanced models of valuation have emerged but the advanced made in the mid 20th century with Markowitz and Sharpe still apply to the modern day. However, the power of these models lies in their ability to study past data and make predictions using broad trends. On a day to day basis, these risk assessment tools will

likely fail to predict the short-term liabilities of the portfolio. A more advanced approach to probabilistic modeling is needed.

2.3 Artificial Neural Network Theory

All financial prediction systems are essentially using time-series data to inform their opinion of the future trajectory of an individual security or of a market as a whole. Neural networks, as exciting as they are, lack the ability to associate trends across the dimension of time. Consequently, ANNs with short memories have been designed to study the underlying trends of time-series data. These networks specifically involve RNNs, which train connections of linear temporal sequences to understand time-dependent trends [6]. This approach to machine learning, known as Long Short-Term Memory (LSTM), has produced exciting results due in part to its capacity to generalize the trends which it observes.

Time series analysis, the study of data with a strong relationship to the progression of time, revolves around the identification of broad trends that exist as statistical subtleties within noisy data sets. This approach is perfect for the study of markets, where there is a high degree of speculative noise which causes the price of a security to deviate from its real value. For investors, this presents a risk assessment challenge: by what means can the underlying value of a security be uncovered? This is a classification problem, where machines are tasked with finding traits within a dataset which can help to distinguish it as on species of data apart from another. Speaking to the power of LSTM networks, Dr. Karim et. al. write that LSTM approaches to time-series classification are showing promising results [7]. In particular, Karim points out that taking a more manual approach to LSTM design by means of an *attention* device cause the network to learn datasets more efficiently, yielding more powerful results.

In one particularly interesting approach to time-series forecasting, Ryo Akita et. al. develop an LSTM to process news articles about particular companies and to estimate investor sentiment using the frequency of positive and negative mentions [8]. This approach contrasts sharply against the strict time-series value approaches documented above. Akita's work demonstrates that time-series classification tasks can be applied to non-numeric datasets where the value system is more qualitative. This marks an important distinction in the realm of machine learning, because now machines can approach problems that only humans were thought to comprehend: the ability to interpret text within a historical context. Networks such as LSTM can begin to interpret linguistic trends which could motivate trading algorithms faster than humans could respond.

3 Empirical Work

I have been tasked with developing a system through which users can request stock portfolio analyses automatically. The elements of this report will include statistical properties of the stocks, efficient portfolio configurations, Monte Carlo worst case scenarios, normalized historical returns, and finally a neural network generated expectation of real equity value.

Users will communicate with this system by email and all data will be pulled from Quandl.[2] Reports will be generated through a Raspberry Pi which will actively monitor the server email as well as the stocks in question. In the following sections, I will detail the inner workings of this system as well as the neural network approach and final results.

The following empirical work focuses on the development of an ecosystem of software components which have been designed to support an amateur financial analyst in their research. The details for the inner workings of this ecosystem can be found in Figure 3.

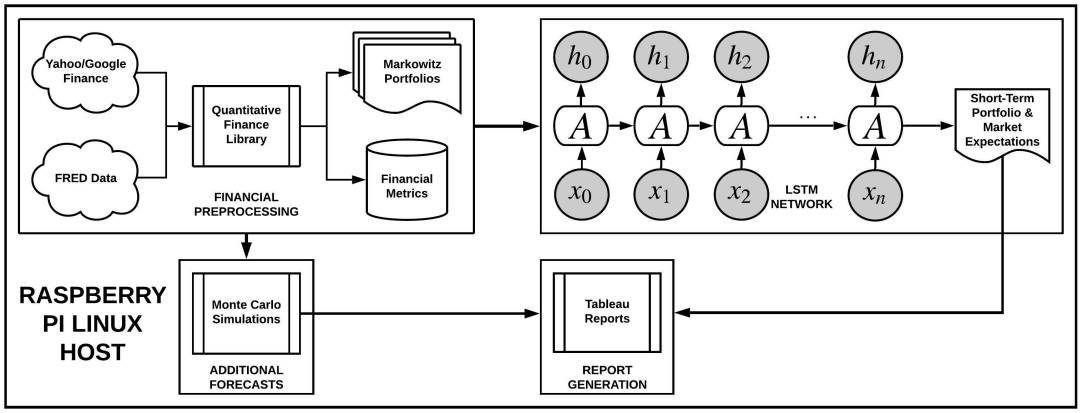

Figure 3: Software ecosystem.

