Time Series Analysis and Recurrent Neural Networks with Financial Data

Douglas Allen - allen.do@northeastern.edu
Karan Shah - shah.karan3@northeastern.edu
Ameya Santosh Gidh - gidh.am@northeastern.edu

Dec 12, 2022

1 Objectives and Significance

In the world of finance, predicting the future value of an asset is a goal as old as stocks and bonds themselves. When trying to predict the future price of a stock or bond, one can either use deep industry and economic knowledge to predict price moves, or use statistical and machine learning methods to use previous prices in order to make predictions about future stock or bond price movement.[7] The opportunities for successfully predicting future asset prices can be as simple as building an automated trading machine to earn better returns than the market, or as complex as policymakers identifying early risks to market stability and taking action to prevent an economic meltdown.

In this project, our goal was to compare two different methods for predicting stock price: Statistical Time Series Analysis, and Recurrent Neural Networks. Time Series Analysis (TSA) methods have been around for almost 100 years [19] and involve modeling the evolution of a value over time. The models used are probabilistic and make assumptions about the correlation between values of interest, in our case financial asset prices, in the past and prices in the future. Recurrent Neural Networks (RNNs) on the other hand, are modern Machine Learning tools that use a large amount of data to train many parameters in order to minimize error based on patterns seen in the historical evolution of the price of an asset. We also were interested in comparing the performance of these models using only one feature

or multiple features such as economic indicators.

What we found is that the RNNs performed slightly worse than traditional series models at prediction stock prices at any interval. Additionally, they are black-box models that provide no insight or interpretability as to why certain relationships exist between past and future prices. However, they require significantly less domain knowledge

2 Background

2.1 Important Concepts

Some important concepts that we studied in order to implement these models fell into three main categories: Background on financial topics, the statistics behind Statistical TSA, and the mechanics of RNNs - specifically Long-Short Term Memory (LSTM)

2.1.1 Financial Background

A financial asset is anything that is expected to generate value (traditionally thought of as 'cash flows') at some point in the future.[6] The current value of that asset is always the present value of expected future cash flows. We say 'present value' because there is a time component to money. $1 today is not the same as $1 a year from now.

The two main types of financial assets are a share of a stock, where you own a piece of the company; and a bond, which can be thought of as debt - a company owes you a certain amount of money on a certain day. Both of these assets have prices that fluctuate over time as the market changes. In this paper, we are exclusively looking at stock price prediction, which have much different characteristics than bonds. Specifically we will be trying to predict indices of stock prices, which reflect an average many different companies of many different industries.

The last important concept from a financial perspective is that stock prices tend to exhibit exponential growth[5] - that is, the market typically thinks about a company in terms of the time it takes to double its share price, rather than increase by $1. This will be important to remember when we get to data transformation methodology.

2.1.2 Statistical Time Series Analysis

A time series is a collection of data points indexed by time - most commonly at equal time intervals apart. Often, these data points are realizations of random variables that are not independent: $X_0, X_1, \ldots, X_t$.

Many times we can model the probability distributions of the value of something at a point in time as a function of the time series' previous values along with independent functions. An example would be something like $X_{t+1} = \alpha X_t + \sigma\epsilon_{t+1}$ where α and σ are constants and ϵ is a standard normal random variable, often called a 'shock'.[16]

In time series analysis, we are be interested in estimating the parameters α and σ. Once we estimate those parameters, we can then calculate the usual metrics of a random variable, such as the expectation and variance.

2.1.3 RNNs and LSTM

Simply put, neural networks take as input a vector of features, and pass those inputs through many layers of both linear and non-linear transformations, creating a multitude of functions of functions with an output that can itself be a vector or scalar or classification.

The trouble with trying to use traditional neural networks on time series is that the 'feature vector' in this case could be an arbitrarily long set of time series data, and there could be instances where information from a month ago is not actually as important as information from a year ago (say for weather prediction).

