Hybrid Model for Stock Price Correlation Coefficient Prediction (April 2020)

Gaurav Thapliyal,
Computers and Information Science and Engineering Department,
University of Florida,
gthapliyal@ufl.edu

***Abstract*—Auto Regressive models such as Auto Regressive Integrated Moving Averages (ARIMA), Seasonal Auto Regressive Integrated Moving Averages with extra variables (SARIMAX) have been used for timeseries forecasting. Recurrent Neural Networks is cutting edge technology that can challenge traditional models in time series forecasting. In this project, a hybrid of Seasonal Auto Regressive model and Long Short-Term Memory Recurrent Neural Network is proposed which can be used for determining stock price correlation coefficient. This model aims to capture linear residual values using Auto Regressive models and nonlinear residual values using Long Short-Term Memory Recurrent Neural Network for Stock Correlation Coefficient prediction.**

***Index Terms*—SARIMAX, LSTM, Asset Correlation Coefficient, Portfolio Selection**

I. Introduction

Stock price of a company is the direct indicator of its market value. An organizations' total net worth is generally defined as '*market cap*'. Market cap is calculated by taking a product of Stock price per share with total number of shares. However, stock price is highly volatile in nature. A number of factors affect it, and it can change with high frequency. For example, shares of companies such as Cisco, Boeing Airlines are down due to ongoing coronavirus, as demand for their products has reduced significantly. Contrastingly, shares for companies such as Netflix and Gilead Sciences are up, again due to the effects of novel coronavirus. Even a small change of fraction of percentage might result in huge change in market cap of the company. Apple, one of the Silicon Valley tech giants had its market cap fluctuate from 1.174 trillion dollars to 1.226 trillion dollars in just a short period of time from 21st April 2020 – 28th April 2020. Thus, market cap of a company is a really critical sign for an investor to consider before creating a portfolio.

In addition to individual market caps, there are other indicators which represent market value of a group of companies, bunched together as a whole. Stock markets in United States of America use a number of these indexes, but the most prominent ones are NASDAQ composite, Dow Jones Industrial Average and S&P500, S&P500 is a collection of nearly 500 companies which represent majority (~85%) of market value of United States stock market. It is a very powerful indicator and is used often by investors to determine a powerful investment strategy at the time of portfolio creation. Due to high volatility, it is very difficult to accurately predict values of indexes or even individual stock prices of a company. At any given time, market is being controlled by a huge number of unknown variables which have a huge impact on the tendency of prices to fall or rise. However certain assumptions are put in place before making the predictions, about the market.
Some assumptions of the assumptions are as follows:

1. Even though there are a lot of unknown variables that drive the volatility of the stock prices, any market is not completely random.
2. History repeats itself in the stock markets.
3. People's behavior in the market is rational. Meaning an investor would want to profit from their investment rather than generating a loss.
4. Perfectness of market meaning the market is perfectly competitive. Meaning each buyer and seller have equal opportunity in the marker.

In 1952, Harry Markowitz put forward a theory for portfolio optimization known as '*Modern Portfolio Theory*' [1]. It is a mathematical model which effectively selects assets which do not move in the same direction. For this, a mean-variance strategy is used, which utilizes means of returns and variances of portfolios. During portfolio creation, risk and return evaluation is done, investor then charts efficient frontier which is a curve that maps all the risks and returns so that optimal portfolios can be separated from the sub optimal portfolios. Investor then selects efficient portfolios lying beneath the curve of efficient frontier. For this pioneering work, Harry Markowitz was awarded with Nobel prize in Finance. However, Modern Portfolio Theory works on a lot of assumptions which resulted in criticism of the theory later. For

example, the theory assumes a static correlation coefficient which is used for risk estimation. However, there have been contradicting studies such as [2] which suggest that it indeed is varying. There have been alternatives to constant correlation model, such as Multi Group Model [3] in which Constant Correlation Model is applied for different business units of the assets, Constant Correlation Model [4] where all the assets are assigned equal values for correlation coefficient. We will refer to them collectively as historical models.