3.1 Statistical Report

The first stage in any financial analysis of securities is to understand the statistical basics behind the securities in question because the fundamentals can be built upon in growing complexity. The statistical report generated by the Python Financial Server (PyFi) includes the average annual rate of return, standard deviation, and variance of each individual security in the portfolio. Additionally, it finds the normalized stock prices, Figure 4. From here, the correlation and covariance can be calculated between the securities.

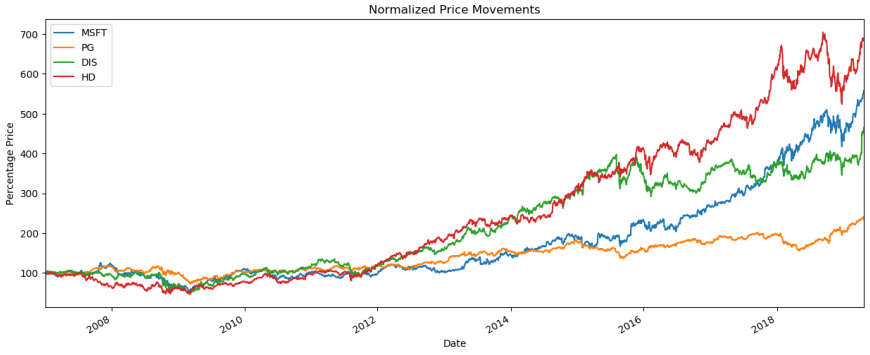

Figure 4: Normalized Stock Returns.

[2] See `www.quandl.com` for more information.

3.2 Risk Assessment Report

The risk assessment section of the report details that quantity of risk that can be eliminated through efficient diversification of the portfolio. When creating a portfolio with several securities involved, it is important to understand how the variances of the individual instruments will affect the variance of the portfolio as a whole. Let w_n represent the weight of the n^{th} security, σ_n represent the standard deviation of the n^{th} security, and ρ_{12} be the correlation of the two securities. Then, using the following algebraic relationship, we can find the variance of a portfolio with two securities:

$$(w_1\sigma_1 + w_2\sigma_2)^2 = w_1^2\sigma_1^2 + 2w_1\sigma_1 w_2\sigma_2\rho_{12} + w_2^2\sigma_2^2$$

Variance makes for an excellent measurement of portfolio risk because it represents the fluctuations of the individual securities and the correlations of several securities in portfolio together. There are two types of investment risk: Un-Diversifiable and Diversifiable risk. Un-Diversifiable risk depends on the variances of individual securities. It results from consumer spending changes, wars, forces of nature, etc. Investors cannot control for risks affecting the economy as a whole. On the other hand, Diversifiable risk is the result of idiosyncrasies within a single company. For instance, some companies are mismanaged, some industries suffer recessions, but by diversifying your equity holdings over multiple companies and industries you can eliminate this idiosyncratic risk. Studies show that by holding 25 or more properly diversified equities, idiosyncratic risk all but disappears. This report details the quantity of risk which can be eliminated by adjusting the asset allocation of the portfolio.

I addition to the risk estimates detailed above, the risk assessment tools run a Monte Carlo simulation of the portfolio based on the last few years of returns. It generates scenarios detailing the possible price movement by using the standard deviation and mean of these returns to model their behavior. It generates each price successively using the following:

$$E[S(t)] = S_0 * e^{\mu t}$$

Where μ represents the drift of the security or portfolio. By creating hundreds of thousands of sessions, the Monte Carlo simulation results in a spread of best, worst and most likely (average) case scenarios for the movement of the portfolio in question, see Figure 5.

For convenience, the best and worst case scenarios have been colorized in black and red respectively. The average case, also the most likely, is highlighted in blue. Despite the simulation's balanced appearance, note that the best and worst case scenarios are not equal in magnitude and that even the most likely case shows positive returns as opposed to zero. These findings reflect that the portfolio is more likely to generate returns over this 60 day period than it is likely to experience a loss. Such concepts become important when studying value-at-risk models.