In this project, we focus on an RNN framework called Long-Short Term Memory (LSTM). A goal of training any neural network is to manage how error impacts weights in the network - 'managing gradient flow'. In a very simple RNN, error corrections tend to either grow to infinity or disappear. The LSTM framework use "gates" to manage how error impacts weight updates. [11]

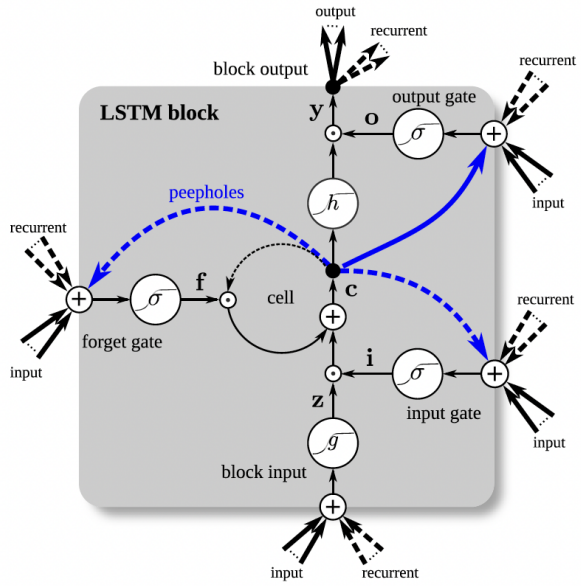

Figure 1: LSTM Gate Setup [3]

2.2 Previous Work

A review of previous literature on TSA and using RNNs for financial time series forecasting is a comparison of old statistical methods and modern machine learning methods. The origin of time series and auto-regressive models (models that rely on previous observations for prediction) goes back to Yule in 1927. [19] [17]. Meanwhile the development of the LSTM model occured in the early wave of machine learning and neural network creation in 1999.[11]

Separately, these two methods have then gone on to enjoy much attention as tools to predict stock prices. From the formalization of the Auto-Regressive Integrated Moving Average (ARIMA) model by Box and Jenkins in their seminal book *Time Series Analysis: Forecasting and Control* in 1970 [1] to the more focused *Analysis of Financial Time Series* by Tsay which is in its third edition [16], statistical TSA is a well-worn topic.

Work on applying LSTM networks to financial time series has grown significantly in the past decade. There are papers of application of LSTM on the single feature of stock price [4], and work has been done on multiple features being input into the model [15]. There has also been a lot of work done on comparing other RNN frameworks with respect to prediction performance, such as a comparison of LSTM with a Gated Recurrent Unit (GRU) model.[9]

Where we think our project is interesting is the comparison of these fundamentally different methods. Relatively little work has been done to compare TSA to LSTM models. A paper by Kobiela shows an ARIMA model having better predictive performance over long time horizons, which is counter to intuition.[8] Another paper shows LSTM outperforming ARIMA models using similar methodologies.[12] This disparity indicates there may be a lack of expert-level collaboration on this comparison. The other point to note about trying to perform a comprehensive literature review is that for obvious reasons much of the work in this area is being done by private companies and any success they may find is not something they would want to share with the wider research community.

Where we think our work may be particularly interesting is that we also compare multivariate auto-regressive models (Vector Auto-Regressive, VAR) to LSTM models trained on multiple features. Our literature review so far has shown very little work so far in comparing these two methods, with only a few papers discussing non-financial applications.

3 Methods

3.1 Description of Data

A fortunate aspect of working on financial data is that much of the data is either regulated by the government (publicly traded companies must provide financial performance in a standard format), or directly provided by the government for economic indicators. This makes gathering data (at least historical data) relatively easy. Our data was split into two categories: stock index prices and economic indicators.

3.1.1 Stock Indices

Any publicly traded company will issue shares on different exchanges around the world. Their stocks will perform differently given the company's individual performance and economic conditions. It can be useful to average a group of representative companies into a single metric, known as an index.[7]

The index that averages the most companies and is widely considered the benchmark of overall stock market performance in the US is the Standard and Poor's 500 (S&P500). It selects 500 companies on various metrics such as financial viability, industry representation,

and most importantly, market capitalization (share price times number of shares on the market).[10]

This index is available from a multitude of sources; we obtained the closing price value of the index for every trading day from 1962 to the present from Yahoo Finance.[18]

