Assessing the ARIMA-LSTM Model on Long-term Stock Predictions

Word Count: 5266

Introduction

In the finance world, the stock market is one of the most unstable models in the economy, making it incredibly difficult to predict when to invest. Especially with the economic recession of 2008, households responded by increasing their savings, which reduced stock investments in several companies by nearly 15% (Farmer 2011, 7). Even today, small-time traders feel reluctant to invest as a recent 2019 poll estimated that only 55% of adult Americans reported placing money into the stock market, which significantly dropped from 63% in 2007 (Saad 2019, 1). Though many former traders chose to back out of the market, financial engineers, on the other hand, found hope through a riskier approach called prediction models. Economists like New York University's Professor Charles Tapiero express how these models can interpret nonsensical data into accurate stock predictions, assisting investors in making the right financial decisions (Tapiero 2013, 7). With Tapiero's philosophy and recent studies producing such prototypes, two major fields of study have risen in popularity - statistical and technical stock analysis.

Most traditional investors favor the classical approach of statistics. Statistical analysts tend to create algorithms, like moving averages, around previous stock prices to predict future trends. Many financial engineers have developed iconic mathematical techniques like ARMA-GARCH models and the Box and Jenkins approach that have been praised in the stockbroker world for several decades (Mohankumari et al. 2019, 2). However, traders nowadays struggle to build more efficient statistical algorithms as they are outdated and only focus on close price data. Luckily, in the technological era, modern investors discovered another unique approach called artificial intelligence (AI). AI models, like machine learning, can incorporate multiple sets of data such as trade volume, high and low prices, etc. Better yet, AI can simulate the human mind by employing neural networks to create algorithms on its own that identify patterns between data (Dong et al. 2020, 1). In terms of stock analysis, traders can utilize AI's recommendation software to gather data and quickly retrieve predicted stock prices. Despite its potential, AI still remains unreliable to investors as many assume current machine learning designs often overestimate stock prices more than statistical approaches do (Tapiero 2013, 9).

Regardless of each fields' drawbacks, there is still much to be learned about both approaches. It's imperative that financial engineers continue investigating the potential of statistical and technical models to establish a reliable base model that investors can depend on. Hopefully, with the experimental results of this paper, traders will be motivated to participate in the stock market by incorporating both models when making their own financial investments.

Literature Review

Statistical analysis is defined as the computational method that financial engineers implement to extract and manipulate data through arithmetic formulas (Sharma et al. 2015, 2). Thanks to numerous studies and support the field has received, traders witnessed the potential of statistics and implemented these features when forecasting future stocks.

For example, Karim Elbahloul, a renowned Professor from New Jersey State University, examined the accuracy of several statistical models against the popular ARIMA model in stock forecasting. The Autoregressive Integrated Moving Average Model (ARIMA) is simply an algorithmic combination of moving average formulas subtracted from extreme outliers (Jansson et al. 2020, 4). In Professor Elbahloul's experiment, he tested how accurate the ARIMA, ADCF, and exponential smoothing algorithms performed with the S&P 500 stock's previous close prices (Elbahloul 2019, 3). After thorough analysis, the ARIMA statistical model trumped all other designs; it obtained an average of 81.9% prediction accuracy when forecasting the last five days of S&P 500's stock. Many future studies branched off from Elbahoul's conclusions and replicated ARIMA models similar to his experimental design. Each of them unanimously acknowledged that the ARIMA design is a consistent model that takes "little to no time to compute data" (Sharma et al. 2015, 12).

In 2014, graduate student Saptarsi Goswami, from the International Journal of Computer Science, sampled the ARIMA algorithm on fifty-six distinct Indian stocks. The model received an astounding 83% prediction accuracy (Goswami et al. 2014, 17). However, Goswami later recognized that the model suffered from high training and testing errors as the model had a mean square error of 54.823. Training and testing errors (prediction errors) are defined as the likelihood that a prediction model will overestimate or underestimate the data provided between training and testing phases (Jansson et al. 2020, 7). If the mean square error is high, the data will likely take much longer to assess, and it will be significantly more error-prone to overestimating predictions. Overall, Goswami corroborates with traders like Professor Elbahloul that ARIMA has a promising future; however, it requires in-depth research to improve its prediction errors.

Though there is a general consensus among traditional economists that the ARIMA model is reliable to some extent, modern traders say otherwise. According to a research study in 2014 by Professor Ayodele Adebiyi at Covenant University, he opposes claims by Elbahloul and Goswami; he asserts that current studies focus solely on stable stocks and completely disregard volatile stocks (Adebiyi et al. 2014, 3). Analysts need to establish a model that can determine price predictions for stocks that frequently change, especially now as the last two decades have seen several inflations and recessions (Tapiero 2013, 19). For the purpose of this paper and Adebiyi's research, volatile stocks are defined as stocks with an average day-to-day percent change greater than +/- 10%, while stable stocks have a daily percent change within +/- 10%.

Additionally, Adebiyi explains that analysts' models should determine long-term price predictions too. This feature will enable investors to gauge when prices will rise and fall within months rather than just over a week. To test his hypotheses, Professor Adebiyi and his team gathered data from NYSE (volatile) and NSE (stable) stocks to assess how their optimized ARIMA model would perform against both. He divided 90% of the stock data into the training set and applied the remaining 10% to the testing set to compare ARIMA outputs with the actual prediction values. After three trials, the experiment affirmed that the model achieved a staggering 94% prediction accuracy for short-term predictions with NSE. However, it only received a 49% prediction accuracy when tested for long-term forecasts. The NYSE trials also resulted in similar results, but prediction/training errors on this volatile stock peaked at almost 212.78 MSE. Though Adebiyi agrees with traditional economists like Elbahloul and Goswami that ARIMA "has a strong potential for short-term prediction," he concluded that statistical approaches are prone to error for long-term volatile stock predictions (Adebiyi et al. 2014, 8).