3.3 Efficient Portfolio Report

When determining how best to allocate funds across several equities, investors are attempting to maximize returns and minimize risk. In doing so, investors must often avoid investing too

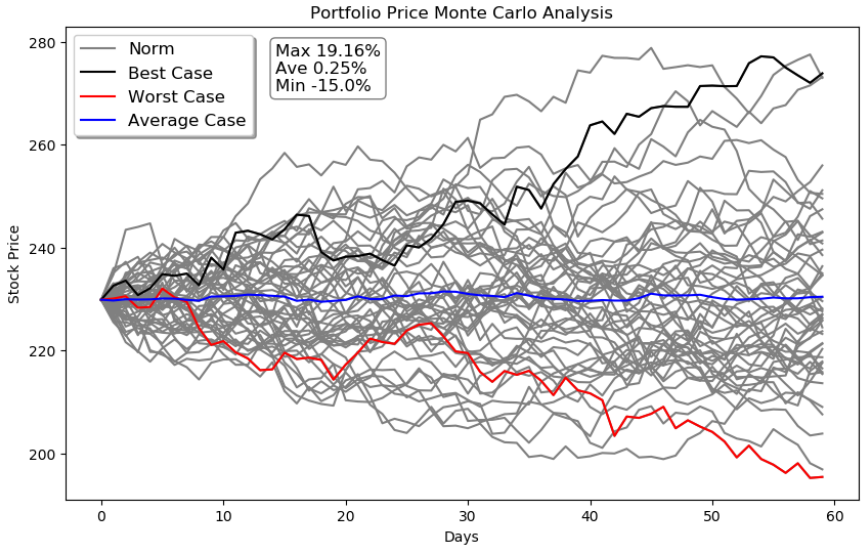

Figure 5: Monte Carlo worst and best case results.

much money into high-yielding stocks and instead favor this balance with lower-yielding and safer stocks. This balance between risk and return for equities was first detailed by Harry Markowitz in 1952 and it allows us to find the highest yielding and lowest variance portfolios [1].

In order to find the portfolio return, let $E(R_p)$ represent the expected portfolio return and $E(R_i)$ represent the expected return of the i^{th} security with weight w_i then the expected portfolio return can be found.

$$E(R_p) = \sum_i w_i \cdot E(R_i)$$

The portfolio variance is similar, let σ_p represent the standard deviation of the periodic returns on an asset and ρ_{ij} reflect the correlation between the returns of assets i and j, then:

$$\sigma_p^2 = \sum_i w_i^2 \sigma_i^2 + \sum_i \sum_{j \neq i} w_i w_j \sigma_i \sigma_j \rho_{ij} = \sum_i \sum_j w_i w_j \sigma_i \sigma_j \rho_{ij}$$

This process can be generalized to a portfolio with any number of assets under management. Using two securities and 5000 samples, the efficient frontier can be visualized in Figure 6.

As assets move south of the upper boundary of the efficient frontier, they earn lower returns despite taking on the same financial risk. Note that there is a point at which the portfolios generate the lowest possible variance for a lower rate of return, farthest left-hand point. Additionally, there exists a high-yield portfolio which generates the highest returns for the highest risks, most upper right-hand point. This relationship indicates that investors would be wise to run a Markowitz analysis lest they stumble into accidentally accepting unreasonable risk or investing in inefficient portfolios and missing out on additional returns at the frontier.

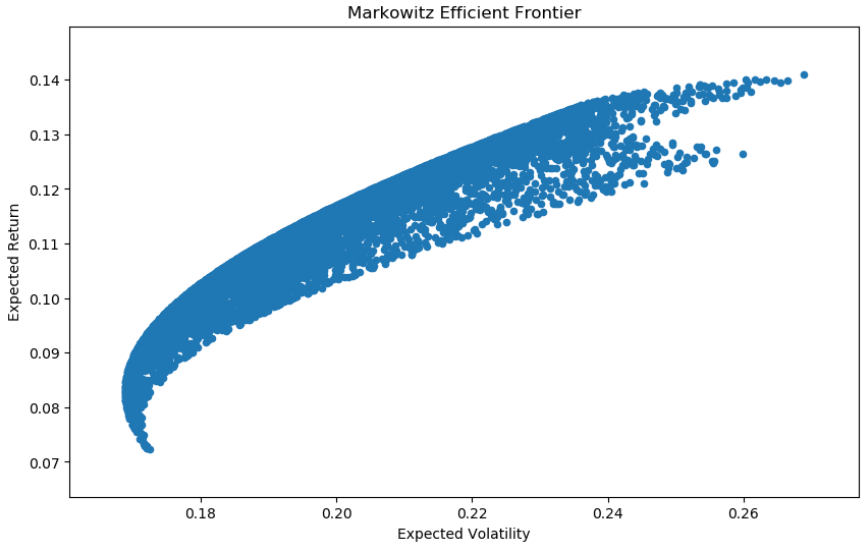

Figure 6: Portfolio efficient frontier.