3.1.2 Economic Indicators

For our multi-feature models, we incorporated four other economic indicators that were similar to, but not exactly, the indicators used by Siami Namin in their multi-variate LSTM model[12]:

1. The US unemployment rate in %
2. The Consumer Pricing Index (CPI) for Urban Consumers - this is an weighted average of prices for common goods, seasonally adjusted
3. The 10 Year US Treasury Rate - this is the rate of return for a 10-year US treasury bond bought on a given day
4. The Volatility Index - This is a measure of volatility of stock prices calculated by the Chicago Board of Options Exchange (CBOE) [2]

The unemployment rate, CPI, and 10-year treasury rate we obtained from the Federal Reserve of St Louis (FRED) website [13] and the VIX was obtained from the CBOE [2].

3.2 Methodology

We had to use two different methodologies to come up with our two different types of models. The time series model is rooted in statistics and has an involved procedure for finding how many terms to include. Creating the LSTM involves understanding the available packages and constructing the model to train appropriately.

3.2.1 Time Series Model Methodology

ARIMA Model

An ARIMA model of auto-regressive order p and moving average order q is defined as follows:

$$X_t - \sum_{j=1}^{p} \alpha_j X_{t-j} = \alpha_0 + \sum_{l=1}^{q} \sigma_l \epsilon_{t-l} \tag{1}$$

[16]

The left side of this equation represents the auto-regressive (dependency on past values) nature, and the right side is a moving average of the shocks to the time series.

In general, building an ARIMA or VAR model involves removing any overt trends in the data resulting in residuals that are known as 'stationary'. A stationary time series $X_0, \ldots, X_T$ is such that the following equality holds for any arbitrary timeshift h and any number of times n:

$$P(X_t = x_t, \ldots, X_{t+n} = x_{t+n}) = P(X_{t+h} = x_{t+h}, \ldots, X_{t+h+n} = x_{t+h+n}) \tag{2}$$

[16]

This is equivalent to saying the probability distribution is invariate under time. This is a difficult property to prove, so often we try to get weak stationarity, which is that $E[X_t] = E[X_{t+h}]$ and $E[X_t^2] = E[X_{t+h}^2]$ - that is the mean and variance are equal over time.

Looking at the S&P500 index, and using domain knowledge, we see that there is an exponential trend to the data, so our first transformation is to take the log of the price:

After taking the log price, we perform a linear regression to remove any linear trend (the α_0 in the ARIMA model), then determine what orders p, q our ARIMA model should be. In order to find p (the number of previous prices to use in our model), we use the partial auto-correlation function PACF. This is an iterative process, where the correlation between the current value and one previous timestep is calculated, then the correlation between the current value and two timesteps ago with the first previous timestep subtracted. This is continued for as many lags as desired, and the p is the last significant correlation.[16] It is easy to see visually:

We see that we include just 1 previous timestep. Then the q is determined by the amount of overall correlation between time series entries. We determine this by looking at the significant lag terms of correlation in the auto-covariance function.[16]

Figure 2: S&P 500 Price Since 1962

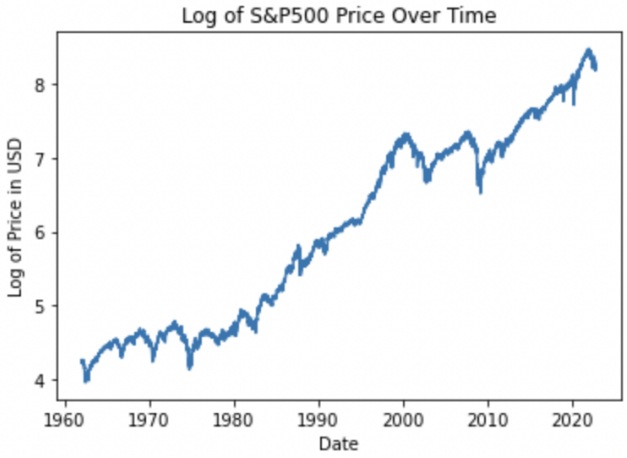

Figure 3: Log S&P 500 Price Since 1962

Here we see that around 10 terms of moving average degree will account for the noise to a statistically significant degree (the shaded area is the statsmodels range of statistical significance confidence).