Fortunately, modern economists who mirrored Adebiyi's view furthered stock prediction through another revolutionary approach called Artificial Intelligence (AI). Unlike statistical methods that focus on only one algorithm, machine learning allows models to be dynamic. AI can adapt and identify patterns in data with its own optimized algorithms (Dong et al. 2020, 7). Though it is a new concept, many notable studies have already explored various AI structures on stock market forecasting. For example, one 2018 study by graduate scholar Navodit Jain from the Massachusetts Institute of Technology experimented with stock prediction on three recursive neural networks (RNNs): basic RNN, LSTM, and GRU models. RNN is defined as an AI recursion model of cycle nodes and links that loop through unpredictable data repeatedly until it finds all the critical data points to make a proper prediction (Makhlouf et al. 2020, 5). Unlike

Professor Adebiyi's statistical model that is limited to only short-term forecasts, Jain's RNN structure can perform long-term predictions. However, he does warn researchers that RNNs suffer from vanishing gradient problems and overlapping hidden layers, which may diminish the amount of data the model can store. Jain filled 5700 stock data points into each model from Carriage Services Stock and witnessed that the basic RNN performed the worst of all three models in terms of prediction accuracy. Contrary to Jain's hypothesis, the LSTM model performed the best with an average of 79% long-term prediction accuracy. An LSTM model, or Long-short Term Memory, is "a special kind of RNN capable of learning long-term dependencies. It is explicitly designed to avoid the vanishing/exploding gradient problem" (Shakya 2020, 6). Though Jain assumed LSTMs would have much longer time complexities and prediction errors when processing stock data, his results disproved that. Jain ultimately concluded that LSTMs combat the disadvantages of basic RNNs and should serve as viable forecast models for long-term predictions.

However, another study by Professor Adil Moghar branched off Jain's discovery and explored LSTMs for short-term predictions. Moghar's experiments revealed that LSTMs received high prediction errors for volatile and stable stock short-term predictions (Moghar et al. 2020, 6). Additionally, prediction accuracy results only maxed out at 57%, which is significantly lower than the long-term prediction results presented by Jain. Though Moghar's results may devalue the LSTM's effectiveness in stock forecasting, Moghar explained that the results clarified LSTMs are effective only for long-term predictions.

By taking into account these previous studies, some financial engineers recognized that LSTMs and ARIMA models worked like counterparts - each functioning better under a specific type of stock prediction. It could be plausible that combining the two frameworks would eliminate each model's drawbacks. In 2018, one compelling study by Professor Shui Ling Yu accomplished just that by developing the new ARIMA-RNN model for long-term stock forecasting. Even though approaches like this have rarely been tested, Yu's successful hybrid concept demonstrated that it's possible to integrate statistical and technical models. Yu began by producing a moving average formula optimizer that would improve the accuracy of the ARIMA model. Later, he incorporated the ARIMA results into the RNN structure as a smoothing constant to help the RNN predict more efficiently. As a result, the ARIMA-RNN achieved typical training errors of approximately 15.76 MSE. On the other hand, the model only reached a maximum of 63.49% prediction accuracy. Though the output was not extraordinary, Yu explained that the RNN base models suffer from vanishing gradients and low data storage, which is precisely what Jain discovered in his RNN experiment. Additionally, the ARIMA-RNN had a significantly better prediction accuracy for volatile stocks compared to Adebiyi's ARIMA or Jain's LSTM model results. Regardless, Yu believed further research in advancing the AI component's memory would enhance the hybrid approach for long-term stock predictions.

Even though previous studies like Professor Yu's ARIMA-RNN model indicated the potential for a hybrid model, there remains a notable research gap in long-term stock prediction using the LSTM model with ARIMA. Based on research conducted by Jain and Moghar, RNN structures are rudimentary and unable to cope with immense amounts of data to create efficient algorithms. This could be why Yu's ARIMA-RNN yielded a much lower prediction accuracy than LSTM structures. According to Jianyu Qiu from Dalian University's Advanced Design and Intelligent Computing Lab, LSTM avoids long-term dependence issues due to its unique storage unit structure, which helps it predict stocks significantly faster than basic RNNs (Qiu et al. 2020, 12). Therefore, I believe an LSTM would fit better with an ARIMA model's data to yield stronger prediction accuracies than that of the ARIMA-RNN approach.

To address this gap, I arrived at the following essential question: How will integrating ARIMA models with LSTMs influence prediction accuracies and errors when predicting long-term stock prices of volatile and stable stocks? I hypothesized that the ARIMA-LSTM hybrid would report higher long-term prediction accuracies and lower training and testing errors compared to the individual ARIMA and LSTM models for both volatile and stable stocks. To test this, I centered my study on using the ARIMA-LSTM to test two volatile stocks, Amazon and Starbucks, and two relatively stable stocks, Apple and Carnival. I selected these particular stocks as they fulfilled the requirements for volatility or stability (+/- 10%) and served as accurate sample representations of what today's stock market looked like. Hopefully, this paper's new understanding will shed light on long-term hybrid prediction models and inspire financial engineers on novel stock forecasting approaches.

Methodology

Experimental Design

To compare the ARIMA-LSTM's prediction power for these four stocks to control groups like the ARIMA and LSTM alone, I realized that I would need to conduct an experiment. After thoroughly examining previous studies, I finally came across a feasible machine learning stock prediction procedure by Professor Alberto Rossi from the University of Maryland. By eliminating the need for a pretest and random assignment, Rossi expressed that AI models are already inherently random participants that do not require these elements for a proper experimental design (Rossi 2018, 22). Consequently, Rossi suggested a quasi-experimental procedure that only compared control groups with the experimental group in the posttest analysis, which is called the experimental posttest control only group design. Additionally, the quasi-experimental design would allow me to experiment with a specific explanatory variable - whether or not the model includes the ARIMA and LSTM combination - and record the response variables of prediction accuracy and error. Best of all, the procedure is easily replicable for multiple trials, allowing me to determine a stronger cause-and-effect relationship. Therefore, I modeled my study similar to Rossi's approach by establishing one experimental group, ARIMA-LSTM, and comparing it to two control groups, ARIMA and LSTM. Yet, to conduct this procedure, I first needed to extract relevant stock data and construct each model.

Finding Stock Data

I ultimately decided on testing four distinct stock values that would accurately compare each model's performance: Amazon (AMZN) and Starbucks (SBUX) as volatile stocks and Carnival (CCL) and Apple (AAPL) as stable stocks. AMZN is highly volatile, with a recent upgrowth of 43% in stock value (Sharma et al. 2015, 3). If this stock is experimented with, the trials will show how each group in the experiment works against rising stock prices. Conversely, Starbucks stock is dropping at exceeding rates with almost a "32% decline in sales from the year 2019 to 2021" (Divine 2020, 3). Therefore, experimenting with SBUX would help me monitor my prediction models against crashing prices. Finally, Apple's and Carnival's stocks are relatively stable; thus, their stock data and corresponding prediction results were used to compare model performances between volatile and stable stocks.