3.4 US Equity Forecast Report

This aspect of the report involved studying the S&P500 composite index to determine the next SP500 price given the historical prices along with a number of ratios which describe over-all S&P500 value and performance.

3.4.1 Dataset Collection

The focus of the neural network forecast model is the S&P500 composite index using S&P500 metrics from the last 20 years. In particular, this network studies the monthly price to earnings ratio (PE), the monthly dividend yield, and the ratio between the earnings yield and the 10 year treasury note (EYTY ratio). The neural network tries to relate these metrics to the monthly adjusted closing price of the index. All data is freshly downloaded from Quandl and is processed in a separate python program before being exported into a *CSV* file. This file is then loaded by the neural network models for study. This model allows for the data to be studied and updated independently.

There were several issues related to the gathering of the S&P500 data. For instance, the data can only be downloaded on a monthly basis, this means that recessions, for instance, are represented by fewer data points than would be ideal for a neural network study. Additionally, the dates for the data features did not align correctly and many dates were missing. As many as 33% of the dataseries instances were missing at least one value, see Figure 7 for the distribution of missing values across the time frame of study.

To account for this, the dataseries was imputed using the average of the instances between the missing value date. For example, if yesterdays date was missing from the series, then the instance would be imputed using the instance from two days ago and the instance from today. Using this method, the general trends of the S&P500 were generally preserved and

the pricing data reflect that of the actual S&P500 data, compare Figures 8 and 9. However, finer details of the S&P500 fluctuations were lost in this process, please read § 3.7 for specific details on future study.

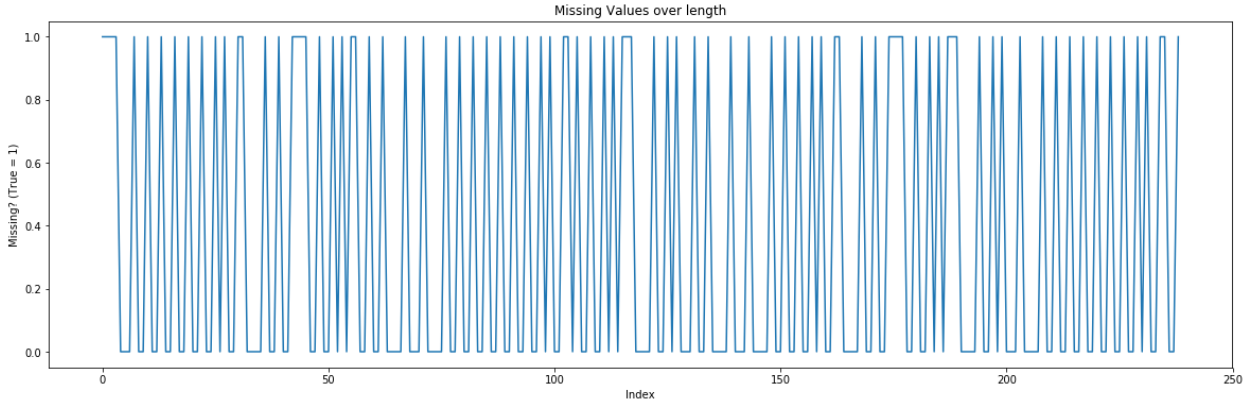

Figure 7: Missing values across time.

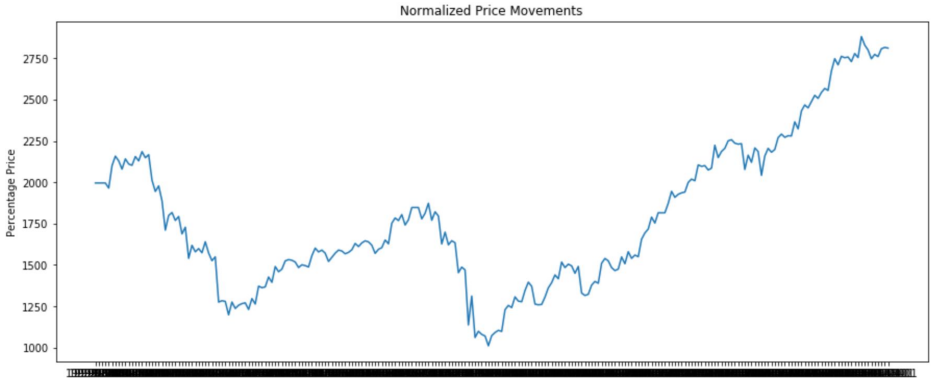

Figure 8: Reconstructed S&P500.