Apart from historical models, there are many ways in which problem of predicting correlation coefficients can be approached. Given the correlation coefficients data can be represented as a time series, models belonging to ARIM A family - Auto Regressive Integrated Moving Averages (ARIMA), Seasonal Auto Regressive Integrated Moving Averages (SARIMAX) can be used forecast values of the future correlation coefficients of stocks. Another popular option of a purely statistical model is Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model which can be used to predict the coefficients. However, these models assume that time series of stock price is a linear dynamic system. In reality, however time series data of stocks is a complex and highly volatile dynamic system.

Neural networks fit the bill perfectly as they are enabled to handle nonlinear dynamic systems. Simplest type of neural networks is Feed Forward Neural Network (FNN). However, FNN's only depend on the current inputs and tend to ignore historical data. Thus, using FNN's to forecast correlation coefficients of stock prices would be counter intuitive with regards to historical nature of market. Hence, instead of FNN a different type of neural network known as Recurrent Neural Network (RNN) can be used to forecast the correlation coefficients. It has been shown that neural networks perform significantly better than the traditional historical models and statistical models in terms of performance and accuracy. A combination of linear models and neural networks can be used to earn more efficient results and is the underlying ground framework for this study.

Rest of the paper is organized as follows. Section 2 describes prior work done in this area. Section 3 describes the problem more broadly and also defines the need of SARIMAX-LSTM model. It also explains how it is different from the related models developed in this area. Section 4 describes the flow of project. Section 5 contains detailed description and working of each module. Section 6 presents result of ARIMA-LSTM and SARIMAX-LSTM in comparison with traditional models. Section 7 explores possibility of future work concludes the paper.

II. Prior Work

Numerous studies have suggested that a combination model of statistical models and neural networks can be used to get a better model than using them individually. Peter Zhang [5] proposed an ARIMA and RNN hybrid model which can be used to capture linear as well as nonlinear aspects of time series and predict more accurate correlation coefficient than the aforementioned models. Ray Nelson and James Hansen [6] also have studied these models. Hyeong Kyu Choi [7] also proposed a similar hybrid model.

III. Description

Model proposed in this paper builds upon the work Hyeong Kyu Choi [7]. In his literature, he proposes an ARIMA-LSTM hybrid model, where LSTM is Long Short-Term Memory neural network. LSTM networks specialize in learning long term dependencies. LSTM's commonly find applications in tasks such as handwriting recognition, Intrusion Detection Systems, etc. They are perfect for deriving predictions from time series data. It is known that LSTM-ARIMA hybrid models perform better than traditional historical models and even standalone ARIMA or LSTM model. However, LSTM-ARIMA model does not take into account seasonality of the stock price time series. ARIMA models eliminate trends from the time series using a process called differencing. Differencing is similar to taking derivative with respect to time, however time is considered discrete. This a very efficient process which eliminates trends from the time series data. However, as we have previously assumed that history tends to repeat itself in a market, seasonality also introduces trends in the time series data. For example, figure 1 below shows S&P500 index trend over 6 years from 2015-2020. Barring a few outliers, we can deduce following points while looking at the figure:

Fig. 1. SP500 index values over 5 years. X axis – years, Y axis – prices in dollars. Courtesy of "finance.yahoo.com"

1. On average, index value rises every year. Exception being effects of coronavirus pandemic in year 2020.

2. First quarter of the year experiences a rise in the index value and peaking somewhere in second quarter.
3. Third quarter experiences a downward trend which stays till the month of September/October. Remainder of 4th quarter experience an upward trend resulting in a strong finish at the end of year.

Clearly there are seasonal trends in the time series data and ARIMA model is not capable of removing the seasonality of data. Thus, a more general Seasonal Auto Regressive Integrated Moving Averages with extra variables (SARIMAX) model can be used. SARIMAX makes use of additional variables to seasonally difference the time series data, thus removing the seasonal trends. Thus, a LSTM-SARIMAX hybrid model should be more accurate than the existing LSTM-ARIMA model.