To collect the stock data, Professor Rossi described an efficient approach through a python programming API called "yFinance." By using the Google Colab interface and a Python package called "pandas_datareader" to read the yFinance database, Rossi was able to fetch several stock measurements like volume, high and low prices, etc. (Rossi 2018, 13). I favored Rossi's procedure as it achieved my research study's main requirement of collecting updated stock data while also being manageable for an amateur coder like me to accomplish. I emulated this process and collected over 7,323 records of close prices from 01/04/2000 to 01/27/2021 for each stock and recorded them into a python readable CSV file. Finally, I used a percent change formula to determine whether a stock price decreased or increased between each day. These percent changes were necessary for finding which prediction model measured the stock better on days with high or low volatility (see the discussion's "Data Volatility to Prediction Relationships" for more information).

$$percent\ change = \frac{close\ price\ -\ previous\ close\ price}{previous\ close\ price} x100$$

Figure 1: Percent Change Formula

Stock Ticker: CCL

Date	High	Low	Open	Close	Volume	Pct Change
2000-01-04	47.562500	46.187500	47.500000	46.312500	1061600.0	-0.012000
2000-01-05	48.562500	46.312500	46.312500	47.875000	1674400.0	0.033738
2000-01-06	49.312500	47.125000	47.812500	48.562500	1437300.0	0.014360
2000-01-07	48.562500	47.000000	47.000000	48.000000	1813900.0	-0.011583
2000-01-10	50.812500	48.437500	48.500000	50.125000	1888900.0	0.044271
...	...	...	...	...	...	...
2021-01-21	20.879999	20.320000	20.700001	20.740000	20241500.0	-0.001925
2021-01-22	20.430000	20.040001	20.309999	20.219999	24702300.0	-0.025072
2021-01-25	19.809999	18.730000	19.799999	19.219999	55839500.0	-0.049456
2021-01-26	19.510000	18.620001	19.400000	18.719999	43961700.0	-0.026015
2021-01-27	19.415001	17.960100	18.410000	19.040001	25129596.0	0.017094

Figure 2: Example of Sample Extracted Stock Data

ARIMA Model

Now that I collected each stock's dataset, I developed the ARIMA model first. I structured my ARIMA model by following a similar method that Professor Yu used for his

ARIMA-RNN. Yu explained that certain stocks have a specific type of ARIMA algorithm that works better than others at estimating future close prices. Researchers have to create a moving average filter to find which formula works best for a given stock (Yu et al. 2018, 17). Therefore, Yu incorporated a statistical comparison test called kurtosis, which is defined as the measure of outliers present in a data distribution (Yu et al. 2018, 16). The optimal kurtosis value should be equal to three, and the moving average filter should determine if the formula, when run with the training data, would have a kurtosis of three. The five moving averages I tested were:

Simple Moving Average (SMA) - calculates an average of the last n prices, where n represents number of periods.	**(P1 + P2 + P3 + P4 + ... + Pn) /n**
Exponential Moving Average (EMA) - a weighted average of the last n prices, where weighting decreases exponentially for previous periods.	**(Close - previous EMA) * (2 / n+1) + previous EMA**
Weighted Moving Average (WMA) - a weighted average of the last n prices, where the weighting decreases based on predetermined weights with each previous price.	**(Price * weighting factor) + (Price previous period * weighting factor-1)**
Double Exponential Moving Average (DEMA) - a double weighted average, where the EMA is calculated twice based on n intervals.	**(2 * EMA(n)) − (EMA(EMA(n)))**
Kaufman's Adaptive Moving Average (KAMA) - moving average designed to account for market noise or volatility and will adjust weight for each swing in prices (monitors by volatility).	**KAMAi-1 + Smoothing Constant * (Price – KAMAi-1)**

Figure 3: Moving Average Formulas Chart

Though Professor Yu's procedure was thorough, it seemed too complicated to achieve, so I utilized another python API called "scipy.stats.kurtosis" that calculated each formula's kurtosis value automatically for me. Overall, the procedure I applied to create the optimal ARIMA can be described through the following:

1. Use a python program to read the dataset of stock prices from the CSV file. Set the first 95% of data as the training set for the ARIMA and place the last 5% as the testing set.
2. Run the training dataset under the moving average filter and loop through each moving average formula to determine each formula's kurtosis value.
3. Identify which algorithm works best based on which formula's kurtosis equals three. Set this algorithm as the final ARIMA model.
4. Test the final ARIMA model to predict future stock prices. If two or more formulas have kurtosis values equal to three, run an ARIMA simulation for each and report the ARIMA results with higher prediction accuracy.
5. Compare the results of the ARIMA predictions with the testing dataset using python's "statsmodels.test()" method. Record the trial's prediction accuracy, MSE, RMSE, and MAPE values.
6. Repeat steps 1-5 for two more trials on the specific stock. Finally, record the average results for prediction accuracy and error.
7. Repeat steps 1-6 for the other three stocks.

The following procedure gives the prediction performance results for the ARIMA model (control group), but the LSTM model also needs to be created to develop the experimental group and second control group.

LSTM Model

Unlike the ARIMA model, the LSTM model is a lot less complicated to create, thanks to Professor Nelson's machine learning approach using Tensorflow. Tensorflow is an open-source python framework by Google that is used for time-series predictions (Nelson et al. 2017, 42). With this python framework, Nelson explains how financial engineers can easily design and manipulate variables, or hyperparameters, of the LSTM. In particular, Nelson references using the default values of 200 epochs, 0.001 learning rate, 40 timesteps, and 35 neurons in each prediction layer for his experiment. (Nelson et al. 2017, 44).

To create my LSTM design, I replicated Nelson's procedure, but I used only 30 timesteps since I only wanted to test a month's worth of stock predictions. The LSTM procedure I followed is described below:

1. Use a python program to read the dataset of stock prices from the CSV file. Set the first 95% of data as the training set for the LSTM and place the last 5% as the testing set.
2. Apply Tensorflow's LSTM into the program and feed the training set into the model.
3. Set the proper hyperparameters and train the model.
4. Use a Tensorflow simulation to test the LSTM's final predictions.
5. Run Tensorflow's "history" feature to retrieve the approximate prediction accuracy, MSE, RMSE, and MAPE values.
6. Repeat steps 1-5 for two more trials on the specific stock. Finally, record the average results for prediction accuracy and error.
7. Repeat steps 1-6 for the other three stocks.

ARIMA-LSTM Model

After creating both control groups and collecting their respective prediction accuracies and errors, I needed to connect both models somehow to start my ARIMA-LSTM experimental group. By taking inspiration from Professor Yu's ARIMA-RNN combination, I recognized it would be most effective to feed ARIMA's predicted values into the LSTM model. Yu believed that ARIMA's forecasts can serve as a smoothing constant to help reduce the model's tendency to overfit data and encourage more accurate predictions (Yu et al. 2018, 17). Therefore, I created the following formula to mirror Professor Yu's view of hybrid stock prediction: yt^ = at^ + lt^ or the final predicted stock (yt^) equals ARIMA predictions (at^) plus LSTM predictions (lt^). Though the values are not explicitly added, the formula indicates that ARIMA data is incorporated with the LSTM training data.