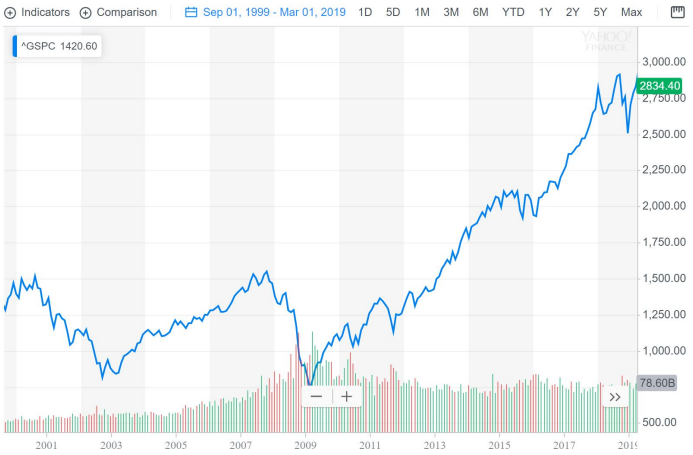

Figure 9: Yahoo S&P500 actual pricing.

3.4.2 CNN Approach

The convolutional neural network approach evaluated the S&P500 data by performing convolution on the dataset, thereby limiting its dependence on time. This is definitely an interesting approach to time series analysis because it is stressing the irrelevance of time and focusing on the S&P500 metrics instead. This approach was selected because of its ability to generalize and to act as a baseline measurement for LSTM network performance.

Despite relying less on the temporal dimension of the dataseries, time is still an important factor, especially for using the CNN as a basis for comparison with other networks. For this reason, the train-test split was not an entirely stochastic process of selecting individual data points for study. Rather, the training split selected a point within the time of study as a line of demarcation between the train and test data. This line of demarcation was selected pseudo randomly to allow for the model to explore different ranges of data in between successive executions of the training routine. The train-test split is visually plotted in Figure 10 below.

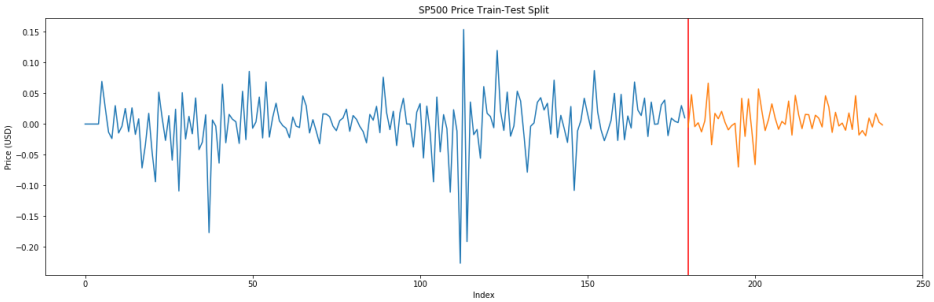

Figure 10: Train-test split.

When training the model, the CNN experienced losses which were indicative of learning behavior. When training neural networks, it is important that the accuracy of the models predictions increase with the number of epochs as the loss of error decreases.[3] Under ideal scenarios, each of these trends would occur exponentially. As illustrated in Figure 11, the network exponentially and successively looses error throughout the standardized 1000 epochs of its training routine.

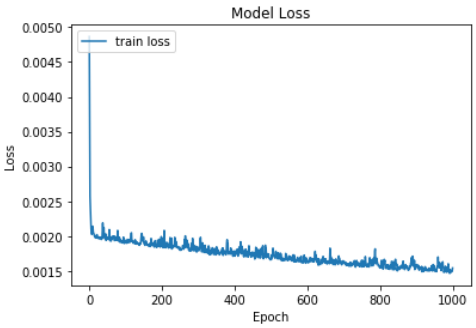

Figure 11: CNN training losses over epochs.

3.4.3 CNN Results

The network training results were mixed. Although the CNN was able predict the trends in the 2008 financial crisis (middle of Figure 12), it clearly fails to detect the many smaller daily returns of normal market activity in the preceding and proceeding years. This could indicate that the nuances of fluctuating earnings and dividends during normal business cycles

[3] An epoch is an event during training when the network is exposed to the entire training data. Successive epochs expose the network to the training data multiple times to reinforce learning.

are too subtle for the network to study. On the other hand, this could involve the approach to the development of training data. Perhaps the violent fluctuations in the features during abnormal market activity were related to investor panic whereas the focus on investors and other market actors in the normal markets was on variable which were not considered in this analysis.