In this project, I've implemented both ARIMA-LSTM hybrid model and SARIMAX-LSTM hybrid model to forecast predictions on Stock Price Correlation Coefficients. Aim of this project is to create an autonomous application, which can help professionals such as Quant Developers, Traders, Investors, who are interested in the field of stock market finance, for portfolio optimization. The project not only runs the models, but also crawls internet to scrape stocks data from data sources on the internet, hence creating an end to end solution.

IV. Methodology

The project follows the general steps of a Data Science Project Life Cycle. Figure 2 shows the flow diagram for major modules in the project. Following is a description of steps that are executed to obtain the final scores and predictions.

1. We start off with the data scraping module. This module pulls out tickers of all the S&P 500 firms from the Wikipedia Page of S&P 500 firms. Then using these tickers individual stock data is extracted. (**Data Acquisition Step**)

2. Perform Extract, Transform and Load process on data to get clean, processed data in the Data Wrangling module. Create portfolio by selecting tickers and impute data if required after checking for missing values. Explore correlation between stock prices. Load the data in the desired form of Correlation matrix. (**Data Preparation / Exploration Steps**)

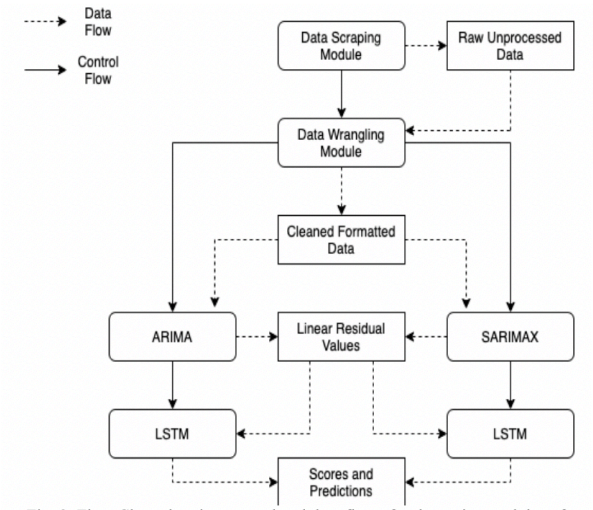

Fig. 2. Flow Chart showing control and data flows for the major modules of the project

3. In the ARIMA/SARIMAX module perform linear evaluation and generate residual values. (**Modelling Step**)
4. In the LSTM-ARIMA/LSTM-SARIMAX module perform non-linear predictions and evaluate the scores. (**Modelling/Evaluation Steps**)

V. Conceptual Design

Each module of the project will be described in detail here. Additional modules not present in the flow chart will also be described.

A. Data Scraping Module

This module makes use of Beautiful Soup library and alpha_vantage library to crawl and scrape data from the web. First off, we need tickers of all the S&P500 companies. This is done with the help of Beautiful Soup, using which we pull of all the tickers and company names from the S&P500 Wikipedia page. It does so by fetching the whole table from the S&P500 web page. 'lxml' parser is used to convert the data to text. Once we have a list of all the tickers, now we proceed to download stock data for each individual stock ticker using the Alpha Vantage API. A key is required to access the data source and I generated a Selenium and JS script to help me automate the process for creating 10 accounts and generate keys which are stored in a text file. Next we pull the time series data for each ticker using this key. Alpha Vantage only allows 5 API access per min per key, so we shuffle between 10 keys maintaining their status in a queue. Once the buffer size equals 5 keys, we introduce a cool off period of 1 min and reset the buffer. Data for each ticker is stored individually in a csv.

B. Data Wrangling Module

Raw data extracted in the Data Scraping module is preprocessed here to obtain clean data. First, we determine

tickers for which data is present for 10 years starting from 2010. Next we pull the adjusted close value for all the tickers and combine those columns in a csv. This data is then checked for redundancies by plotting the missing values and against the ticker. If more than 5 values are missing, then delete that ticker else we impute values by putting in the price of next day if the price of current day is missing. After this the values of the matrix are stored in a csv.

C. ARIMA Module

Clean data after data wrangling is now ready for modeling. ARIMA module first divides the data in 4 sets train, dev, test 1 and test 2. Dividing data in 4 sets allows for model correction during train and dev set execution, thus achieving more accuracy in predictions. This is done using rolling window of 100 days. We start of at 5 different points, then from those points we assume a window of 100 day each and of 100 strides. This results in a time series of size 55875 sets. Each of these sets contain 24 time period data. These sets form the train/dev/test sets and are divided into 4 equal parts. After this we start fitting ARIMA model.

