Predicting the Future: Using SARIMAx-based Trend Analysis to Forecast Short Term Real-Time High Volume Stock and Index Prices with Loss Mitigation

Word Count: 5000

Introduction

The stock market has proven to be more than just one of the many minor innovations arising several centuries ago. According to gallup news, in recent times, approximately 56% of American adults, or 144.6 million Americans, have some amount of money invested in the stock market (Jones, 2021). However, what may prove to be even more impressive than the widespread interest in the stock market is its enduring uprising value. The market capitalization of all publicly traded stocks across to globe increased from $2.5 trillion in 1980 to $93.7 trillion in 2020, for a an over 3000% increase, according to the World Bank. Following the rising interests and holdings in the stock market, new strategies for generating profit from stocks are created to succeed in an adapting market. Global stock exchanges and trader investment strategies have evolved heavily with the rise of new forms of technology. As computer automation sweeps different industries through numerous new applications, the financial-technology (fintech) industry has looked to algorithms to automatically assist or execute stock trades. The process of using automated computer programs to execute stock orders is known as algorithmic trading. Algorithmic trading has been proven time and again to demonstrate high efficacy in securing sizable profit margins for trade programmers.

Literature Review

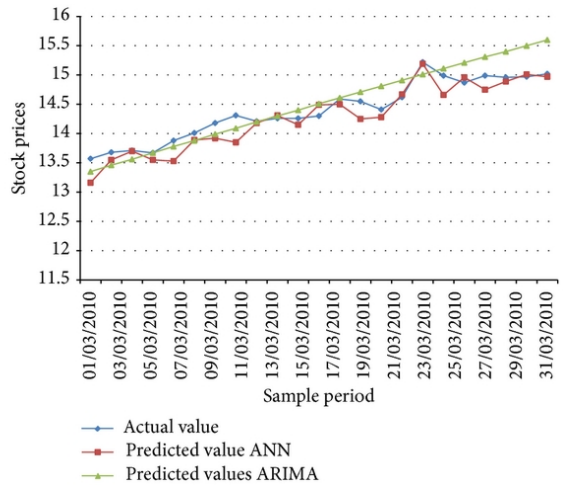
With the age of artificial intelligence (AI) sweeping the technology and fintech market, the relatively quick arrival of both democratized algorithmic trading resources and artificial intelligence softwares have occurred nearly at the same time. According to a historical analysis of fintech trade strategies by Duke University author Michael J. McGowan, the first-known algorithmic trading software was developed in 1976 and was known as the New York Stock Exchange's "designated order turnaround" system (DOT) (McGowan, 2010). Similarly, the concept of artificial intelligence as a whole was first widespread in the early 70's, dating all the way back to its introduction in 1957 (Anyoha, 2017). For this reason, relatively few studies have tested the efficacy of traditional methods of stock price analysis ("traditional" referring to non-AI algorithms), as the hedge fund market quickly shifted its interest towards AI in the form of Artificial Neural Networks (ANN) given their similarly timed upbringings, leaving sparse experimentation with traditional non-AI approaches.

In the fintech industry, it is widely known that many ANNs are capable of relatively high success rates in predicting the market. Initially acknowledging this widespread success of ANNs, a 2020 study led by professor Santur Yunus (PhD Computer Science) attempted to forecast the prices of Istanbul Bist-30 stock index using both a simple-moving average (SMA) algorithm and multivariable ANN, which utilizes past data to uncover hidden nonlinear trends in data (Yunus et al., 2020). The moving average-based price predictor had an accuracy of just 50.4%, while the ANN model had an accuracy of about 55.1% (Yunus et al., 2020). The SMA model was did not have statically significant accuracy, while the ANN model did. However, one should be hesitant to deny the capabilities of traditional linear analysis methods.

Complex moving averages (MA), such as the Autoregressive Integrated Moving Average (ARIMA), are forecasting techniques that use a summative-regression of past data to determine the future trajectory of data. These forecasts are strictly linear and seldom have the ability to forecast hidden nonlinear trends, an area which ANNs have the upper hand (Demirel et al., 2021). Using trends in complex mathematical averages and regression models, MAs are able to generate predictions from a sample without employing modern AI techniques. MAs has been used for over a century in applications far beyond the financial industry. In a 1909 research publication, British mathematician and Guy Statistician Award winner G. U. Yule described the first known model for “moving-averages” for use in statistical research (Journal of the Royal Statistical Society, 72, 721-730). Since then, ARIMA has evolved to fit a wide variety of applications. Recently, the Center for Convergent Research of Emerging Virus Infection in the Korean Institute of Science and Technology implemented an ARIMA of COVID-19 cases reported in a period of 12 days to devise a predictive risk score of viral transmission in South Korea (Choi & Ahn, 2020). Researchers have used many types of MAs to predict significant and minor phenomena alike as a consequence of its high versatility and accuracy.

Although there has been much concentration on ANN models, some research has been done to determine the efficacy of MAs in predicting simulated past and future data. A 2017 study led by financial research Professor Muhammad Ahmad et al. was conducted to determine the accuracy of a simple moving average backed by 4,217 days of past data to determine future stock prices (Ahmad, Muhammad Ishfaq et al., 2017). The algorithm decided to invest in a stock the following day if a market price exceeds the regression moving average. Ultimately, the value of return ranged from 2.86% to as high as 10.49%, with an accuracy of about 60% (Ahmad, Muhammad Ishfaq et al., 2017). This is a significantly large range, however, it demonstrates a possibility that MA trade models may have a higher margin of accuracy compared to many ANN models (compared to the 55.1% accuracy of the ANN model used in the Yunus et al. study).

Finally, a 2014 study by Adebiyi et al. has already debunked the differences in marginal accuracies between ANN and ARIMA models of forecasting stock prices (Adebiyi et al, 2014). The research found that ANNs had only a very slight advantage in their long-term simulation of both algorithms using past data of the New York Stock Exhcange (Adebiyi et al., 2014). The ANN algorithm performed much better at predicting short term movements, however the ARIMA model had similar accuracy in reporting the longitudinal price values of each stock



(Adebiyi et al, 2014). The ARIMA and ANN's performance in this study is graphed on the right (Adebiyi et al, 2014). As algorithmic trading resources become more democratized and simplified, prospective trade programmers would benefit from ascertaining the realistic (pragmatic) profit margin they could expect from either algorithm. This margin of profit differs from the dependent variable studied in the Adebiyi et al. experiment and most of the aforementioned stock trading studies, and is more applicable than the aforementioned variables (marginal accuracy) for forecasting the pragmatic feasibility of applying a trade algorithm.

Pragmatism refers to several different key aspects that attempt to make the findings of this research more applicable, and is also the major gap that is not addressed in the previous studies. First, the vast majority of professional traders implement a loss-mitigation strategy to complement their manually-executed trades, while none of the aforementioned studies had (Keupper, 2021). This would likely increase profit margins by decreasing the impact of poor stock purchases. Second, rather than solely measuring general accuracy, this study will also measure profit margins based on a virtual implementation of the algorithm in real-time trading environments. Measuring real profit margin, as opposed to marginal accuracy as most of the previous studies had, will be a more applicable indicator of the finances one could expect to gain (or lose) using this algorithm in the actual market. Third, rather than testing the algorithm in a simulation of past data, this study will implement the ARIMA algorithm in a real-time market simulation. While some studies had measured real profit margins or performed a real-time market implementation of these trading strategies, none had incorporated both. Lastly, all of the aforementioned studies used default (static) ARIMA (p, d, q) coefficients. More information on these coefficients and its impact is described in the "methodology" section. Lastly, while ARIMA is generally better at forecasting more complex trends than the simple moving average, ARIMA only forecasts a single line with a static slope. This means that ARIMA is more useful for analyzing the eventual path of longer trends, as opposed to the short peaks and dips of the future data. Seasonal-ARIMA (SARIMAx) would prove to be better suited, as it repeats the construction of ARIMA models over a given (seasonal) time interval.