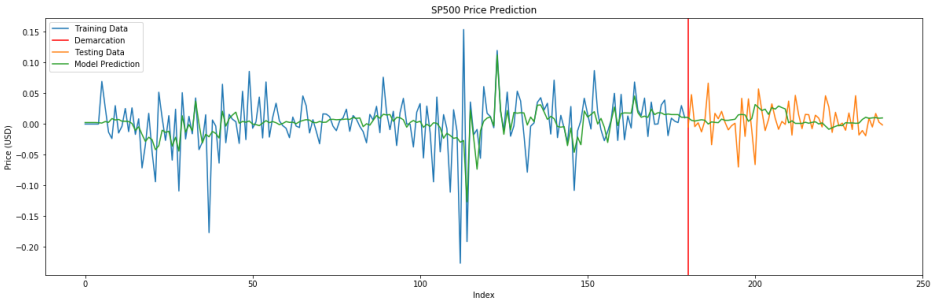

Figure 12: CNN monthly return predictions.

Using the starting S&P500 closing price, the prices of the composite index can be simulated from the predicted monthly returns generated by the CNN model. The results of the model are shown in Figure 13 in the form of S&P500 prices instead of returns. This visualization is more intuitively understood and illustrates an interesting result. Past the line of demarcation, the model departs from the actual closing prices but continues to predict its trends. For instance, economic downturns and recoveries can still be seen in this testing data despite the estimated closing prices being lower than predicted. In other words, the CNN is correctly guessing the trends, but not the values. Although this approach as focused on pricing, the price of the index is irrelevant, it is a system which the study imposes on the data. The trends of the predictions are the highlight of the analysis and they are not vacant from this model's predictions.

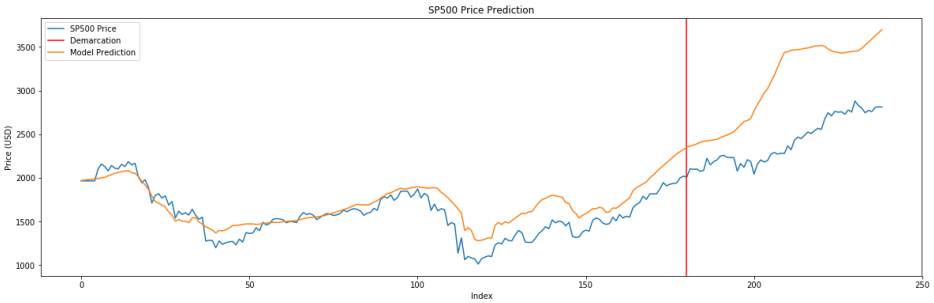

Figure 13: CNN monthly price predictions.

3.4.4 LSTM Approach

Whereas the CNN approach endeavored to reduce the effect that time had on the time series, the LSTM approach works in the opposing direction. Long short-term memory networks, as described previously, recall previous instances when developing their next prediction. Consequently, time not only remains relevant but its relevance is exaggerated. For this reason, LSTM networks are recommended for time series analysis. As with the CNN, the training

and testing data were determined by splitting the dataseries along a line of demarcation, see Figure 14.

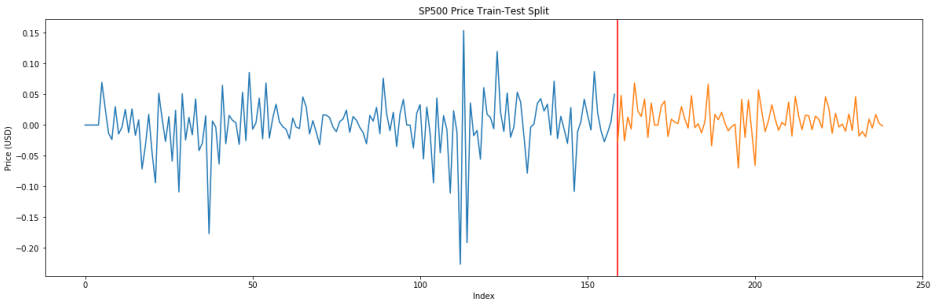

Figure 14: Train-test split.

The training losses of the LSTM network as similar to the CNN in that they decay almost exponentially from an initial high, see Figure 15. Although the loses occasionally sputter, the overall trend of the losses is clearly downward. However, in hindsight, the model does not appear to have reached the end of the loss decay and could use more training. The reason that the epochs were stopped at 1000 despite this observation is to serve as an effective comparison to other models under identical testing conditions. Additionally, this study labored to avoid over training the networks to respond to false trends such as the imputing averages which were described earlier.