ARIMA is a more generalized version of Auto Regressive Moving Averages Model. ARIMA is generally applied on data with non-stationarity to eliminate those trends with the method of differencing. A typical ARIMA model is specified as ARIMA (p, d, q) where p, d and q stand for dimensions of Auto Regressive, Integrated and Moving Averages respectively. Auto Regressive part specifies that variable is regressed on its own historical or lagged values. Integrated specifies that current data values be replaced by historical value. Moving averages part specifies error generated during regression is a linear combination of historical error terms. For example, ARIMA (1,1,1) will perform each of this step once while model fitting. Ideally, auto ARIMA with an iterator is used to try out different variations of ARIMA, however looking at the data (2,1,0) models seems the best fit. We will be using 5 different models of ARIMA to fit the data and they are as follows:

- ARIMA (0,1,1)
- ARIMA (1,1,0)
- ARIMA (1,1,1)
- ARIMA (2,1,1)
- ARIMA (2,1,0)

Ideally parameters are optimized using a process in which different types of indicators are used. One such indicator is Akaike Information Criterion or AIC. AIC is a direct measure of the quality of a model. We have used the common ARIMA parameters to fit the model, however we iterate between them using AIC. We start off with model ARIMA (0,1,1) and ARIMA (1, 1, 0). If AIC of prior model is better than the later one, we use the first one, else we go for second and compare it with others, for each data set of size 24 time periods. ARIMA (2, 1, 0) is the last resort and is used as the default values if AIC values for all the other models are deemed not good. After model fitting, residual value is stored as train/dev/test data for LSTM module.

D. SARIMAX Module

As described in description section, ARIMA models are limited against data that shows seasonality. To handle seasonality, extra parameters are required for seasonal regression as well. SARIMAX models thus are represented using 7 parameters. A typical SARIMAX model can be shown as (p, d, q) (P, D, Q) m where p, d and q are standard ARIMA descriptors for AR, I and MA. P, D, Q are the seasonal regression parameters and m signifies period of seasonality. Since our stock correlation coefficients time series shows a periodicity every 12 months or in a year, we keep the value of m = 12 for all the SARIMAX models. Similar to ARIMA module, correlation coefficient calculation is done using rolling window. This generates 55875 set which are again divided similarly in 4 equal parts train, dev, test1 and test 2. Following common SARIMAX models will be used in the model:

- SARIMAX (0, 1, 1) (0, 1, 1) m =12
- SARIMAX (1, 1, 0) (1, 1, 0) m =12
- SARIMAX (1, 1, 1) (1, 1, 1) m =12
- SARIMAX (2, 1, 1) (2, 1, 1) m =12
- SARIMAX (2, 1, 0) (2, 1, 0) m =12

Perfect fit model will be determined using Akaike Information Criterion. If all the models are deemed unfit for a particular set, model reverts to SARIMAX (2, 1, 0) (2, 1, 0) m =12 model as the default model and continues with the forecasting.

E. LSTM Module

Neural networks are a paradigm of computing environments which are capable of learning without being programmed for a specific task. They are simply a collection of neurons which can learn to perform some task without being specifically programmed for that task. Neural networks learn with the help of a process known as training. Once a neural network is adequately trained, it can be used to solve a specific set of problems with a certain accuracy. Its designed similar albeit on a very low scale, to a human brain wherein humans think and learn to solve a problem by training on solving similar set of problems. Neural networks are designed in layers, first layer receives the input, second layers of input process something. This goes on till the last layer which produces an output. Layers are comprised of neurons. In Feed Forward

Neural Network, neuron receives input, the output of previous neurons it is connected to. These values are multiplied by a weight and summed up. Weights are determined during the training. On this summation, applying a function known as the activation function generates the output. A process known as backpropagation helps in identifying the correct weights. Backpropagation basically involves propagating back to a neuron to determine change in output if weight was changed.