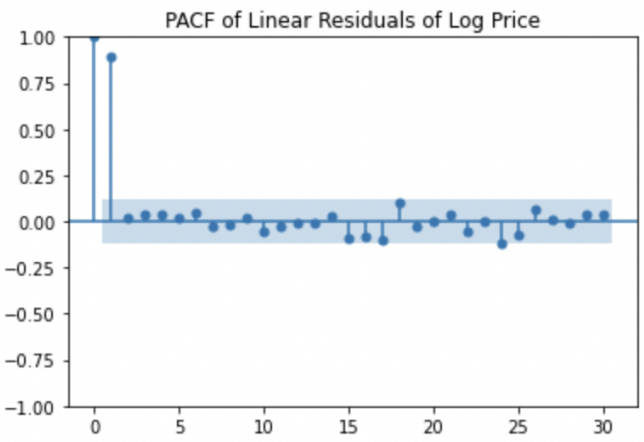

Figure 4: PACF for Linear Residuals of Log Price

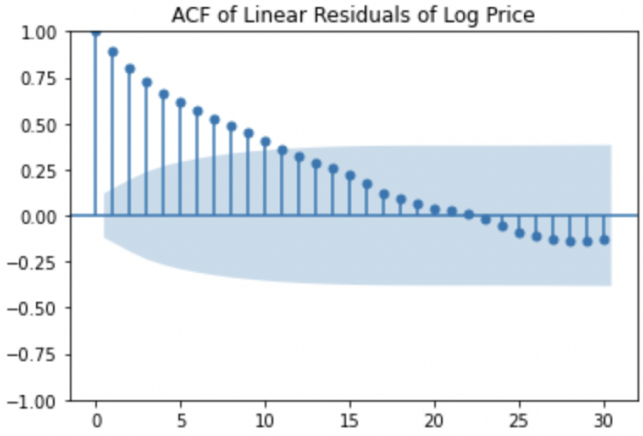

Figure 5: ACF for Linear Residuals of Log Price

VAR Model

A VAR model of order p is defined for k features as follows:

$$\mathbf{x}_t = \mathbf{a}_0 + \sum_{j=1}^{p} A_j \mathbf{x}_{t-j} + \Sigma \epsilon_t \tag{3}$$

All of the time series components are now $k \times 1$ vectors, and A_j are $k \times k$ matrices along with Σ. For the VAR model, the process is much the same, but all input series need to transformed into stationary processes, and a similar process is performed.

It is important to note that for both ARIMA and VAR we are predicting stationary movement away from the trend that we already removed from our time series. Therefore to

predict a price we need to add our linear regression trend and then take the exponent (the reverse of what we did to make the time series stationary).

3.2.2 LSTM Model Methodology

Our LSTM methodology was much less involved, but required setup of the model in Tensorflow and transformation of data into the proper format. The general steps we took to make predictions on an LSTM model were as follows:

1. We divided the time series into train and test time chunks (we will outline that process in section 4)

2. Given n training data, we transform it into n differences of $X'_t = X_t - X_{t-1}$, with $X_0 = 0$

3. Then we create a pseudo-supervised training set where each X'_t is then 'labeled' with X'_{t+1}, giving us $n - 1$ labeled pairs

4. Finally, we need to use min-max transformation to get all values between -1 and 1 so that the Tensorflow hyperbolic tangent prediction function would work appropriately

[14]

At the end of this process we had $n - 1$ scaled, labeled pairs of sequenced data on which our RNN could be trained. Then our model trained using the ADAM optimization method using a mean squared error as a cost function (the prediced difference in yesterday's versus today's price being the estimate of the neural network).

3.3 Evaluation Strategy

Overall, our evaluation of the methods will come down to how close our prediction was to actual stock prices. We will measure that performance using Root Mean Squared Error RMSE

In the following formulas, we have n trading days of closing prices that we are trying to predict, $\hat{y}_i$ as our estimate of price and y_i as the actual price. The period ends at time T

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \tag{4}$$

We will also take into account interpretability and barriers to use when discussing model performance.