In determining the pragmatic efficacy of linear analysis models of algorithmic trading, this research will also be a potential resource in determining what type of algorithmic trading models will better serve trade programmers. A 2019 Oxford study analyzed the application advantages to using either MA-based algorithms in predicting stock prices (Noureddine et al., 2020). According to the study, the reason moving-average based models may be more ideal for traders is that it generally involves much less time, resources, and specialized knowledge to produce, in contrast to ANN's, which are relatively complex equations (Noureddine et al., 2020).

Research Question

Combining these gaps with the strengths of the aforementioned studies, this experiment will attempt to use trend analysis to answer the question, "What is the normal profit/loss margin and accuracy of an hourly Seasonal-ARIMA price forecasting model in a real-time trend analysis of 20 high volume stocks and 4 indexes with loss mitigation?"

Methodology

ARIMA Parameters

For the purpose of this experiment, trend analysis is conducted through numerical forecasting. An SARIMAx(p, d, q, M) model was used for such forecasting. “p” represents the quantity of lag observations in the given algorithm model; “d” represents the number of times that differences are taken between raw observations; “q” represents the size of the moving average interval; and M refers to the frequency of the seasonal (repeating) trend (Hayes, 2021). The (p, d, q, M) coefficients are also known as the order. The equation for the SARIMAx algorithm is shown below:

$$\underbrace{(1-\phi_1 B)}_{\left(\substack{\text{Non-seasonal}\\ \text{AR(1)}}\right)}\ \underbrace{(1-\Phi_1 B^4)}_{\left(\substack{\text{Seasonal}\\ \text{AR(1)}}\right)}\ \underbrace{(1-B)}_{\left(\substack{\text{Non-seasonal}\\ \text{difference}}\right)}\ \underbrace{(1-B^4)}_{\left(\substack{\text{Seasonal}\\ \text{difference}}\right)}y_t = \underbrace{(1+\theta_1 B)}_{\left(\substack{\text{Non-seasonal}\\ \text{MA(1)}}\right)}\ \underbrace{(1+\Theta_1 B^4)}_{\left(\substack{\text{Seasonal}\\ \text{MA(1)}}\right)}e_t.$$

SARIMAx Equation (Hyndman et al.)

Unlike Santur, Adebiyi, and Ahmad’s study, the SARIMAx order is not predetermined in this experiment. This is due to the fact that one sample of data may be predicted more accurately with a different order arrangement, therefore, applying a standard set of coefficients may not be optimal for obtaining the highest accuracy (Nau, 2015). To find the optimal (p, d, q, M) values, an open source Python program is run prior to initializing the SARIMAx formula. The program automatically creates short prediction intervals based on the provided stock data with many variations of (p, d, q, M) values. Then, it compares the forecasted model with the actual data to see which combination of coefficients are the most accurate, and thus would be a better probable fit for predicting future data.

Programming Resources

After preliminary experimentation with several approaches to collecting stock data and constructing SARIMAx models, Python proved to be the best program for the task. Other viable contenders included JAVA, Excel, and online calculators. JAVA and Excel have little open-source support for web scraping–extracting stock prices from online websites in this case. Online calculators have no support for any sort of data collection, they would require another program to complete this which is inefficient.

There are many open source packages and programs available for Python that simplify numerous aspects of this project's procedure. For collecting real-time stock data, an open-source Python package called Beautiful Soup allows the extraction of data from a specific field of an online source into a readable String (text) (Richadson, 2007). This, paired with another open source Python package called CSVwriter, allows for an easy translation of real-time stock prices displayed online into a spreadsheet that can be used as the input data for the ARIMA model. CSV refers to a spreadsheet (comma separated value) arranged in rows and columns. The Statsmodels package is an open source Python package that contains several complete ARIMA constructors (Seabold et al., 2010). All it requires is the data, the aforementioned (p, d, q, M) values, the data interval, and a forecasting interval to generate and graph forecast models. None of the alternative programming resources have reliable open-source programs for all or most of these objective (or comparable ones). Given that these packages have all the required methods for getting online data, recording it onto a readable file, and creating a mostly pre-configured SARIMAx model for such data, the aforementioned packages will be ideal for use in this experiment due to their relative simplicity.

Subject Selection

20 stocks across 6 distinct industries along with 4 major indexes were selected for trend analysis. The industries are: technology, health/pharmaceutical, finance, travel, automotive, and retail. According to NASDAQ.com, these categories are considered the most popular for traders, meaning that an analysis of stocks from these categories would better appeal to most traders (Samuel, NASDAQ).

A complete list of the selected stocks and indexes, along with the algorithm's final forecasting data, are provided below in appendix A and B. These stocks were selected based on market capitalization and the 1 month average derivative of volume. According to Investopedia's Chris Murphy, market capitalization is often reflective of high long term trader interest (Murphy, 2021). The average derivative of the trade volume indicates recent interests in a stock, giving more applicability for a modern usage of the algorithm. 15 stocks per category leading in market capitalization and average change in trade volume were ranked based on these criteria; their ranks in the two respective categories were averaged out. The top 20 stocks with the lowest average scores were selected. Indexes were selected as they are also a vital aspect of forecasting the general stock market atmosphere. The Santur study solely relied on forecasting indexes due to their wide ranging applicability to the general market. However, not all stocks follow the trends of indexes, and in terms of pragmatic analysis, traders tend to value the trends of individual stocks just as much as the major indexes (Mitchell, 2022). For this reason, both stocks and indexes are used for wide-ranging indications of forecasting ability.

"24" securities were analyzed based on the specifics of the hardware being used to compute the ARIMA model. This is due to the fact that Python, the language being used to construct the ARIMA model, is generally only capable of running one script on one CPU core.

Despite this, computational efficiency can be maintained by running separate SARIMAx model programs, one for each core. The particular server being used to calculate the SARIMAx model has 12 cores from two 6-core Intel Xeon platform processors, and is therefore capable of effectively running 12 Python scripts at one time. SARIMAx models are calculation intensive–each one may require billions of calculations. By making the number of data sheets equally divisible by the number of CPU cores available on the computer used, data is forecasted with much higher efficiency in terms of calculation time. Since 24 securities were selected, 24 SARIMAx models would be calculated in the same amount of time as 13-23 models, that is, two trials of calculations. Two trials allows for the coverage of the most popular securities while maintaining computational efficiency. This would be important for traders looking to implement an SARIMAx algorithm in a real time trading environment, as stock prices will change in the time it takes for SARIMAx calculations to be made, delaying the intended entry point for new trades.

Extenuating Circumstances and Response

In determining the efficacy of an ARIMA algorithm in a real-time market with pragmatic applicability, it becomes ideal to have normal market conditions. Adverse market conditions create a situation that is unlikely to resonate with traditional trading environments, and therefore lacks this necessary pragmatic applicability. Recent socio-political tensions in Eastern Europe have prompted unprecedented market volatility across all major exchanges. In response, this study has been altered from its original design to compensate for these abnormalities.

Making long term predictions would prove to be difficult since the current state of the market is heavily influenced by (unpredictable) political activities. Short-term predictions, not exceeding a prediction of 1-hour future price, should be less affected by these tensions, and were therefore replaced in this experimental design. Some changes were made to this procedure and will be reflected by italicizing the updated design.