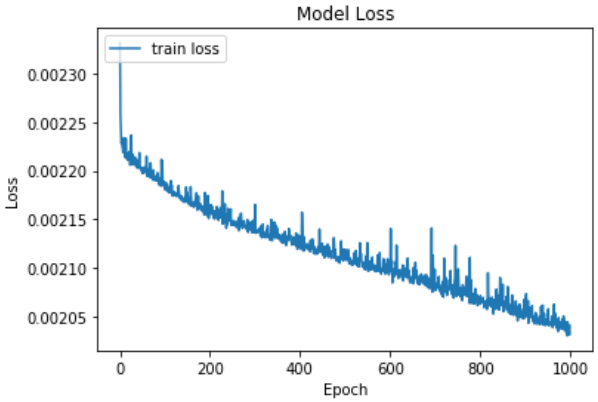

Figure 15: LSTM training losses over epochs.

3.4.5 LSTM Results

The monthly return predictions of the LSTM network are even more muted than that of the CNN for obvious reasons. Whereas the CNN was convolving the data and taking time less seriously, the LSTM network apparently has a smoothing effect as it takes time even more seriously than it should, diminishing the networks ability to respond sharply to changes in the input. Consequently, the violent return trends exhibited by the CNN are absent from the LSTM, see Figure 16.

When the returns are converted into simulated S&P500 prices, Figure 17, the effects of the smoothing become more apparent. The LSTM does not respond sharply to the decline in US equities in 2008 and lags behind the closing price whereas the CNN was irrationally exuberant in its expectations. Just like the CNN, the LSTM is not accurate in terms of pricing but it able to pick up some trends in S&P500 behavior.

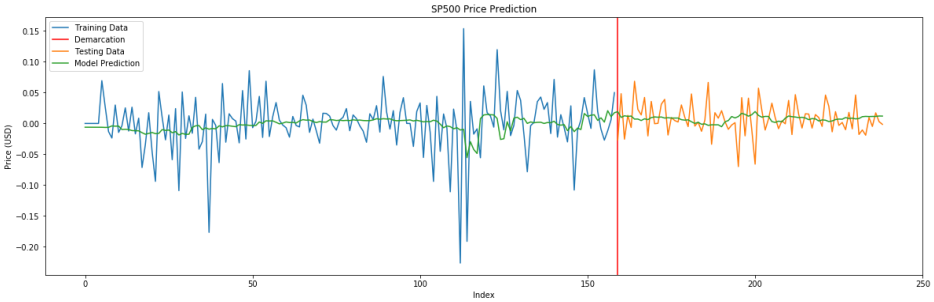

Figure 16: LSTM monthly return predictions.

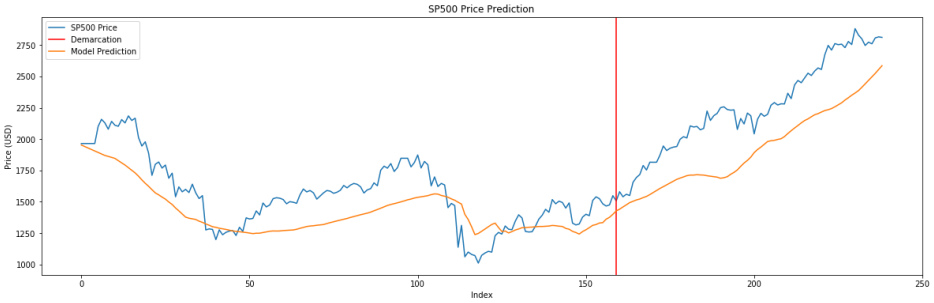

Figure 17: LSTM monthly price predictions.

3.5 Embedded Development

The software ecosystem described within this section is capable of running on an embedded platform. In particular, this ecosystem is running on a Raspberry Pi (RPi) which is capable of sending automatic emails, thus acting as an embedded Linux server. This ecosystem operates by pulling data from the markets via Quandl, analyzing the results, and sending regular emails to the user. To process the timing of this system, CRON tables were employed which have the ability to schedule tasks for the embedded system to regularly run. As of the writing of this report, the RPi is scheduled to run its analysis and contact the user every month.