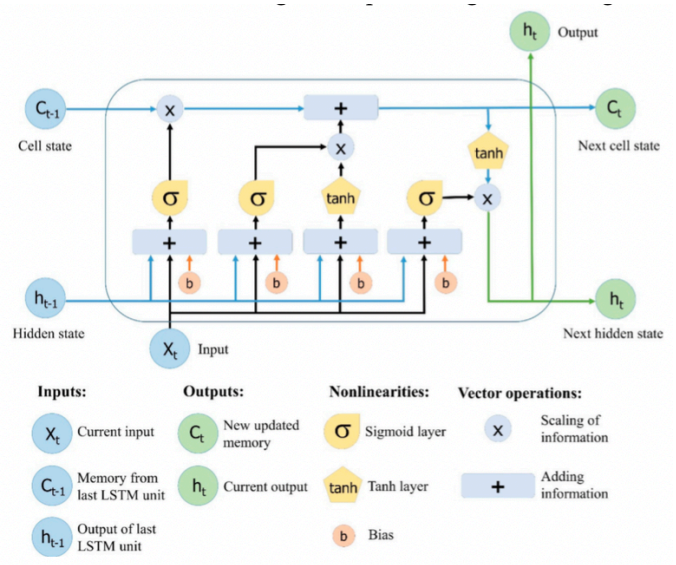

Figure. 3. Structure of a LSTM cell. Courtesy of "www.mdpi.com"

Long Short-Term Memory or LSTM is a special type of neural network which is also a Recurrent Neural Network (RNN), meaning they are stateful neural networks which perform a task over and over again keeping in mind the output of previous task. Figure 3 shows a typical cell in a LSTM neural network. LSTM used in this module depends on back propagation to calculate weight matrices and is composed of a forget gate. tanh function or the inverse tangent function generates value ranging from -1 to 1 and hence fits our regression task perfectly.

Sigmoid function behaves as the activation function for our LSTM input gate and takes input (h_{t-1}) from the previous LSTM cell along with the input values X_t , at the time t. Sigmoid function is responsible for determining which part of information from previous output and current input should be kept. Thus mathematically, output of sigmoid can be shown in the form of following equation:

$$f_t = \sigma(W_f\,[h_{t-1}, X_t] + b_f).$$

This is the structure known as forget gate. W_f is the weight matrix and b_f is the bias for forget gate. Bias is a quantity which is pre-determined, before training starts. Next, as sigmoid function has output 0, 1, it decides whether to keep the value or not. If yes, then those values are passed on to the tanh function so that appropriate weight is assigned. After weight is assigned, the values are stored in memory new cell state C_t is created.

$$i_t = \sigma(W_i\,[h_{t-1}, X_t] + b_i),$$
$$N_t = tanh(W_n\,[h_{t-1}, X_t] + b_n),$$
$$C_t = C_{t-1}\,f_t + N_t\,i_t.$$

Once new cell state is saved, again a sigmoid function decides the values which will be passed out in the output. These values again get a weight assigned by the tanh function.

$$O_t = \sigma(W_o\,[h_{t-1}, X_t] + b_o),$$
$$h_t = O_t\,tanh(C_t).$$

Thus, h_{t-1} and C_t are passed on to the next cell. Output gate thus uses an inverse tangent function as the activation function. This module implements a LSTM using the Keras library in python. The model is evaluated based on Mean Square Error and Mean Absolute Error Metrics. Optimizer used is adam. Module responsible for performing model fit, evaluation and prediction

F. TEST_ASSET Module (SARIMAX-LSTM/ARIMA-LSTM)

There are 2 modules, one for SARIMAX-LSTM, other for ARIMA-LSTM. These modules are used to test the final models generated on data outside train/dev/test set. This module serves 2 purposes

- Check the correctness of models
- Compute score and prediction values for the new data.

G. Model Evaluator Module

Generate score values for Single Index model, Historical model and Constant Correlation model. Model uses same data which was used in dev and test phases while evaluating ARIMA-LSTM and SARIMAX- LSTM models.