Time Intervals

Certain times and intervals were rationalized based on various criteria and circumstances. Initial stock data will be collected for *3 days* (starting on Monday) and forecasting will begin on *day 4 (Thiursday)*. This allows for ample data to be collected prior to constructing the SARIMAx for greater accuracy. A full (7 day) week was not used as trend analysis could not be performed during weekends since stock prices aren't updated; dividing the week for both data collection and forecasting allows for one continuous trend to be analyzed. From days *4-5,* stock prices will still be recorded to improve the accuracy of the ARIMA model, even while forecasting is conducted.

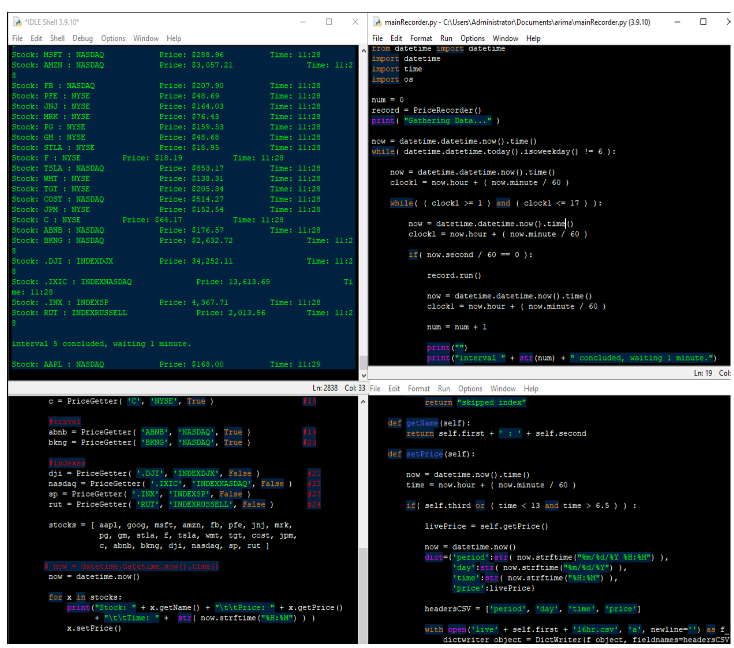
Stock price data collection started at 4 AM (EST) due to the fact that such time is when the pre-market opens. Stock price data collection stopped at 8PM EST each day as this is when after-hours closes. While the greatest price shifts in a stock often occur during regular hours, the changes that occur outside of regular hours are enough to indicate a change in the trend of a stock. Additionally, the Statsmodels SARIMAx package will reject predictions from datasets that have significant gaps between adjacent data points; neglecting pre-market and after-hours data would cause a large enough gap to where the forecasting program will yield an exception. Indexes, however, stop updating after regular hours, so the price data for indexes are only collected during regular hours (9:30 AM - 4 PM EST).

Prices were collected and written to a spreadsheet every 1 minute within the time period. This is ideal for making short-term predictions as it will include small and gradual price changes alike. 1 minute is also enough time for the algorithm to scrape and record online price data for all 24 securities in time before the next minute, preventing overlap.

Under ideal conditions, this would produce around 96,000 data points for 20 stocks after 5 days. For all indexes, this would produce 7,800 data points after 5 days. In total, there will be over 100,000 data points analyzed to construct the SARIMAx models; plenty of data for forecasting short-term price intervals.

Procedure

1. Begin stock price data collection at 4:00 AM EST (start of premarket trading), and index price data collection at 9:30 AM EST (start of regular market hours)
2. Begin recording prices for all selected stocks/indexes by doing to following for each:
    a. Use Beautiful Soup to scrape the updated price of the stock/index from Google Finance
    b. Write the stock/index name, price, and time in a new row in a large CSV file
3. Repeat step 2 every minute until 4:00 PM EST for indexes and 8:00 PM EST (end of after hours trading)
4. Repeat steps 1-3 for 2 more days (actual code/output shown to the right)
5. Starting on day 4, run a specialized Python for-each loop to scan every cell in each price CSV file and determine the (%) difference between each adjacent data point. If the difference for any two cells exceeds 1%, data was



likely missing (usually due to downtime issues with Google's servers) and will be rectified by populating new cells between the two that gradually transition the data

6. Starting on day 4 at 6:00 AM EST, run the AutoARIMA method from the statsmodels package to automatically determine the optimal (p, d, q, M) order for each stock forecasting model
7. Starting at 6:30 AM, run the Statsmodels SARIMAx forecasting method for each stock/index with the optimal order to predict the next *hour's* price
8. Record the 1-hour predicted value on a CSV (along with the respective identifying attributes) using CSVwriter
9. Update a stock list array with stocks that have a positive *1-hour* forecast (representing "buy orders," not performed for indexes as they cannot be bought/sold)
10. Starting on *day 4,* prices will be recorded the same way as before (from 4AM - 8PM EST), but these prices will also be analyzed for loss mitigation (LM) (starting every hour after regular hours). If the forecasted price of any of the 'purchased' stocks is predicted to decrease by the SARIMAx model, the CSV writer will add a new entry in the log for "sell" + (stock name) to represent a cut loss "stop limit" order
11. At the end of each market *hour*, record the final price of each stock and index for comparison to the predicted price using CSV writer, calculate marginal profit/loss for the "holdings," and margin of the previous forecast's trend accuracy
12. Repeat steps 4-8 every hour to forecast new 1-hour prices (and represent buy-orders) with the updated stock/index info from 6:30 AM EST until 12:30 PM EST
13. Repeat steps 4-12 for day 5
14. At the end of day 5, sell any unsold holdings to calculate the overall profit

changed from original design due to extenuating circumstances

Tested Hypotheses

This study is primarily concerned with testing the efficacy of the configured SARIMAx model based on profitability and accuracy. The following hypotheses were tested with these interests in mind:

1. H_0: The Seasonal-ARIMA's 1-hour overall trend forecasts are not accurate.

   H_a: The Seasonal-ARIMA's 1-hour overall trend forecasts are accurate.

2. H_0: The Seasonal-ARIMA's simulated trades are not profitable.

   H_a: The Seasonal-ARIMA's simulated trades are profitable.

3. H_0: The Seasonal-ARIMA's 1-hour uptrend forecasts are not accurate.

   H_a: The Seasonal-ARIMA's 1-hour uptrend forecasts are accurate.

Hypothesis 1 is similar to the tests performed in the Santur, Adebiyi, and Ahmad study. Hypothesis 2 is based on the aforementioned gap in research, measuring the profitability of the algorithm. Hypothesis 3 was proposed following the conclusion of the data collection to analyze the conditions where the algorithm was the most accurate. 1-Proportion Z-Tests are performed at the 0.05 confidence level to test each hypothesis.