To evaluate the experimental ARIMA-LSTM, I repeated steps 1-5 of the ARIMA model setup for a given stock and stored the ARIMA predictions into a dataset. Then, while conducting the LSTM's model setup, I incorporated the ARIMA dataset and the training dataset as separate parameters into the LSTM during step 2. Finally, I finished the rest of the enhanced LSTM procedure to step 6 and recorded the average results of the prediction accuracies and errors for the ARIMA-LSTM. I repeated this experiment three more times for the other three stocks. After collecting all the data, I conducted a posttest analysis on prediction accuracies and errors between each of the three models.

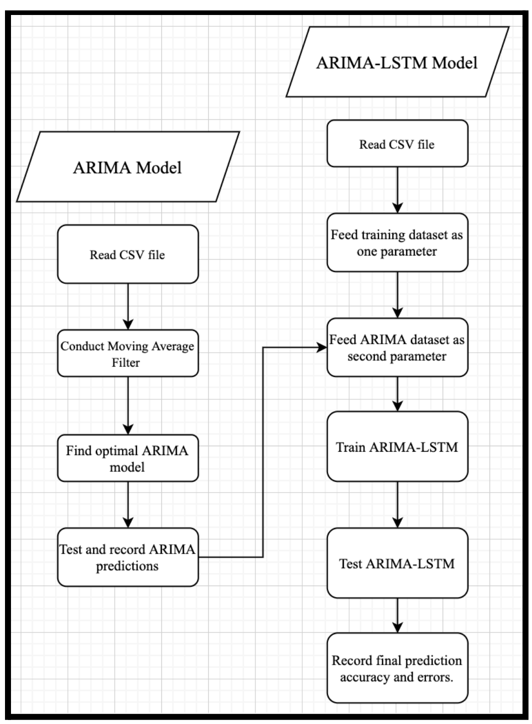

Figure 4: Flowchart Diagram of ARIMA-LSTM Procedure

Results

Prediction Accuracies

Google Colab's python simulation was used to identify last month's predicted price values for each stock on each prediction model, and the data points were graphed. The stocks' actual close price values are shown in green. The ARIMA-LSTM's predicted prices are shown in red, while the LSTM's and ARIMA's are presented in blue and yellow, respectively.