VI. EVALUATION

Models are evaluated on the basis of Loss, Mean Square Error, Mean Absolute Error and prediction values. Learning curves for SARIMAX-LSTM and ARIMA-LSTM are as follows:

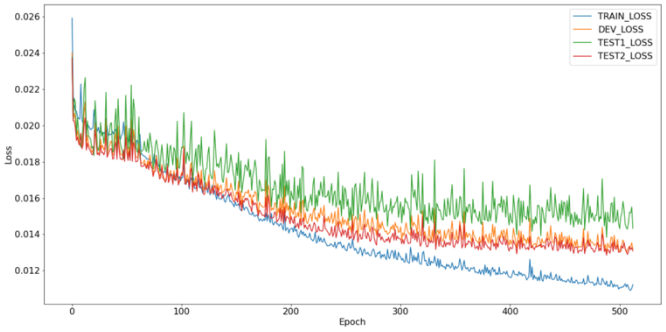

Fig. 4. SARIMAX LOSS

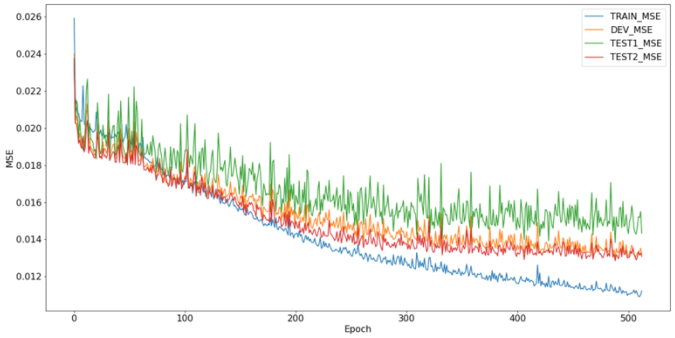

Fig. 5. SARIMAX MSE

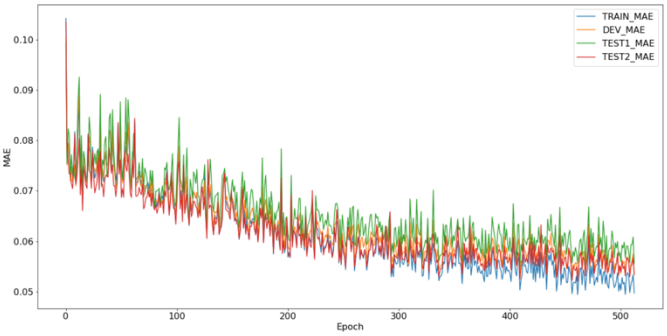

Fig. 6. SARIMAX MAE

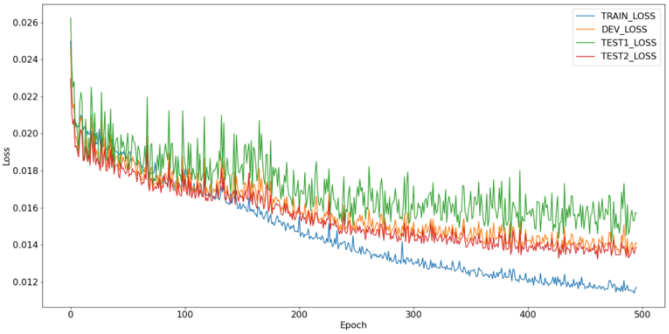

Fig. 7. ARIMA Loss

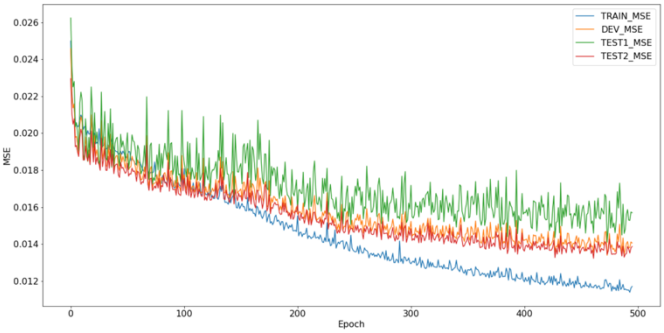

Fig. 8. ARIMA MSE

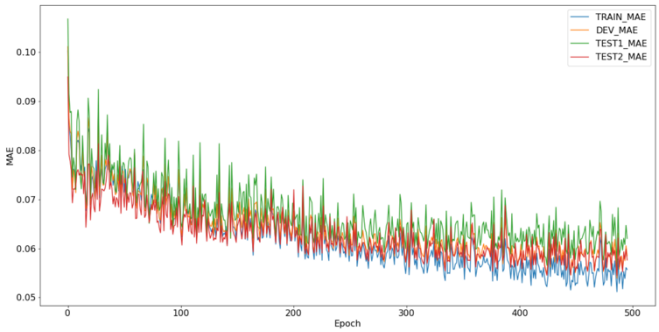

Fig. 9. ARIMA MAE