Results

Stocks

In terms of pure accuracy, the algorithm shows mixed results in predicting short term price changes across the 20 stocks. For the day 4's 140 predicted stock intervals, the algorithm had successfully predicted the next hour price trend of the stock 67 times (47.86%), and was incorrect 73 times (52.14%). For day 5, it predicted the stock trends correctly 79 times (56.43%) and incorrectly 61 times (43.57%), for a combined total of 146 correct (52.14%) and 134 incorrect (47.86%) predictions. It should be noted that it is expected for a randomized guess for a stock trend to have a 50% accuracy in successfully predicting the trend (uptrend or downtrend), however this varies based on general market conditions. At the 5% significance level, these results are not statistically significant with a p-value of 0.23. We therefore fail to reject the first null hypothesis, which states that the algorithm is incapable of accurately predicting 1-hour stock trends. However, as predicted earlier, these results do not tell the full story in terms of SARIMAx's potential.

```
bought 66 indexes
sold 66 stocks

number of correct buy predictions: 40
number of wrong buy predictions: 26
number of correct sell predictions: 22
number of wrong sell predictions: 44
number of total correct predictions: 145
number of total wrong predictions: 135
number of total correct uptrend predictions: 113
number of total wrong uptrend predictions: 53
number of total correct downtrend predictions: 32
number of total wrong downtrend predictions: 82


required balance for $100 share purchases: $1801.7240868953472
return $ : $40.771881000266966
return % : 2.262937951642161%
required balance per dollar invested: $18.017240868953472

******************* END OF DAY **********************
>>>
```

Output of Stock Price Prediction Results

The algorithm demonstrated a combined +2.26% overall simulated marginal profit in the 14-hours (over day 4 and 5) that the algorithm was tested. The algorithm made 66 trades over the two days, 40 of which were profitable, for a p-value of 0.042. This suggests truth to the 2nd alternative hypothesis (algorithm is profitable). A question may be raised as to how this margin of return is possible given that the SARIMAx model tested lacks statistically-significant price trend forecasting ability. There are several different reasons for this. For one, the forecasting model had a much higher accuracy in predicting uptrends compared to downtrends. The algorithm predicted uptrends accurately 114 out of the 166 times the stock price went up

(68.26% accuracy), giving a statistically-significant p-value of ≈0.00. This suggests truth to our third alternative hypothesis (algorithm accurately predicts uptrends). Conversely, the algorithm only accurately predicted downtrends a mere 28.32% of the time. For these reason, suggested “buy” orders ended up being profitable 60.61% of the time, while suggested loss mitigation (LM) sales were ‘favorable’ sales (reduced loss) only 33.33% of the time. 66 buy and 66 sell executions were made within the two 7-hour (regular market hours) periods, for an average of slightly less than 11 trades per hour.

While the algorithm may not be effective in predicting downtrends, this lack of accuracy does not directly affect marginal profits. This is due to the fact that when the algorithm incorrectly considers a stock to be on a downtrend, it simply misses the opportunity to make additional profit, which is not the same as losing capital. However, since loss mitigation sold any purchased stock if a downtrend was predicted, it is likely that loss mitigation did not often work as intended.

Indexes

```
number of total correct predictions: 39
number of total wrong predictions: 17
number of total correct uptrend predictions: 37
number of total wrong uptrend predictions: 10
number of total correct downtrend predictions: 2
number of total wrong downtrend predictions: 7
```

Output of Index Forecasting Results

For the four indexes, the price trend was accurately predicted 69.64% (39 out of 56) of the time over the two days. The algorithm accurately predicted uptrends 78.72% of the time (37 out of 47), while only being able to accurately predict downtrends 22.22% of the time (2 out of 9). These figures suggest that the general market was experiencing an overall rise in prices, which may partially affect the realistic profit margin one could expect to attain with SARIMAx compared to what was observed in this study. Marginal profits were not evaluated since indexes cannot be purchased/sold. If the same hypotheses were tested for indexes, there would be evidence to support all of the alternative hypotheses.

Discussion

Implications

The results of the SARIMAx's trials suggest that it is generally profitable when used in a real-time market for 1-hour forecasts and predicts 1-hour uptrends accurately, but is not generally accurate due to poor downtrend accuracy. Even with this poor downtrend accuracy, the algorithm had a higher overall accuracy than the Santur study's MA model, but lower than the Ahmad study. This is likely due to the fact that each stock had its own SARIMAx order (p, d, q, M) modeled and fit prior to forecasting using recent price data, making it more accurate than Santur's basic configuration. All of the other referenced studies had used default coefficients for all subjects such as (0, 0, 0) or (0, 0, 1). Certain coefficients appeared to work better than others depending on the stock-specific trends and general market environment. It can be speculated that the uptrend accuracy of the algorithm is higher than other studies, and that the overall accuracy was burdened solely by a poor downtrend accuracy.

Where this design appeared to be less efficient than others was in its general accuracy. The Ahmad study had an ARIMA accuracy as high as 60%, while this design only had an accuracy of around 52% in predicting stock prices. The reason for this can only be speculated. It could be that SARIMAx is perhaps more accurate at predicting long-term price trends like the other studies had used it for; or that it is particularly less effective at predicting downtrends in 1-hour intervals. SARIMAx may also not be as accurate as traditional ARIMA or moving average in forecasting short term intervals, given that there may exist less seasonal trends in the short-term market environment than in the long-term aspect.

The accuracy of forecasting index prices was likely statistically significant due to the fact that the stock market generally experienced price increases, which means that the indexes likely had fewer downtrends to skew the accuracy. In a more mixed market environment, it would be expected for the accuracy of the index price predictor to be similar to that of the stock predictor.

While the margin of return and accuracy may not be up to the standards of optimized ARIMA models or machine learning frameworks, SARIMAx's statistically significant uptrend accuracy and profitability may prove it worthy of consideration in fintech investment strategies.

Limitations

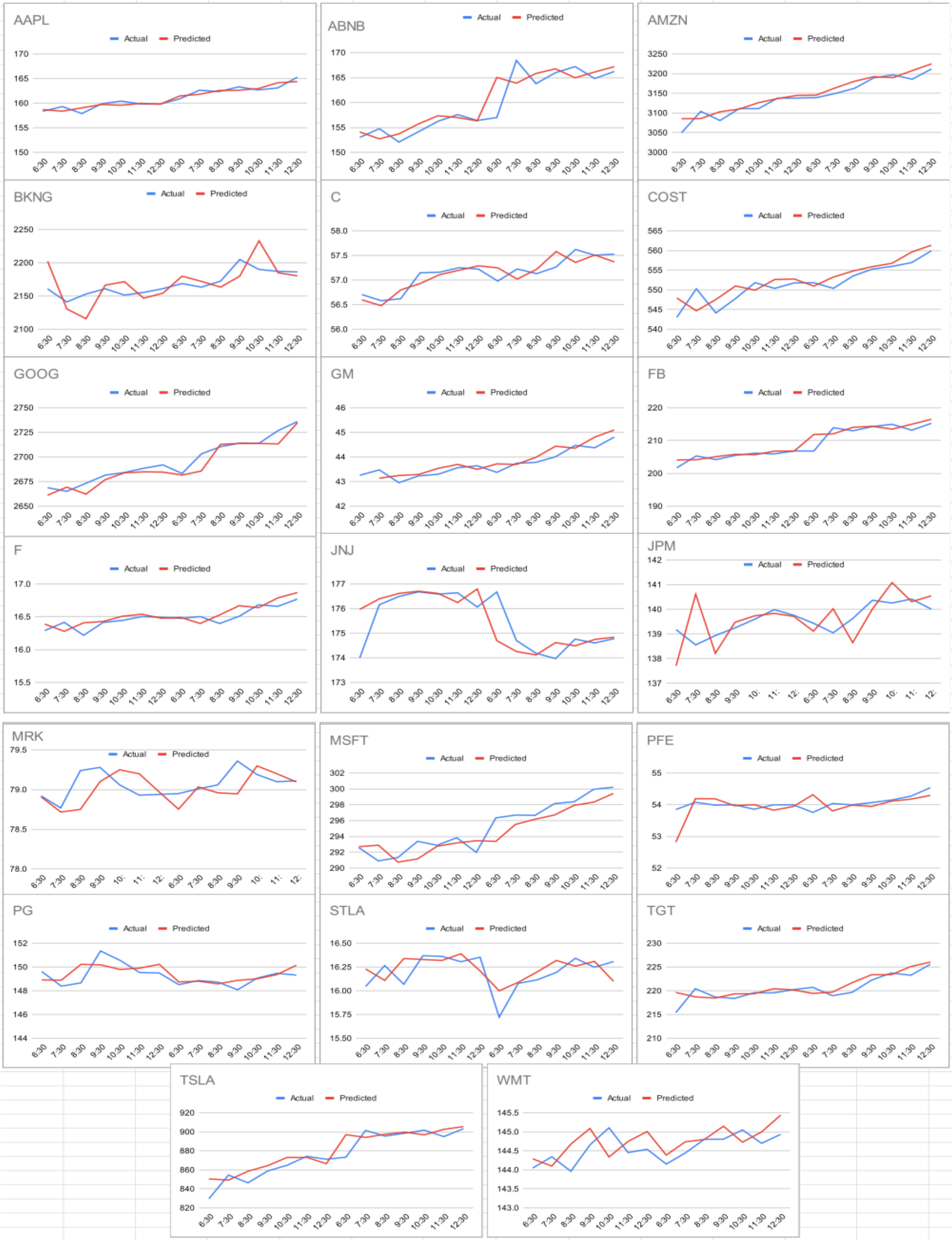
This study experienced several limitations. The original design of this study involved the implementation of this algorithm in a 14-day forecasting interval as opposed to 2-days. However, abnormal market conditions made longitudinal analysis of forecasting accuracy more influenced by current events than the actual performance of the algorithm. For this reason, this study was not able to produce a larger sample of forecasting data beyond 336 prediction intervals from over 100,000 price data points, an area that could be explored in future studies. Second, though the sample of predictions is large enough to be normal, this study did not incorporate a diverse set of market conditions. The market generally experienced a price increase, possibly skewing the profitability data's reliability. Lastly, it took up to 10 minutes to conclude the 24 forecasting models every market hour, which if used to make real sale executions, would affect the buy and sell price and profit by extension. Multiple servers and multiprocessing techniques may be used to improve time efficiency.