4 Results

We trained ARIMA, VAR, and LSTM models on the data described in section 3.1 in order to predict S&P500 index prices over a time horizon. We varied the time horizon on which the models needed to predict - the time horizons always ended on 12/31/2021, and allowed for a full year (252 trading days) of model training. For example, to predict over a 20 day time horizon, we would have the training data be from 12/3/2020 to 12/3/2021, then the 20 trading days from 12/3/2021 to 12/31/2021 would be our prediction horizon.

4.1 Example Predictions for 20 Days

Before looking at the LSTM performance, we can see some qualitative differences in how the ARIMA and VAR models perform. Both revert to the mean which is the nature of auto-regressive models. After a certain amount of forecasting periods, the model is predicting off of its own predictions, and the expectation will be a stationary trend.

However the VAR model appears to, in this case, either reverting to the mean faster or predicting the coming up-swing versus the ARIMA model. It will take different periods of prediction to figure out.

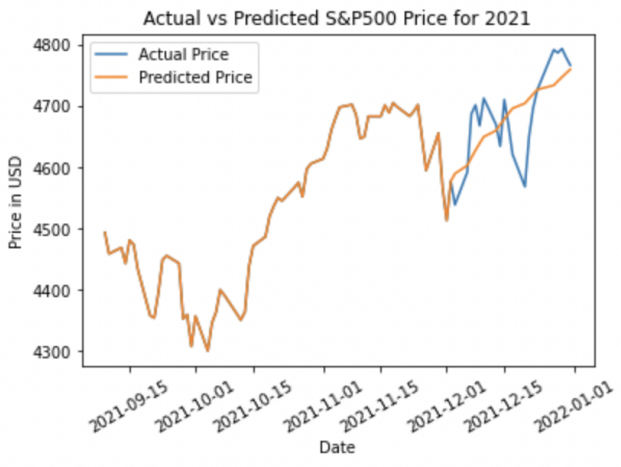

Figure 6: ARIMA-Predicted S&P 500 Price versus Actual - 20 Day Time Horizon

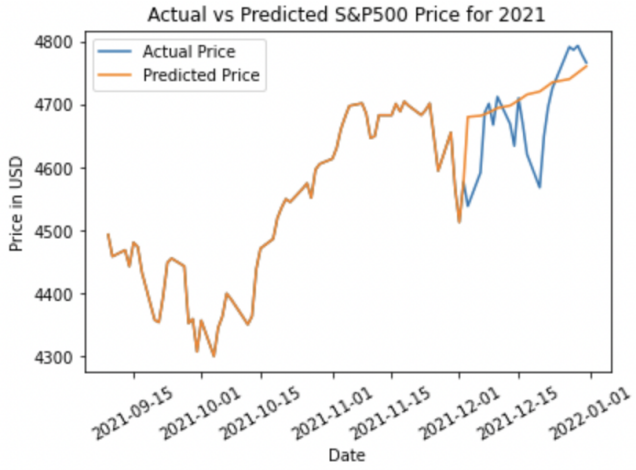

Figure 7: VAR-Predicted S&P 500 Price versus Actual - 20 Day Time Horizon

For the ARIMA and VAR models, we also came up with our idea of a 'trivial' predictor, which was the log-linear trend we removed from the stock price. This would be our benchmark performance since it involves no other time series analysis other than what is in essence OLS. Our formula for the trivial predictor was:

$$\hat{y} = e^{\beta_0 + \beta_1 t} \tag{5}$$

For predicting 20 days before 12/31/2021, $\beta_0 \approx 8.228, \beta_1 \approx 9.233 \cdot 10^{-4}$, and t is number of trading days after 12/3/2020.