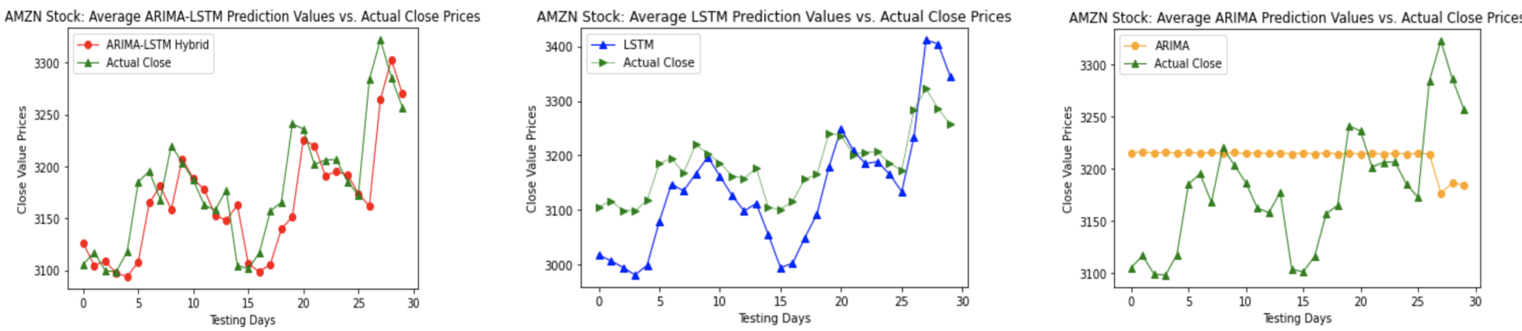

Figure 5: Prediction Results for Amazon's Stock

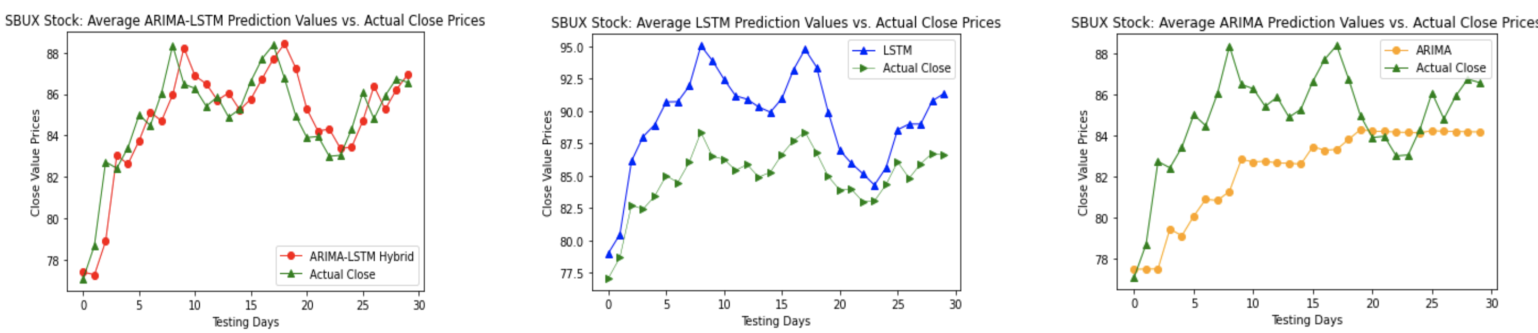

Figure 6: Prediction Results for Starbucks' Stock

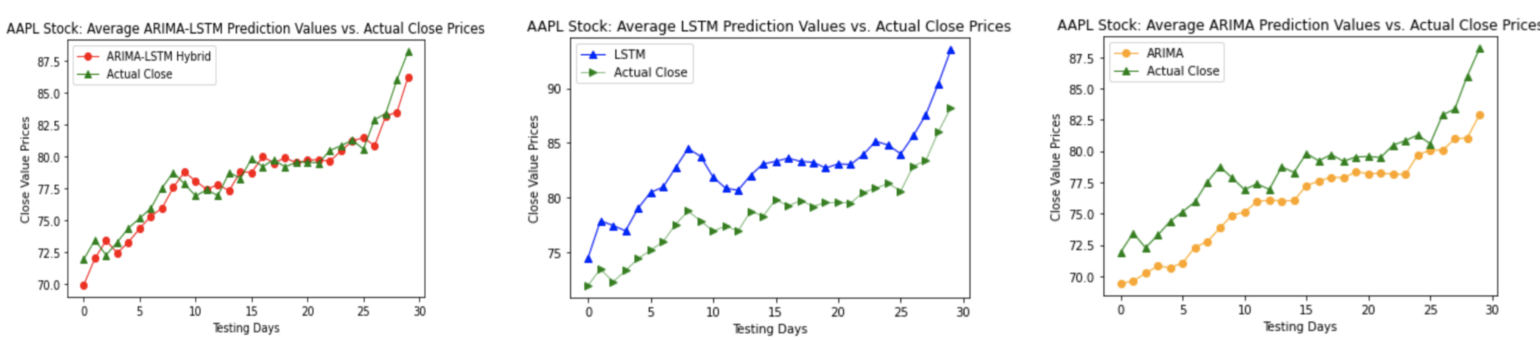

Figure 7: Prediction Results for Apple's Stock

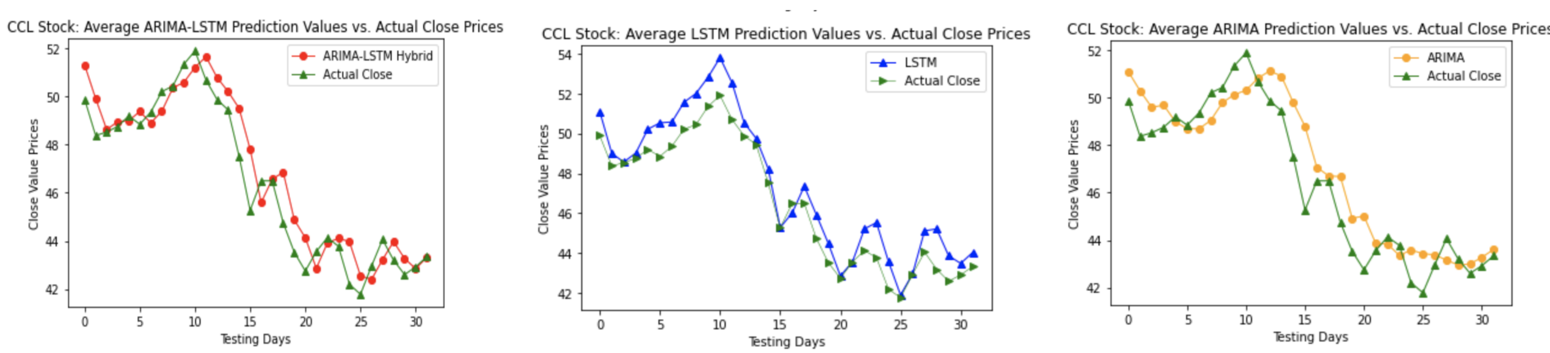

Figure 8: Prediction Results for Carnival's Stock

Upon conducting three trials for each stock with each model, predicted prices were recorded with the actual close prices to retrieve a prediction accuracy. Then, the overall prediction accuracy was calculated and plotted below to compare each model's results.

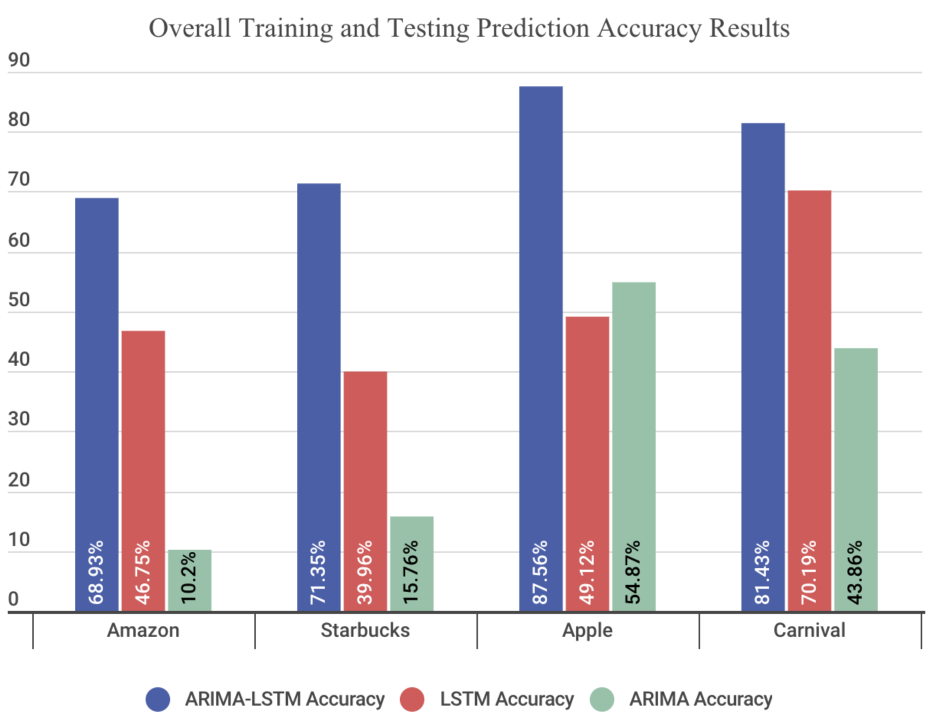

Figure 9: Overall Prediction Accuracy Bar Graph for Each Model on Each Stock

Next, a scatter plot was graphed for each stock to identify which model did better in prediction accuracy for each given day. This determines which models performed better in scenarios where a specific stock experienced volatility (outside of the red zone) or stability (inside the red zone). Additionally, the zone above the red line indicates a positive percent change (rising value), while the zone below indicates a negative percent change (falling value).