Future Studies

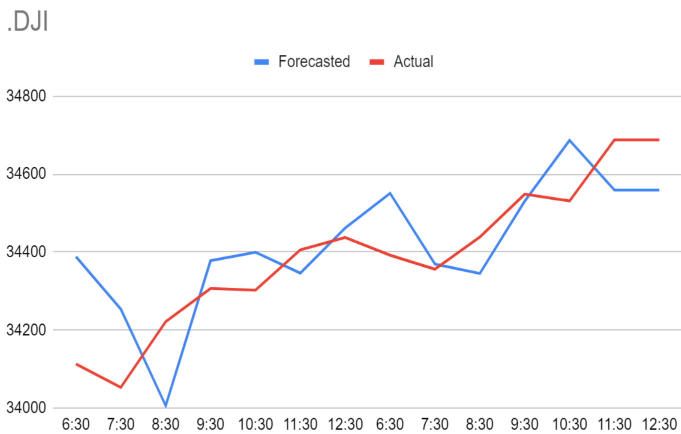
The use of a dynamic SARIMAx configuration likely allowed for this particular design to be more accurate than the Santur study's SARIMAx model. For this reason, it is highly recommended that a best-fit SARIMAx program is run during the period of time following after hours to update the SARIMAx coefficients for each stock for a better-fit model. Given that the algorithm exhibited a poor downtrend accuracy, the use of alternative (and possibly simpler) forms of LM may be more appropriate since SARIMAx-based LM reduced profit more times than it reduced loss. Future experiments should test other forms of LM, such as a stop-limit or trailing stop limit, to compare its performance. Future studies should also attempt to incorporate a greater variety of market conditions and stock types to test SARIMAx's overall performance in different market environments and usages, expanding the reliability of any supported hypotheses.

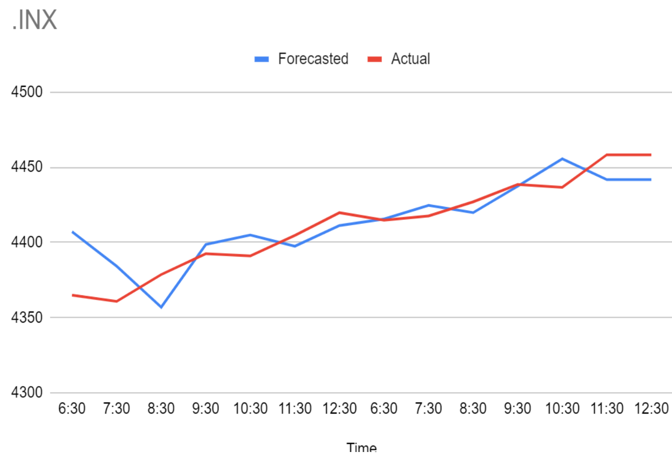
Given that major surrounding socio-political tensions had adversely affected the design of this study, it may be advisable for future research to control for such confounding variable using other prediction techniques. The use of sentiment analysis of reported market conditions or stock prices could be incorporated alongside the SARIMAx numerical forecasting method to prevent the investment of funds in companies or stock market conditions that are likely to experience an abnormal decrease in value.

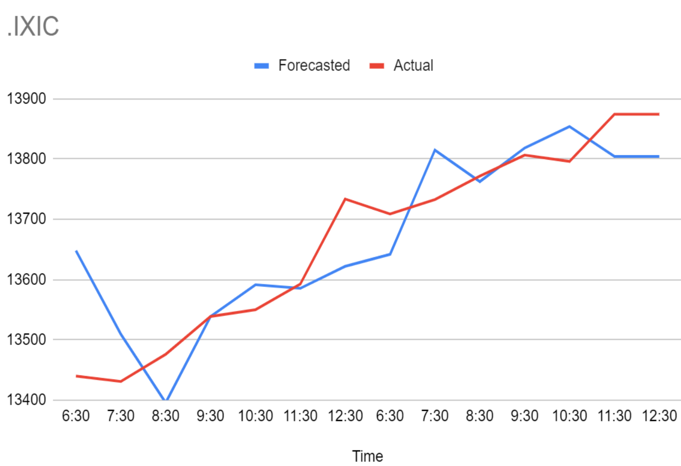
Appendix A: Stock List, Price Trends, & Forecasts (3/17/22 - 3/18/22)

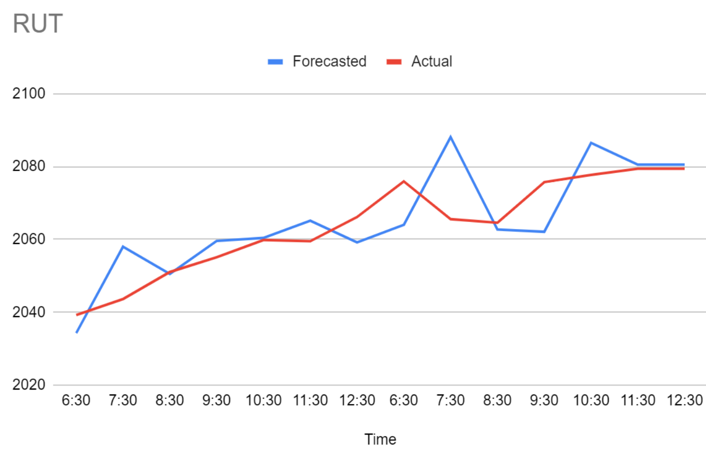


Appendix B: Index List, Price Trends, & Forecasts (3/17/22 - 3/18/22)