3.6 Empirical Conclusions

The financial analyses explored in this report have been promising. The technical background behind Monte Carlo analysis and Markowitz Efficient Frontiers, also know as Modern Portfolio Theory (MPT), is strong and well established. However, the neural network forecasting approach is relatively new and untested. A proper balance between excitement and skepticism is needed to proceed further in this field. Although the CNN and LSTM networks strayed farther from real closing prices than is acceptable, their capacities to capture market trends is apparent and remarkable.

Unfortunately, traditional measurements of accuracy do not apply to this problem. For example, an R-squared measurements would punish the networks for failing to adhere to the actual pricing values but would disregard the networks ability to adhere to actual pricing trends. The trends of the market might in fact be more valuable to investors and speculators,

allowing them to create volatility spreads based on the expected movements of the markets. It is imperative, given the publicity which neural networks and machine learning receive, to stress that these methods are new and unreliable. These networks show promise but further study is needed before money can be risked on their foresight, and lack thereof.

3.7 Areas for Future Study

The most troubling aspect of this project can be found in Figure 7, missing values in data gathering. The project was always intended to be run on an embedded platform and for this reason it was important to develop the data gathering in an autonomous manner. Were this project focused entirely on the neural networks and to be run on a non-embedded platform, then the data gathering would be forced to obey stricter standards where such large quantities of data were not missing. Be that as it may, the data was surprisingly meaningful even after the imputing process, as can be seen in Figures 8 and 9, as it resembles the S&P500.

The metrics used for this study were selected on an availability basis only. In other words, Quandl did not have the 52 week price range nor the trading volume so these factors were omitted from the analysis. This is problematic because a securities price when hovering close to its YTD high closing price is likely to face a correction. This is primarily because prices ought to grow gradually and sudden or prolonged highs are indicative of exuberant behavior, not of dramatic changes in underlying value. Additionally, volume ranges can indicate the degree to which the market is participating in a price movement. Without these factors, it is difficult to train a neural network to predict prices. In future versions, these variables must be taken into account.

Many methods were explored to support investors in their quest for at or above market returns. However, many models remain to be explored. In particular, stock options were entirely ignored but could allow for investors to take more risk with their portfolios and hedge against this risk through derivatives. The Black-Scholes model was never explored, which could price options and determine where opportunities in derivatives markets exist. Building off of the efficient frontier, the Capital Asset Pricing Model (CAPM) could be explored in more detail to include the risk free rate of return and its intersection with the frontier. The methods explored in this report are intriguing but far from exhaustive.

4 Project Conclusions

This project set out to explore the quantitative methods behind modern financial models and to develop a new machine learning approach for price forecasting. What began at rates of return and volatility for individual securities rolled into efficient portfolios and then into the realm of speculation with ANN models used to study broad US markets. The culmination of this semesters work is a software ecosystem which can independently complete financial analysis and forecasting tasks and communicate them to the user through email. It is capable of running without supervision and for extended periods of time on an embedded system

which never needs to turn off. This system is also highly configurable, enabling the user to reconfigure it to complete other finance tasks.

Although neural networks are in their infancy, they offer the promise of simplifying our approach to problems which cannot be modeled with directly analytic methods. The results enclosed in this report indicate that neural networks offer varied and often inconsistent results but are also able to understand underlying trends that humans cannot detect with conventional means. Further study is needed for these systems to become useful to the mainstream forecasters, but for now they offer a glimpse into a world of terrifying artificially intelligent means. Like Alice looking through the keyhole, we stand at the precipice of what awesome worlds lie beyond.

5 References

[1] H. Markowitz, "Portfolio selection," *The Journal of Finance*, vol. 7, pp. 77–91, 1952.

[2] F. Black, "The pricing of options and corporate liabilities," *Journal of Political Economy*, vol. 81, pp. 637–654, 1973.

[3] C. Sims, "Macroeconomics and reality," *Econometrica*, vol. 48, pp. 1–48, 1980.

[4] S. Hochreiter, "Long short-term memory," *Neural Computation*, vol. 9, pp. 1735–1780, 11 1997.

[5] W. Sharpe, "Capital asset prices: A theory of market equilibrium under conditions of risk," *Journal of Finance*, vol. 19, pp. 425–442, 1964.

[6] D. Rumelhart et. al., "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, 10 1986.

[7] F. Karim et al., "Lstm fully convolutional networks for time series classification," *IEEE Access*, vol. 6, pp. 1662–1669, 2018.

[8] R. Akita et al., "Deep learning for stock prediction using numerical and textual information," 2016.