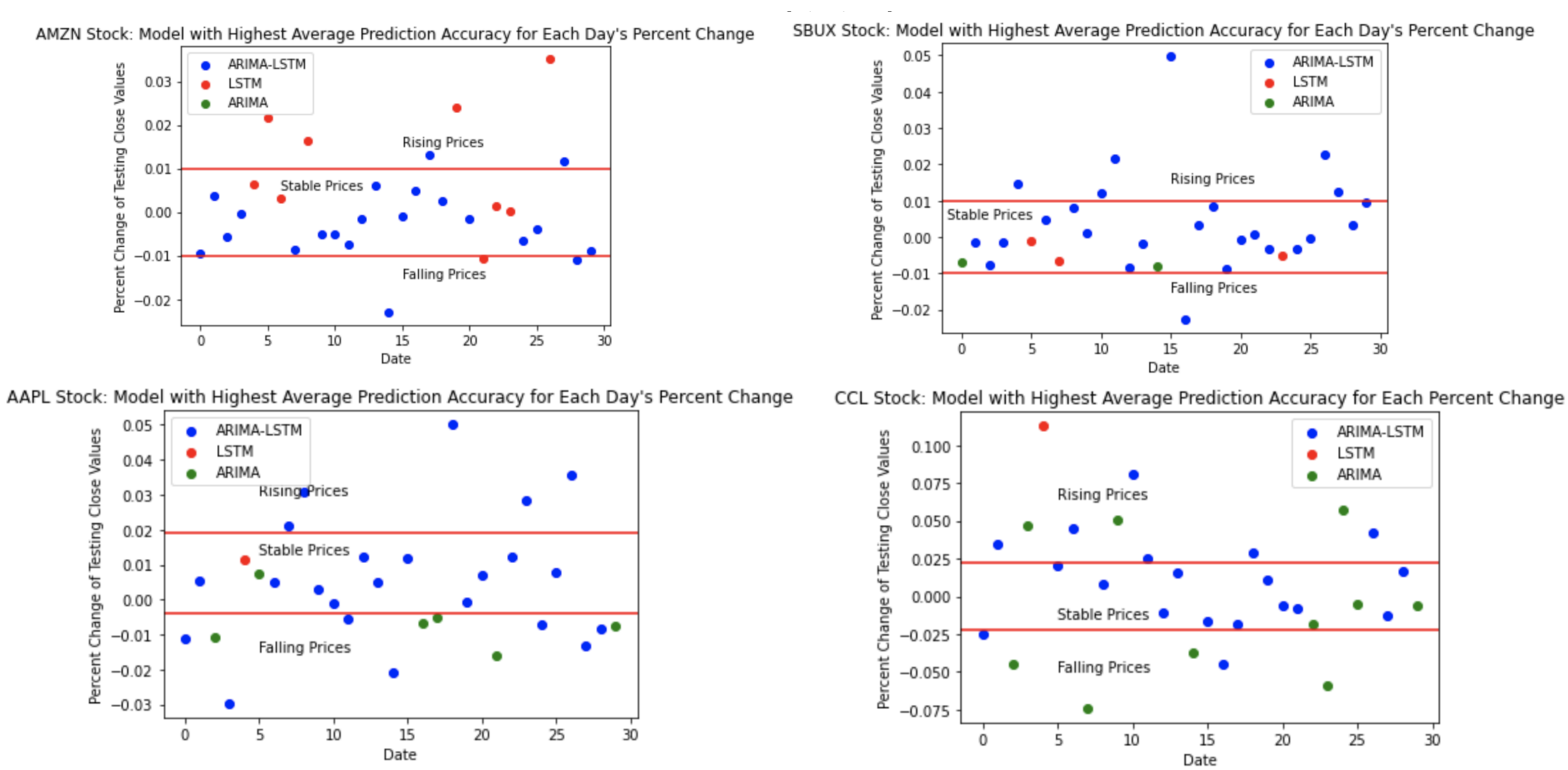

Figure 10: Best Model Prediction Accuracy for Each Day Scatterplots

As seen within the scatter plots, there are relative frequencies of 0.67 for Amazon, 0.83 for Starbucks, 0.77 for Apple, and 0.63 for Carnival that support higher prediction accuracies by the ARIMA-LSTM on a daily basis. The LSTM fared around 0.33, 0.1, 0.03, and 0.03 frequencies respectively, while ARIMA had 0.00, 0.06, 0.20, and 0.33 frequencies respectively.

Prediction Error Results

Finally, each model was checked for prediction errors among three trials, and results were averaged to identify which AI structure was the most efficient. The results from tables 1-3 showcase the MSE, RMSE, and MAPE values for ease of comparison.

Table 1: Average Prediction Error Results for Each Stock's ARIMA-LSTM Hybrid Model

	MSE	RMSE	MAPE
Amazon	8416.293	91.741	2.421
Starbucks	124.675	11.166	2.892
Apple	14.892	3.859	2.731
Carnival	1.816	1.348	5.388

Table 2: Average Prediction Error Results for Each Stock's LSTM Model

	MSE	RMSE	MAPE
Amazon	1432.678	37.851	21.57
Starbucks	59.711	7.727	24.89
Apple	5.717	2.391	15.39
Carnival	4.853	2.203	12.78

Table 3: Average Prediction Error Results for Each Stock's ARIMA Model

	MSE	RMSE	MAPE
Amazon	990.122	31.466	8.901
Starbucks	312.699	17.683	4.566
Apple	0.088	0.297	0.237
Carnival	0.198	0.445	1.003

Discussion

Overall Implications

Based on the results received from the comprehensive study, the study revealed that the ARIMA-LSTM hybrid model had substantially better prediction accuracies of stock prices than its predecessors: the ARIMA and LSTM models. However, the hybrid model simultaneously suffered larger prediction errors due to the complexity of its parameters and model as a whole. To support these implications, the discussion will analyze the data through relationships of prediction forecasts, data volatility, and overall training and testing efficiency.

Prediction Forecasts

The ARIMA-LSTM calculated a prediction accuracy greater than that of the LSTM and ARIMA for each volatile and stable stock. Under volatile stocks, AMZN's prediction accuracy with the LSTM averaged at 46.75%, while the ARIMA-LSTM predicted at an average accuracy of 68.93%. Conversely, the ARIMA only received a 10.2% accuracy rate. Since the Amazon market is facing an incredibly volatile period with shares “[surging] more than 3,600% from the company’s IPO in May 1997 to May 2019” (Stank 2021, 1), it was apparent that the ARIMA model’s accuracy would suffer, leading to less prediction accuracy for ARIMA-LSTM.

On the other hand, the stable Apple and Carnival markets shared a much higher result in prediction accuracy. The ARIMA-LSTM prediction for Apple peaked at its highest average of 87.56%, and the ARIMA fulfilled a consistent 54.87% average. Moreover, the LSTM model alone seemed to increase in prediction accuracy from volatile to stable markets, especially for Carnival at 70.19% accuracy. Based on these observations, it is clear that stable stocks managed more success with the ARIMA-LSTM model than the volatile markets did due to the higher ARIMA prediction accuracy. This relationship also reinforces that ARIMA has a positive effect on an ARIMA-LSTM’s overall stock prediction, and an ARIMA-LSTM can surpass the prediction power of an individual LSTM. Moreover, the ARIMA-LSTM combination seemed to perform better than pre-existing hybrid models. For example, Yu’s ARIMA-RNN calculated a maximum of 63.49% accuracy for the Shanghai Composite Index (Yu et al. 2018, 8), which is a significant drop in accuracy from the ARIMA-LSTM’s average results of over 70%. Based on the prediction accuracy discrepancy between volatile and stable stocks, it’s safe to conclude that ARIMA data strengthens ARIMA-LSTM’s prediction accuracy under stable stock conditions.