Appendix C: Sample Price Collection Output

```
interval 5 concluded, waiting 1 minute.

Stock: AAPL : NASDAQ            Price: $168.00          Time: 11:29
Stock: GOOG : NASDAQ            Price: $2,630.17                Time: 11:29
Stock: MSFT : NASDAQ            Price: $289.46          Time: 11:29
Stock: AMZN : NASDAQ            Price: $3,064.34                Time: 11:29
Stock: FB : NASDAQ              Price: $208.24          Time: 11:29
Stock: PFE : NYSE               Price: $48.71           Time: 11:29
Stock: JNJ : NYSE               Price: $164.09          Time: 11:29
Stock: MRK : NYSE               Price: $76.44           Time: 11:29
Stock: PG : NYSE                Price: $159.53          Time: 11:29
Stock: GM : NYSE                Price: $48.74           Time: 11:29
Stock: STLA : NYSE              Price: $18.96           Time: 11:29
Stock: F : NYSE         Price: $18.22           Time: 11:29
Stock: TSLA : NASDAQ            Price: $857.15          Time: 11:29
Stock: WMT : NYSE               Price: $138.35          Time: 11:29
Stock: TGT : NYSE               Price: $205.57          Time: 11:29
Stock: COST : NASDAQ            Price: $514.81          Time: 11:29
Stock: JPM : NYSE               Price: $152.61          Time: 11:29
Stock: C : NYSE         Price: $64.17           Time: 11:29
Stock: ABNB : NASDAQ            Price: $176.92          Time: 11:29
Stock: BKNG : NASDAQ            Price: $2,636.64                Time: 11:29
Stock: .DJI : INDEXDJX          Price: 34,266.22                Time: 11:29
Stock: .IXIC : INDEXNASDAQ              Price: 13,629.51                Time: 11:29
Stock: .INX : INDEXSP           Price: 4,371.12         Time: 11:29
Stock: RUT : INDEXRUSSELL               Price: 2,013.83         Time: 11:29

interval 6 concluded, waiting 1 minute.
```

Appendix D: Simulated Stock Order List ()

Name	Order	Time	Date	Purchase $	Current $	Predicted $	Delta	Actual $	Margin	Index
AMZN	buy	6:30	3/17/2022	3050	3050	3050.08	0.00%	3085.82	0	0
JPM	buy	6:30	3/17/2022	137.71	137.71	137.71	0.00%	139.18	0	1
BKNG	buy	6:31	3/17/2022	2167.05	2167.05	2202.44	1.63%	2160.85	0	2
C	buy	6:30	3/17/2022	56.6	56.6	56.71	0.19%	56.6	0	3
FB	buy	6:30	3/17/2022	201.41	201.41	201.72	0.15%	204.11	0	4
GM	buy	6:30	3/17/2022	42.88	42.88	43.26	0.88%	43.36	0	5
GOOG	buy	7:30	3/17/2022	2668.83	2668.83	2669.42	0.02%	2665.17	0	6
ABNB	buy	7:30	3/17/2022	154.1	154.1	154.76	0.43%	152.72	0	7
TSLA	buy	7:30	3/17/2022	850.56	850.56	854.58	0.47%	849.27	0	8
WMT	buy	7:30	3/17/2022	144.29	144.29	144.35	0.04%	144.1	0	9
TGT	buy	7:30	3/17/2022	219.66	219.66	220.46	0.36%	218.73	0	10
JNJ	buy	7:31	3/17/2022	175.97	175.97	176.16	0.11%	176.4	0	11
STLA	buy	7:30	3/17/2022	16.23	16.23	16.27	0.22%	16.11	0	12
COST	buy	7:30	3/17/2022	547.97	547.97	550.3	0.43%	544.71	0	13
AAPL	buy	7:30	3/17/2022	158.7	158.7	159.32	0.39%	158.38	0	14
C	sell	7:30	3/17/2022	56.6	56.6	56.58	-0.03233	56.48	0	15
PFE	buy	7:31	3/17/2022	53.85	53.85	54.19	0.64%	54.08	0	16
F	buy	7:30	3/17/2022	16.39	16.39	16.42	0.16%	16.28	0	17
BKNG	sell	7:31	3/17/2022	2167.05	2160.85	2130.69	-1.39591	2141.11	-0.2861	18
MSFT	buy	7:30	3/17/2022	292.6	292.6	292.9	0.10%	290.91	0	19
GOOG	sell	8:37	3/17/2022	2668.83	2665.17	2662.32	-0.10678	2673.35	-0.13714	20
JPM	sell	8:37	3/17/2022	137.71	138.56	138.22	-0.24433	138.94	0.61724	21
ABNB	sell	8:37	3/17/2022	154.1	152.72	152.08	-0.42076	153.72	-0.89552	22
AMZN	sell	8:37	3/17/2022	3050	3086.05	3080.95	-0.16526	3103.27	1.18197	23
TSLA	sell	8:37	3/17/2022	850.56	849.27	846.46	-0.33115	858.64	-0.15166	24
TGT	sell	8:37	3/17/2022	219.66	218.73	218.71	-0.00831	218.48	-0.42338	25
COST	sell	8:37	3/17/2022	547.97	544.71	544.17	-0.09994	547.58	-0.59492	26
WMT	sell	8:37	3/17/2022	144.29	144.1	143.96	-0.09385	144.68	-0.13168	27
AAPL	sell	8:37	3/17/2022	158.7	158.38	157.9	-0.30149	159.05	-0.20164	28
F	sell	8:37	3/17/2022	16.39	16.28	16.22	-0.37403	16.41	-0.67114	29
C	buy	8:37	3/17/2022	56.48	56.48	56.62	0.25%	56.8	0	30
STLA	sell	8:37	3/17/2022	16.23	16.11	16.07	-0.25531	16.34	-0.73937	31
MSFT	sell	8:37	3/17/2022	292.6	290.91	290.76	-0.0515	291.35	-0.57758	32
GM	sell	8:37	3/17/2022	42.88	43.14	42.96	-0.42552	43.25	0.60634	33
GOOG	buy	9:37	3/17/2022	2673.35	2673.35	2677	0.14%	2681.58	0	34
AMZN	buy	9:37	3/17/2022	3103.27	3103.27	3111.4	0.26%	3109.98	0	35
ABNB	buy	9:37	3/17/2022	153.72	153.72	154.15	0.28%	155.71	0	36
TSLA	buy	9:37	3/17/2022	858.64	858.64	858.82	0.02%	864.45	0	37