Since we need to transform either time series based on this log-linear estimate, the

ARIMA and VAR equations will look similar, with the same values as above for the 20-day β_0 and β_1. The ARIMA model is as follows:

$$\hat{y}_{ARIMA} = e^{\beta_0+\beta_1 t+\hat{x}_t} \tag{6}$$

$$\hat{x}_t = .8655x_{t-1} + \sum_{l=1}^{10} \sigma_l \epsilon_{l-i} - 8 \cdot 10^{-4} \tag{7}$$

Here x_t is the deviation from the log-linear trend of the S$P500 price. All σ_l are small constants with absolute value under .1

The VAR model produces equations for predicted all of the input variables, but the prediction for the S$P500 is as follows:

$$\hat{y}_{ARIMA} = e^{\beta_0+\beta_1 t+\hat{x}_t} \tag{8}$$

$$\hat{x}_t = .89x_{t-1} + .0043u_{t-1} + .0011c_{t-1} \tag{9}$$

$$+ 7.4 \cdot 10^{-5} v_{t-1} - 7.5 \cdot 10^{-5} \tau_{t-1} + 5.4 \cdot 10^{-4} \tag{10}$$

Here u_t is the unemployment rate, c_t is the CPI, v_t is the VIX, and τ_t is the 10-year US treasury rate, all at time t.

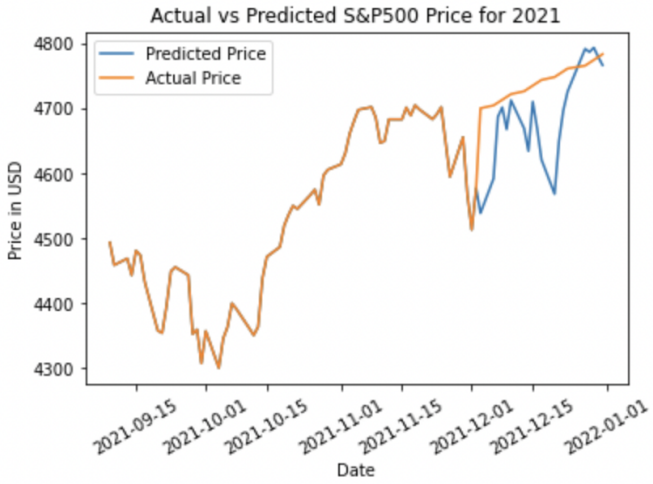

Figure 8: Log-Linear (Trivial) Predicted S&P 500 Price versus Actual - 20 Day Time Horizon

In comparing the TSA methods to this 'trivial' predictor, we can see that both statistical methods revert to this trend, but carry recent information for some amount of time before reverting. The VAR model was more apt to revert to the mean (not predict any upswings)

than the ARIMA model, which had a powerful method of moving averages to keep it from snapping in any direction too fast. However on time scales between 5 and 30 trading days, the ARIMA model did a slightly better job of predicting price than the trivial predictor.

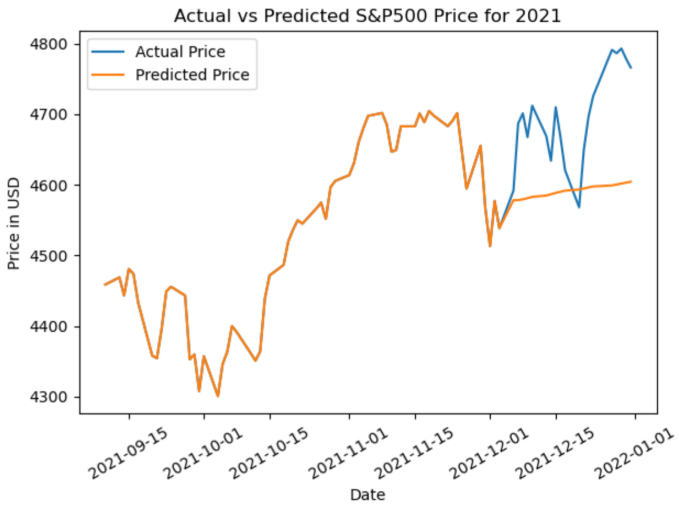

Figure 9: LSTM-Predicted S&P 500 Price versus Actual - 20 Day Time Horizon

The LSTM performance was surprisingly poor relative to the time series models. We believe what was happening is that is was learning some general linear upward trend and reverting to it very quickly after being left to predict. We believe we need to incorporate more lag-steps into training and predicting, and multiple features should help with training the model to predict.