Data Volatility to Prediction Relationships

Though the ARIMA-LSTM successfully predicted overall close values better than previous models, it’s essential to identify how the ARIMA-LSTM performed under days of high or low volatility between stock prices. I calculated the percent change extreme of each stock for each day and plotted them on the scatterplots.

Fortunately, the ARIMA-LSTM outperformed the LSTM and ARIMA models as its overall frequency was the highest for each stock's scatterplot, regardless of whether the market was volatile or stable. More importantly, however, the volatile markets, Amazon and Starbucks, had LSTM frequencies of 0.33 and 0.1, respectively, which was significantly higher than the frequencies of the stable markets. This reinforces the concept that the ARIMA-LSTM relied heavily on the accuracy of the LSTM model's results to make its final predictions when forecasting volatile stocks. On the other hand, the stable markets, Apple and Carnival, tended to have an increased prediction accuracy for ARIMA models with relative frequencies of 0.20 and 0.33, respectively. Therefore, I can reasonably conclude that stable stocks will have better prediction values from ARIMA, while volatile stocks will have stronger LSTM prediction values. In both cases, however, the day-to-day prediction accuracy was performed the best by ARIMA-LSTM, which implies that the ARIMA and LSTM prediction accuracies individually contribute to the hybrid model depending on whether the stock is volatile or stable.

Overall Testing/Training Efficiency

Despite these positive results, the ARIMA-LSTM model had much larger training errors than the other two models. The ARIMA-LSTM produced a Mean Square Error of 8416.293 for Amazon, which is significantly larger than the MSEs of ARIMA (990.122) and LSTM (1432.68). However, there is some outlook for the ARIMA-LSTM on stable stocks as it calculated a low 1.816 MSE for Carnival and a typical 14.892 MSE for Apple.

Similarly, when compared to Professor Yu's ARIMA-RNN model, the MSE values were generally less at around 0.701 for SP500 Index and 771.582 for SSEC Stock (Yu 24). Despite the higher prediction accuracy, the ARIMA-LSTM performed slower and could be a lot more error-prone to overestimation than the ARIMA-RNN. Current financial engineers and statisticians have also noted that "a [RMSE] value closer to 0 indicates that the model has a smaller random error component" and that the prediction fit will take a lot less time to process (Garvan 2020, 1). From the above results, the RMSE values for the ARIMA-LSTM were the greatest (91.741 to 1.348) compared to the other models' results, further validating that the ARIMA-LSTM took an immense amount of time to fit the final data before making its predictions. Therefore, a conclusion can be drawn that the ARIMA-LSTM produced much higher prediction accuracies for both volatile and stable stocks at the cost of significantly higher prediction errors. More research should be conducted with these hybrid approaches to explore AI structures that are more efficient during training.

Limitations

Though the results partially validated my hypothesis, some limitations may have hindered the efficacy of my experiment. To start, the whole experiment centralized on long-term stock movement (training over twenty-year periods and testing within the last thirty days). According to financial engineer Bin Weng, some investors, often called daytraders, are looking toward a financial expert system that uses multiple data sources and can accurately predict stock prices over short time frames like twenty-four hours. Even though I attempted to look at percent change volatility for each day and its relation to the frequency that ARIMA-LSTM performed the best, the results still rely on long-term trend predictions. Moreover, the model is not accurate enough to Weng's expectations as he expresses that an 85% accuracy is required to thoroughly rely on short-term prediction (Weng et al. 2018, 13).

In terms of the methodology, a few limitations were present that I had not noticed until later in the research process. For instance, I collected over twenty years of previous stock price data for each experimental test, which may have led to an oversampling bias. Though I believed this large amount of data would enhance prediction accuracies between each model, I discovered from previous studies that excessive training data can lead to much higher prediction errors than intended, ultimately influencing models to overestimate stock predictions (Nosratabadi et al. 2020, 14). The potential oversampling bias may have caused the uncommonly high prediction errors present in my ARIMA-LSTM model. Future studies should investigate what optimal amount of sampling data would be most efficient.

Finally, I did not manipulate the exact variables, or hyperparameters, that may have influenced the LSTM model in creating its algorithms. Specifically, I left the epoch count, learning rate, and neurons as default measures. Consequently, these factors could serve as confounding variables in my study that may have impacted the results of my LSTM and ARIMA-LSTM models. Nevertheless, future research should focus on blocking these confounding variables more carefully when experimenting with the prediction accuracy of AI long-term stock forecasting, since this may lead to potentially better stock predictions.

Conclusion

The results of the experiment confirmed my hypothesis that incorporating ARIMA in LSTM models enhanced prediction accuracies. Conversely, the data also revealed that my hypothesis for prediction errors was disproved as these errors were the highest for the ARIMA-LSTM. Nevertheless, this study successfully explored the primary gap of intersecting ARIMA with LSTM to predict long-term stock trends. Whether the stock was volatile or stable, the experiment revealed how each separate model, ARIMA or LSTM, played a significant role in contributing to the hybrid model's prediction performance. Percent change volatility and low MSE testing results also showcased how ARIMA-LSTM could predict more efficiently on a day-to-day basis than previous models, like Professor Yu's ARIMA-RNN. Despite poor training performance on all four stocks, the substantial improvement in prediction accuracy from predecessor models is thrilling news for the world of financial engineering.

As shown by the positive results, hedge fund managers, stockbrokers, and small-time investors can utilize ARIMA-LSTM models to encourage smarter financial decisions in the stock market. However, future studies could investigate short-term stock price prediction with this hybrid model to combat my study's limitation of only looking into long-term forecasts. More moving average formulas, such as Triple Exponential Moving Average (TEMA) or Triangular Moving Average (TRIMA), could be tested to optimize the ARIMA component of the hybrid approach. Better yet, current studies from financial engineers like Professors Chao Chen and Xiao Lin of the University of South Carolina have already found AI classification tools like Bayesian-LSTM models that can predict short-term and long-term stock movement much faster. Integrating the ARIMA-LSTM model with more advanced AI structures like Bayesian-LSTM can substantially improve our knowledge of AI stock prediction. Hopefully, by building off the ARIMA-LSTM base model, we can come one step closer to creating a revolutionary model capable of accurately forecasting the stock market.