JPM	buy	9:37	3/17/2022	138.94	138.94	139.48	0.39%	139.25	0	38
COST	buy	9:37	3/17/2022	547.58	547.58	547.8	0.04%	551.01	0	39
AAPL	buy	9:37	3/17/2022	159.05	159.05	159.87	0.51%	159.73	0	40
F	buy	9:37	3/17/2022	16.41	16.41	16.42	0.03%	16.43	0	41
BKNG	buy	9:38	3/17/2022	2153.12	2153.12	2166.47	0.62%	2160.99	0	42
PG	buy	9:38	3/17/2022	150.23	150.23	151.36	0.75%	150.19	0	43
STLA	buy	9:37	3/17/2022	16.34	16.34	16.37	0.18%	16.33	0	44
PFE	sell	9:38	3/17/2022	53.85	53.99	53.97	-0.03397	54	0.25998	45
TGT	buy	10:39	3/17/2022	219.37	219.37	219.61	0.11%	219.39	0	46
WMT	buy	10:39	3/17/2022	145.09	145.09	145.11	0.01%	144.34	0	47
JNJ	sell	10:40	3/17/2022	175.97	176.71	176.59	-0.06587	176.62	0.42053	48
GM	buy	10:39	3/17/2022	43.29	43.29	43.3	0.02%	43.54	0	49
PFE	buy	10:40	3/17/2022	54	54	54	0.01%	53.86	0	50
JNJ	buy	11:42	3/17/2022	176.62	176.62	176.64	0.01%	176.25	0	51
STLA	sell	11:42	3/17/2022	16.34	16.32	16.31	-0.08136	16.39	-0.1224	52
BKNG	sell	11:43	3/17/2022	2153.12	2151.53	2146.86	-0.21695	2155.5	-0.07385	53
F	sell	11:42	3/17/2022	16.41	16.51	16.5	-0.0317	16.54	0.60938	54
PG	sell	11:42	3/17/2022	150.23	149.8	149.54	-0.17258	149.91	-0.28623	55
PFE	sell	11:42	3/17/2022	54	53.86	53.83	-0.05741	54	-0.25926	56
MSFT	buy	11:41	3/17/2022	292.89	292.89	293.18	0.10%	293.84	0	57
MRK	buy	11:41	3/17/2022	79.06	79.06	79.2	0.18%	78.93	0	58
GOOG	sell	12:41	3/17/2022	2673.35	2688.53	2684.69	-0.14285	2692.01	0.56783	59
JPM	sell	12:42	3/17/2022	138.94	139.99	139.72	-0.19521	139.76	0.75572	60
TSLA	sell	12:42	3/17/2022	858.64	873.19	871.26	-0.22097	866.55	1.69454	61
ABNB	sell	12:42	3/17/2022	153.72	156.99	156.4	-0.37473	156.33	2.12724	62
TGT	sell	12:42	3/17/2022	219.37	220.45	220.25	-0.08902	220.2	0.49232	63
JNJ	sell	12:42	3/17/2022	176.62	176.25	176.07	-0.10354	176.8	-0.20949	64
WMT	sell	12:42	3/17/2022	145.09	144.75	144.54	-0.14368	145.01	-0.23434	65
COST	sell	12:42	3/17/2022	547.58	552.64	551.79	-0.15325	552.79	0.92407	66
AAPL	sell	12:41	3/17/2022	159.05	159.96	159.83	-0.082	159.8	0.57215	67
GM	sell	12:42	3/17/2022	43.29	43.7	43.65	-0.1236	43.5	0.9471	68
MSFT	sell	12:41	3/17/2022	292.89	293.84	293.47	-0.12729	292	0.32435	69
GOOG	buy	6:30	3/18/2022	2680.21	2680.21	2681.68	0.05%	2683.29	0	70
JPM	buy	6:30	3/18/2022	139.1	139.1	139.12	0.01%	139.44	0	71
TSLA	buy	6:30	3/18/2022	873.25	873.25	873.35	0.01%	897.1	0	72
JNJ	buy	6:30	3/18/2022	176.47	176.47	176.68	0.12%	174.7	0	73
TGT	buy	6:30	3/18/2022	220.63	220.63	220.73	0.05%	219.46	0	74
C	sell	6:30	3/18/2022	56.48	57.06	56.98	-0.13886	57.25	1.02691	75
COST	buy	6:30	3/18/2022	551.25	551.25	551.84	0.11%	550.98	0	76
BKNG	buy	6:30	3/18/2022	2155.74	2155.74	2179.92	1.12%	2168.83	0	77

AAPL	buy	6:30	3/18/2022	160.6	160.6	160.9	0.19%	161.48	0	78
GM	buy	6:30	3/18/2022	43.36	43.36	43.38	0.04%	43.72	0	79
F	buy	6:30	3/18/2022	16.46	16.46	16.48	0.10%	16.49	0	80
MRK	sell	6:30	3/18/2022	79.06	78.78	78.75	-0.03278	78.95	-0.35416	81
ABNB	buy	7:31	3/18/2022	165.03	165.03	168.44	2.07%	163.89	0	82
WMT	buy	7:30	3/18/2022	144.39	144.39	144.45	0.04%	144.74	0	83
PG	buy	7:31	3/18/2022	148.75	148.75	148.87	0.08%	148.81	0	84
TGT	sell	7:30	3/18/2022	220.63	219.46	218.99	-0.21511	219.73	-0.5303	85
COST	sell	7:30	3/18/2022	551.25	550.98	550.41	-0.10408	553.28	-0.04898	86
PFE	buy	7:30	3/18/2022	53.76	53.76	53.81	0.09%	54.04	0	87
STLA	buy	7:30	3/18/2022	16	16	16.08	0.49%	16.09	0	88
MRK	buy	7:30	3/18/2022	78.95	78.95	79.03	0.11%	79.01	0	89
JPM	sell	8:30	3/18/2022	139.1	139.05	138.65	-0.28426	139.63	-0.03595	90
AMZN	sell	8:30	3/18/2022	3103.27	3163.95	3162.92	-0.03259	3180.79	1.95536	91
ABNB	sell	8:31	3/18/2022	165.03	163.89	163.78	-0.06422	165.83	-0.69078	92
JNJ	sell	8:31	3/18/2022	176.47	174.26	174.19	-0.04097	174.12	-1.25234	93
C	buy	8:30	3/18/2022	57.02	57.02	57.13	0.19%	57.22	0	94
COST	buy	8:30	3/18/2022	553.28	553.28	553.55	0.05%	554.81	0	95
PFE	sell	8:30	3/18/2022	53.76	54.04	53.99	-0.09526	54	0.52083	96
F	sell	8:30	3/18/2022	16.46	16.4	16.4	-0.00882	16.53	-0.36452	97
PG	sell	8:31	3/18/2022	148.75	148.81	148.74	-0.05012	148.59	0.04034	98
MRK	sell	8:30	3/18/2022	78.95	79.01	78.96	-0.06303	79.06	0.076	99
JPM	buy	9:30	3/18/2022	139.63	139.63	140.01	0.27%	140.37	0	100
ABNB	buy	9:31	3/18/2022	165.83	165.83	165.99	0.10%	166.76	0	101
AMZN	buy	9:30	3/18/2022	3180.79	3180.79	3189.5	0.27%	3192.32	0	102
TGT	buy	9:30	3/18/2022	221.71	221.71	222.25	0.24%	223.39	0	103
STLA	sell	9:30	3/18/2022	16	16.2	16.19	-0.03442	16.32	1.25	104
MSFT	buy	9:30	3/18/2022	296.67	296.67	296.72	0.02%	298.15	0	105
GOOG	sell	10:30	3/18/2022	2680.21	2714.08	2713.77	-0.01151	2713.95	1.26371	106
WMT	sell	10:30	3/18/2022	144.39	145.15	145.05	-0.06696	144.73	0.52635	107
JNJ	buy	10:31	3/18/2022	174.62	174.62	174.76	0.08%	174.49	0	108
PFE	buy	10:30	3/18/2022	54.07	54.07	54.11	0.08%	54.15	0	109
PG	buy	10:31	3/18/2022	148.88	148.88	149.07	0.13%	149.02	0	110
F	buy	10:30	3/18/2022	16.67	16.67	16.68	0.08%	16.64	0	111
STLA	buy	10:30	3/18/2022	16.32	16.32	16.34	0.14%	16.26	0	112
MSFT	sell	10:30	3/18/2022	296.67	298.15	297.94	-0.07191	298.39	0.49887	113
AMZN	sell	11:30	3/18/2022	3180.79	3189.97	3185.99	-0.12489	3207.79	0.28861	114
TGT	sell	11:30	3/18/2022	221.71	223.39	223.28	-0.04812	225.11	0.75775	115
ABNB	sell	11:31	3/18/2022	165.83	164.96	164.82	-0.08576	166.12	-0.52463	116
TSLA	sell	11:30	3/18/2022	873.25	896.9	895.06	-0.20517	902.63	2.70827	117