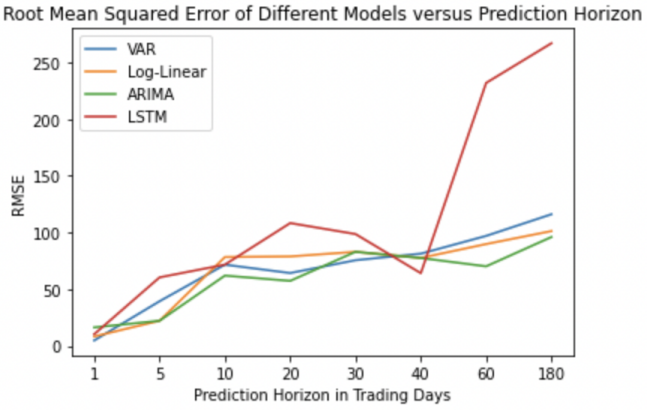

Figure 10: RMSE of Each Model vs Prediction Horizon

We saw that there were certain time horizons that each model predicted quite well, like 10 and 40 days, however this was simply due to the stock index price matching the linear trend over that horizon. Overall, this matches the performance of Kobiela et al on their ARIMA versus LSTM performance[8], and we did not see added performance from the VAR model. However we believe this was due to the difficulty in removing all non-stationary trends from all input features.

5 Conclusion

Overall, the results of this project provided both an aspect that we expected and a surprise. The way in which our results aligned with our expectations was that an LSTM neural network would outperform traditional statistical models. However our other expectation was that the VAR model would outperform the ARIMA model by taking in other economic indicators into consideration when making predictions.

5.1 Comparison of ARIMA and VAR

We believe that our VAR model did not perform as well because of the difficulty in truly removing the non-stationary trends from all of the different data sources. Some economic indicators were published monthly and required us to perform a fill-forward process to transform them into daily indicators, but perhaps a linear interpolation would have been more appropriate.

On the other hand, with ARIMA, much more focus and attention can be put into understanding the single feature, removing trends appropriately, and determining orders of auto-regression and moving average.

5.2 Comparison of Time Series and LSTM

The LSTM model was much more set-and-forget in terms of understanding the nature of the underlying data, but as seen in the results provided no transparency as to the connection between past stock performance and future predictions. In fact, during one iteration of LSTM training, we seemed to be getting unreasonably good results on prediction. We were inadvertently including the test period into our training cycle, but there was no metric from

the LSTM model that would indicate this, we just had to not trust the output and dig into our code more.

5.3 Improvements and Future Considerations

There are many aspects to this project that we would want to explore with more time. The first would be to get more facile with the TensorFlow package and LSTM to incorporate multiple features into our prediction with a neural network. The second would be to improve our trend-removal for both ARIMA and VAR models, perhaps even automating them so that we can select any time window and create a stationary time series. Third would be to try and perform more data transformations like time lags and include different features for multi-variate models. Finally, as mentioned in the background section, there are many more financial assets other than stocks. Bonds such as US and corporate debt, options on commodities, and foreign exchange futures all have different trends and dependencies on other economic indicators.

It is clear why financial time series analysis can be the sources of focus for ones' entire career. With this project we scratched the surface, but we were able to perform unique comparisons between older and newer machine learning methodologies.

6 Individual Tasks

6.1 Karan

I worked on building a scraper to scrap the S&P500 data from Yahoo Finance.[18] In addition, I built Git repository to make our workflow dynamic and control source code management for non-linear development. I also synced Git with one drive for team to consistently update all of our files and create a cloud backup. I modularized code to preprocess scraped data and build full dataset that would be used for both time series and deep learning approach. Then, using the source blog[14], created logic to build LSTM layer and forecast on uni-variate data.

6.2 Ameya

I determined the fundamental papers that would be useful in this thought experiment, helped with data persistence and management. I came up with initial idea for scraping S&P500 data, and found several skeletons online that could be followed. Reviewed any pending code work and kept the data management process systematic. Finally, played assistive role on the LSTM generation and acted as soundboard to discuss the internal math. Also, produced separate data analysis to make some general conclusions of the data necessary for carrying out both the ARIMA and LSTM modelling

6.3 Doug

I used financial domain knowledge to ensure data quality and data transformations were sound. I built the ARIMA and VAR models and removed non-stationary components from the time series in order to fit the models. Additionally I was the primary LaTeXteam member for the project, managing the visuals and flow of the project report.