References

Adebiyi, A. A., Ayo, C., & Adewumi, A. (2014, March). Stock price prediction using the ARIMA model. Retrieved March 3, 2021, from https://www.researchgate.net/publication/261179224_Stock_price_prediction_using_the_ARIMA_model

Chen, C., Lin, X., Huang, Y., & Terejanu, G. (2019). Approximate Bayesian Neural Network Trained with Ensemble Kalman Filter. 2019 International Joint Conference on Neural Networks (IJCNN), 1-8. doi:10.1109/IJCNN.2019.8851742

Divine, J. (2020, October 5). 7 Stocks That Have Dropped the Most in 2020. Retrieved January 27, 2021, from https://money.usnews.com/investing/stock-market-news/slideshows/

Dong, Y., Hou, J., Zhang, N., & Zhang, M. (2020, October 28). Research on How Human Intelligence, Consciousness, and Cognitive Computing Affect the Development of Artificial Intelligence. Retrieved January 13, 2021, from https://www.hindawi.com/journals/complexity/2020/1680845/

Elbahloul, K. (2019). Stock Market Prediction Using Various Statistical Methods. Stock Market Prediction Using Various Statistical Methods, I, 1-4. doi:10.13140/RG.2.2.13235.17446

Farmer, R. E. (2011). The Stock Market Crash of 2008 Caused the Great Recession: Theory and Evidence. *NBER WORKING PAPER SERIES,* 1-37. Retrieved January 20, 2021, from https://www.nber.org/system/files/working_papers/w17479/w17479.pdf.

Jain, N. (2019, August 2). Sequence Learning: Predicting Stock Prices using Multivariate Analysis. Retrieved February 15, 2021, from https://github.com/ninja3697/Stocks-Price-Prediction-using-Multivariate-Analysis#i-abstract

Jansson, P., & Larsson, H. (2020). ARIMA Modeling Forecasting Indices on the Stockholm Stock Exchange. Handelshogskolan Karlstad Business School, 1-37. Retrieved March 3, 2021, from https://www.diva-portal.org/smash/get/diva2:1413099/FULLTEXT01.pdf.

Garvan, A. (2020). Goodness-of-Fit Statistics - University of New South Wales. Retrieved March 4, 2021, from https://web.maths.unsw.edu.au/~adelle/Garvan/Assays/GoodnessOfFit.html

Goswami, S., Shit, L., & Mondal, P. (2014, April). Study of Effectiveness of Time Series Modeling (Arima) in ... Retrieved February 20, 2021, from https://www.researchgate.net/publication/276197260_Study_of_Effectiveness_of_Time_Series_Modeling_Arima_in_Forecasting_Stock_Prices

Makhlouf, W., & Malhotra, R. (2020, October 10). STOCK MARKET PRICE PREDICTION USING LONG-SHORT TERM MEMORY (LSTM). Retrieved March 20, 2021, from https://www.academia.edu/45981223/STOCK_MARKET_PRICE_PREDICTION_USING_LONG_SHORT_TERM_MEMORY_LSTM_

Moghar, A., & Hamiche, M. (2020, April 14). Stock Market Prediction Using LSTM Recurrent Neural Network. Procedia Computer Science. Retrieved April 11, 2021, from https://www.sciencedirect.com/science/article/pii/S1877050920304865#.

Mohankumari, C., Vishukumar, M. & Chillale, N.R.. (2019). Analysis of daily stock trend prediction using ARIMA model. *International Journal of Mechanical Engineering and Technology,* 1-2. Retrieved February 20, 2021, from https://iaeme.com/MasterAdmin/Journal_uploads/IJMET/VOLUME_10_ISSUE_1/IJMET_10_01_176.pdf

Nelson, D., Pereira, A., & deOliveira, R. (2017). Stock market's price movement prediction with LSTM neural networks. 1419-1426. doi: 10.1109/IJCNN.2017.7966019.

Nosratabadi, S., Mosavi, A., Duan, P., Ghamisi, P., Filip, F., Band, S. S., . . . Gandomi, A. H. (2020, October 16). Data Science in Economics: Comprehensive Review of Advanced Machine Learning and Deep Learning Methods. Retrieved March 3, 2021, from https://www.mdpi.com/2227-7390/8/10/1799/htm

Qiu, J., Wang, B., & Zhou, C. (2020). Forecasting stock prices with long-short term memory neural network based on attention mechanism. *PLOS ONE*. Retrieved January 17, 2021, from https://journals.plos.org/plosone/article?id=10.1371%2Fjournal.pone.0227222#:~:text=Long%20short%2Dterm%20memory%20(LSTM,helps%20predict%20financial%20time%20series.

Rossi, A. G. (2018). Predicting Stock Market Returns with Machine Learning. University of Maryland Papers, 1-44. Retrieved April 12, 2021, from https://mendoza.nd.edu/wp-content/uploads/2019/07/2018-Alberto-Rossi-Fall-Seminar-Paper-1-Stock-Market-Returns.pdf.

Saad, L. (2021, March 23). What Percentage of Americans Owns Stock? Retrieved April 24, 2021, from https://news.gallup.com/poll/266807/percentage-americans-owns-stock.aspx

Shakya, S. (2020, January). Analysis of look back period for stock price prediction with RNN variants: A case study on banking sector of NEPSE. Retrieved April 20, 2021, from https://www.researchgate.net/publication/340705916_Analysis_of_look_back_period_for_stock_price_prediction_with_RNN_variants_A_case_study_on_banking_sector_of_NEPSE.

Sharma, C., & Banerjee, K. (2015, April 22). A Study of Correlations in the Stock Market. Retrieved January 13, 2021, from https://arxiv.org/abs/1504.05844

Stank, K. (2021, March 18). Anthony Scaramucci says it's not too late to buy bitcoin - just look at Amazon's run since 2009. Retrieved April 12, 2021, from https://www.cnbc.com/2021/03/18/anthony-scaramucci-compares-bitcoins-trajectory-to-amazons-stock-.html

Tapiero, C. S. (2013). The Future of Financial Engineering. SSRN Electronic Journal, 1-25. Retrieved March 3, 2021, from https://www.researchgate.net/publication/256061541_The_Future_of_Financial_Engineering.''

Weng, B., Lu, L., Wang, X., Megahed, F. M., & Martinez, W. (2018, June 15). Predicting short-term stock prices using ensemble methods and online data sources. Retrieved January 18, 2021, from https://www.sciencedirect.com/science/article/abs/pii/S0957417418303622