FB	sell	11:30	3/18/2022	201.41	213.51	213.18	-0.15684	215	6.00765	118
BKNG	sell	11:31	3/18/2022	2155.74	2190.05	2185	-0.23073	2187.11	1.59156	119
STLA	sell	11:30	3/18/2022	16.32	16.26	16.25	-0.06997	16.31	-0.36765	120
MRK	buy	11:30	3/18/2022	79.19	79.19	79.2	0.01%	79.1	0	121
F	sell	12:30	3/18/2022	16.67	16.79	16.77	-0.11159	16.87	0.71986	122
GM	sell	12:30	3/18/2022	43.36	44.81	44.81	-4.72E-06	45.1	3.3441	123
PG	force sell	12:31	3/18/2022	148.88	149.36	149.33	-0.0224	150.15	0.32241	124
MRK	force sell	12:30	3/18/2022	79.19	79.1	79.1	-0.00368	79.11	-0.11365	125
AAPL	force sell	12:30	3/18/2022	160.6	164.15	163	0	0	0.022105	126
C	force sell	12:30	3/18/2022	57.02	57.51	57	0	0	0.008593	127
COST	force sell	12:30	3/18/2022	553.28	559.64	550	0	0	0.011495	128
JPM	force sell	12:30	3/18/2022	139.63	140.42	139	0	0	0.005658	129
JNJ	force sell	12:30	3/18/2022	174.62	174.77	174	0	0	0.000859	130
PFE	force sell	12:30	3/18/2022	54.07	54.27	54	0	0	0.003699	131

References

Ahmad, Muhammad Ishfaq, et al. (2017, Jun.)“Assesing Performance of Moving Average Investment Timing Strategy over the Uk Stock Market.” Journal of Developing Areas, vol. 51, no. 3, pp. 349–362. EBSCOhost, doi:10.1353/jda.2017.0077.

Analytics & Intelligent Automation Time Series Analysis with ARIMA: Part 2 Bradley Wise. (2020, Oct. 5). *Time series analysis with Arima: Part 2*. Cisco Blogs, from https://blogs.cisco.com/analytics-automation/arima2

Chakole, Jagdish, and Manish Kurhekar. (2020, Aug.)“Trend Following Deep Q‐Learning Strategy for Stock Trading.” Expert Systems, vol. 37, no. 4, pp. 1–16. EBSCOhost, doi:10.1111/exsy.12514.

Choi SB, Ahn I. (2020) Forecasting imported COVID-19 cases in South Korea using mobile roaming data. PLoS ONE;15(11):1-10. doi:10.1371/journal.pone.0241466

Demirel, Ugur, et al. (2021, Mar.) “Predicting Stock Prices Using Machine Learning Methods and Deep Learning Algorithms: The Sample of the Istanbul Stock Exchange.” Gazi University Journal of Science, vol. 34, no. 1, pp. 63–82. EBSCOhost, doi:10.35378/gujs.679103.

Definition and types of moving averages: Orderhive. Cin7. (n.d.). from https://www.cin7.com/industry-terms/moving-average/

Hayes. A (2022, Feb. 8) *Autoregressive Integrated moving average (ARIMA)*. Investopedia, from https://www.investopedia.com/terms/a/autoregressive-integrated-moving-average-arima.asp

Huotari, Tommi, et al. (2020, Dec.) “Deep Reinforcement Learning Agent for S&P 500 Stock Selection.” Axioms (2075-1680), vol. 9, no. 4, p. 130. EBSCOhost, doi:10.3390/axioms9040130.

Kouaissah, Noureddine, et al. (2020, Jan.) “Theoretical and Practical Motivations for the Use of the Moving Average Rule in the Stock Market.” IMA Journal of Management Mathematics, vol. 31, no. 1, pp. 117–138. EBSCOhost, doi:10.1093/imaman/dpz006.

Kokaz, A. (2020, Jul. 8) *The types of automated trading algorithms*. Medium., from https://medium.datadriveninvestor.com/the-types-of-automated-trading-algorithms-228d537254a8

Kuepper, J. (2021, Sept. 8). *Risk management techniques for active traders*. Investopedia. Retrieved January 12, 2022, from https://www.investopedia.com/articles/trading/09/risk-management.asp

Luo, Tai-Li, et al. (2020, May 5)“A Framework of Deep Reinforcement Learning for Stock Evaluation Functions.” Journal of Intelligent & Fuzzy Systems, vol. 38, no., pp. 5639–5649. EBSCOhost, doi:10.3233/JIFS-179653

Lydia Saad and Jeffrey M. Jones. (2021, November 20). *What percentage of Americans owns stock?* Gallup.com, from https://news.gallup.com/poll/266807/percentage-americans-owns-stock.aspx

Mitchell, C. (2022, Apr. 4). *Rules for picking stocks when Intraday trading*. Investopedia, from https://www.investopedia.com/day-trading/pick-stocks-intraday-trading/

Mundra, A., et al. (2020). A deep learning based hybrid framework for stock price prediction. Journal of Intelligent & Fuzzy Systems, 38(5), 5949–5956. https://doi.org/10.3233/JIFS-179681

Murphy, C. B. (2022, Feb. 8). *Why do companies care about their stock prices?* Investopedia, from https://www.investopedia.com/investing/why-do-companies-care-about-their-stock-prices/

Richardson, L. (2007). Beautiful soup documentation.

Richard Hardy. (2012, Sept. 1). *Sarima model equation*. Cross Validated, from https://stats.stackexchange.com/questions/129901/sarima-model-equation

Santur, Yunus. (2020, Oct.)“Deep Learning Based Regression Approach for Algorithmic Stock Trading: A Case Study of the Bist30.” Gümüshane Üniversitesi Fen Bilimleri Enstitüsü Dergisi, vol. 10, no. 4, pp. 1195–1211. EBSCOhost, doi:10.17714/gumusfenbil.707088.

Samuel Joe Samuel StockMarket.com, J. (2021, May 30). *The 11 sectors of the Stock Market & their biggest etfs*. Nasdaq, from https://www.nasdaq.com/articles/the-11-sectors-of-the-stock-market-their-biggest-etfs-2021-05-30

Seabold, Skipper, and Josef Perktold. (2010) “statsmodels: Econometric and statistical modeling with python.” Proceedings of the 9th Python in Science Conference.

Statistical forecasting: notes on regression and time series analysis. Duke Mathematics. (n.d.), from https://people.duke.edu/~rnau/411home.htm

Thorlie, Milton Abdul, et al. (2015, Aug.) “Modeling and Forecasting of Stock Index Volatility with APARCH Models under Ordered Restriction.” Statistica Neerlandica, vol. 69, no. 3, pp. 329–356. EBSCOhost, doi:10.1111/stan.12062.

Yadav, Yesha. (2015, Nov.) “How Algorithmic Trading Undermines Efficiency in Capital Markets.” Vanderbilt Law Review, vol. 68, no. 6, pp. 1607–1671. EBSCOhost, search.ebscohost.com/login.aspx?direct=true&db=asn&AN=111246205&site=ehost-live.