SARIMAX-LSTM slightly edges over ARIMA-LSTM. For SARIMAX-LSTM LOSS, MSE and MAE values start converging around **278th** epoch and for ARIMA-LSTM values converge around **297th** epoch. SARIMAX-LSTM hybrid model also yields a tighter scatter between actual and predicted values, than ARIMA-LSTM.

Fig. 10. SARIMAX Test 1 Scatter

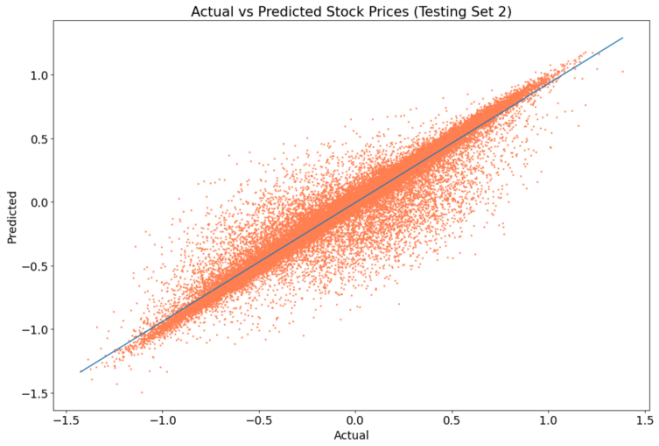

Fig. 11. SARIMAX Test 2 Scatter

Fig. 12. ARIMA Test 1 Scatter

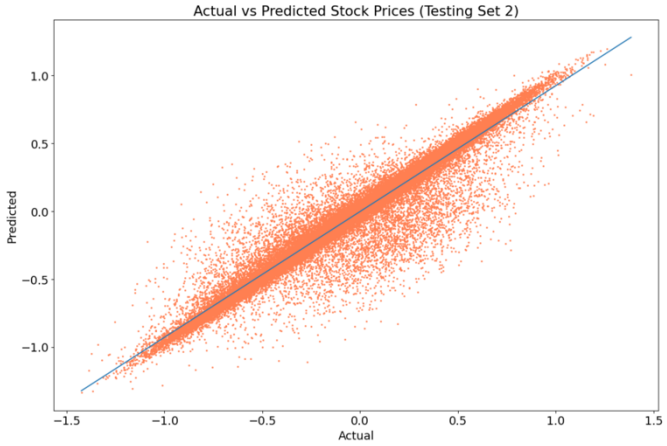

Fig. 13. ARIMA Test 2 Scatter

After model generation, both the models were evaluated on data which was not present in the 150 companies of train/dev/test sets. This is a relatively smaller data set of only 20 firms and 180 stock prices yielding following results:

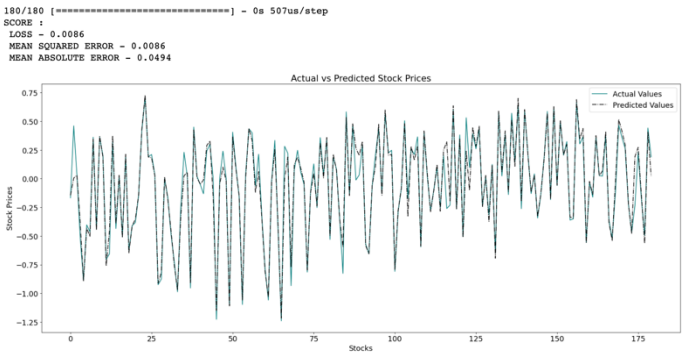

Fig. 14. Actual vs Predicted values, Score values for ARIMA-LSTM

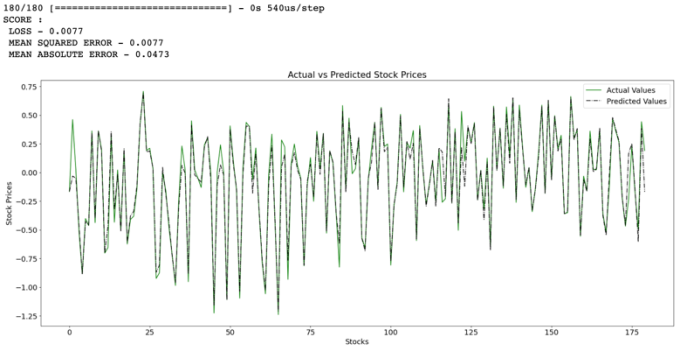

Fig. 15. Actual vs Predicted values, Score values for SARIMAX-LSTM

SARIMAX-LSTM also performs better against traditional models by a huge margin. Following table shows the score values for ARIMA-LSTM, SARIMAX-LSTM and other traditional models.

Models	Dev MAE	Test 1 MAE	Test 2 MAE
SARIMAX-LSTM	0.06132	0.06279	0.05945
ARIMA-LSTM	0.06219	0.06434	0.06408
Single Index	0.46552	0.50147	0.48905
Historical	0.53908	0.52796	0.51641
Constant Correlation	0.43044	0.39848	0.40834

Table. 1. MAE scores for all the models.

Models	Dev MSE	Test 1 MSE	Test 2 MSE
SARIMAX-LSTM	0.01422	0.01522	0.01377
ARIMA-LSTM	0.01440	0.01529	0.014332
Single Index	0.31492	0.37543	0.37240
Historical	0.44573	0.43676	0.41686
Constant Correlation	0.24601	0.21334	0.25763

Table. 2. MSE scores for all the models.

SARIMAX-LSTM outperforms ARIMA-LSTM even though scores are for later epoch for ARIMA-LSTM, signifying SARIMAX-LSTM scores converge faster. Not only does SARIMAX-LSTM beats ARIMA-LSTM but also beats historical models by a huge margin.

VII. Summary and Conclusion

Proposed a SARIMAX-LSTM model for predicting stock price correlation coefficient. Evaluated different existing models on the same data set. Results show that proposed model outperforms every other model. Hybrid models such as ARIMAX-LSTM, SARIMAX-LSTM show great promise. Other models such as those belonging to GARCH family can also be studied and used for deriving linear tendencies.

Custom activation functions in LSTM is another area which can be explored to provide more efficient scores. Eventually, hybrid models like these can also be extended to asset management as whole.

REFERENCES

[1] H. M. Markowitz. Portfolio selection. The Journal of Finance, Vol.7, No.1:77–91, Mar. 1952.

[2] E. Jondeau F. Chesnay. Does correlation between stock returns really increase during turbulent periods? Economic Notes, Vol.30, no.1-2001:53– 80, 2001.

[3] M. W. Padberg E.J. Elton, M. J. Gruber. Simple rules for optimal portfolio selection: The multi group case. Journal of Financial and Quantitative Analysis, 12(3):329–349, 1977

[4] T. J. Urich E. J. Elton, M. J. Gruber. Are betas best? Journal of Finance, 33:1375–1384, 1978.

[5] G.P. Zhang. Time series forecasting using a hybrid arima and neural network model. Neurocomputing, 50:159–175, 2003.

[6] R. D. Nelson J.V. Hansen. Time-series analysis with neural networks and arima-neural network hybrids. Journal of Experimental and Theoretical Artificial Intelligence, 15:3:315–330, 2003.

[7] Stock Price Correlation Coefficient Prediction with ARIMA-LSTM Hybrid Model, Hyeong Kyu Choi, 808.01560, arXiv