References

[1] George Box and Gwylim Jenkins. *Time Series Analysis: Forecasting and Control*. 2016. ISBN: 978-1-118-67502-1.

[2] *CBOE Vix*. https://www.cboe.com/tradable_products/vix/. Accessed: 2022-12-11.

[3] Klaus Greff et al. "LSTM: A Search Space Odyssey". In: *IEEE Transactions on Neural Networks and Learning Systems* 28.10 (Oct. 2017), pp. 2222–2232. DOI: 10.1109/tnnls.2016.2582924. URL: https://doi.org/10.1109%2Ftnnls.2016.2582924.

[4] Magnus Hansson. "On stock return prediction with LSTM networks". In: (2017).

[5] J. Hull. *Options, Futures, and Other Derivatives*. 2012. ISBN: 9780132164948.

[6] *Investopedia - Financial Assets*. https://www.investopedia.com/terms/f/financialasset.asp. Accessed: 2022-12-11.

[7] *Investopedia - Types of Financial Analysis*. https://www.investopedia.com/terms/f/financial-analysis.asp. Accessed: 2022-12-11.

[8] Dariusz Kobiela et al. "ARIMA vs LSTM on NASDAQ stock exchange data". In: *Procedia Computer Science* 207 (2022). Knowledge-Based and Intelligent Information Engineering Systems: Proceedings of the 26th International Conference KES2022, pp. 3836–3845. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2022.09.445. URL: https://www.sciencedirect.com/science/article/pii/S1877050922013382.

[9] Sang Il Lee and Seong Joon Yoo. "Threshold-based portfolio: the role of the threshold and its applications". In: *The Journal of Supercomputing* 76.10 (2020), pp. 8040–8057.

[10] *SP 500 Fact Sheet*. https://www.spglobal.com/spdji/en/indices/equity/sp-500/#. Accessed: 2022-12-11.

[11] Jürgen Schmidhuber, Sepp Hochreiter, et al. "Long short-term memory". In: *Neural Comput* 9.8 (1997), pp. 1735–1780.

[12] Sima Siami-Namini and Akbar Siami Namin. "Forecasting Economics and Financial Time Series: ARIMA vs. LSTM". In: *CoRR* abs/1803.06386 (2018). arXiv: 1803.06386. URL: http://arxiv.org/abs/1803.06386.

[13] *St Louis Federal Reserve Economic Data FRED*. https://fred.stlouisfed.org/. Accessed: 2022-12-11.

[14] *Time Series Forecasting with the Long Short-Term Memory Network in Python*. https://machinelearningmastery.com/time-series-forecasting-long-short-term-memory-network-python/. Accessed: 2022-12-11.

[15] Luigi Troiano, Elena Mejuto Villa, and Vincenzo Loia. "Replicating a Trading Strategy by Means of LSTM for Financial Industry Applications". In: *IEEE Transactions on Industrial Informatics* 14.7 (2018), pp. 3226–3234. DOI: 10.1109/TII.2018.2811377.

[16] R. Tsay. *Analysis of Financial Time Series*. 2005. ISBN: 9780471690740.

[17] Ruey S. Tsay. "Time Series and Forecasting: Brief History and Future Research". In: *Journal of the American Statistical Association* 95.450 (2000), pp. 638–643. ISSN: 01621459. URL: http://www.jstor.org/stable/2669408 (visited on 12/12/2022).

[18] *Yahoo Finance SP 500*. https://finance.yahoo.com/quote/. Accessed: 2022-12-11.

[19] George Udny Yule. "VII. On a method of investigating periodicities disturbed series, with special reference to Wolfer's sunspot numbers". In: *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* 226.636-646 (1927), pp. 267–298. DOI: 10.1098/rsta.1927.0007. eprint: https://royalsocietypublishing.org/doi/pdf/10.1098/rsta.1927.0007. URL: https://royalsocietypublishing.org/doi/abs/10.1098/rsta.1927.0007.