A Work Project, presented as part of the requirements for the Award of a Master's degree in Finance from the Nova School of Business and Economics.

The ARIMA model for stock price prediction: studying the impact of news announcements through the use of agent-based modelling.

David Godinho Vieira Duarte Soares (55675)

Work Project carried out under the supervision of:

Pedro Souto

17/05/2024

Abstract

The relationship between new information and stock price movement is of great interest for investors. News announcements generally cause changes in stock prices, as they affect public perception of products and companies. An accurate predictive model allows us to take advantage of this unexpected movement. In this project, we use Agent-Based modelling to study the impact of news announcements on the ARIMA model's accuracy. Overall, we conclude that a significant shock can cause long-term effects on the model's precision.

Keywords: Agent-based models, ARIMA model, News, Stock Price Prediction

Acknowledgements: I'm grateful for the support given by my thesis supervisor Pedro Souto, as well as the helpful feedback given by my family and friends.

This work used infrastructure and resources funded by Fundacão para a Ciência e a Tecnologia (UID/ECO/00124/2013, UID/ECO/00124/2019 and Social Sciences DataLab, Project 22209), POR Lisboa (LISBOA-01-0145-FEDER-007722 and Social Sciences DataLab, Project 22209) and POR Norte (Social Sciences DataLab, Project 22209).

1. Introduction

Whether through the TV, radio, newspapers, or smartphones, we are always, sometimes involuntarily, subjected to new information. News spread faster and further than ever, greatly impacting multiple aspects of our day to day lives. The news (good or bad) can affect our perception of companies or products, which may ultimately affect their valuation. As such, news constantly affect the stock market, as seen by Alanyali, Moat, and Preis (2013), who reported positive correlation between news announcements and trading volume. In this project, we will be looking into the effects of these events on the stock market through the lenses of Agent-Based Modelling.

Agents-Based Models (ABMs) are computational models which replicate intricate real-life systems, by allowing the simulation of the behavior of autonomous agents and observing the way in which they interact with their environment and with each other. In this case, the environment will be the stock market and the agents will be traders. To do this, we will need to define how the market will work and how each trader interacts with it, creating the most realistic scenario possible.

The main advantage that we get from using this type of modelling is that it is extremely customizable. We can control every characteristic of the market and its participants, enabling us to elaborate a wide range of relevant queries from which we can obtain potentially interesting conclusions that are essential to understand market dynamics. We can, for instance, control how many traders of each type place orders in our market. This allows us to study the individual impact of each trader and precisely adjust the behavior of each agent in each independent scenario. We can define our initial market characteristics with great flexibility, greatly impacting outcomes and stepping into previously unexplored scenarios. Hence, Agent-Based Modeling allows for endless customization options.

We will study a widely used predictive model, the ARIMA, and understand its behavior when exposed to unexpected shocks in stock prices. These are mostly caused by news announcements, as they provide new insights into companies and the market in general.

This project takes advantage of Agent Based Modelling to study the effect of news announcements, or other events that create significant shocks in the market, in the capability of the ARIMA model to forecast stock prices. In the following chapter, we revise literature on the concepts that are most essential to construct this model.

2. Literature Review

2.1. Noise Traders

The first type of trader acting in our market is the noise trader. This type of trader usually refers to uninformed or inexperienced traders, who appear to have no clear reasoning behind their trading choices. They introduce unpredictability into the market and usually, for the sake of simplicity, it is assumed that they trade "randomly".

Several studies aimed to identify noise traders and understand how they operate, to better comprehend market dynamics and microstructure. For example, S. Grossman (1976) proposes a model which studies the interaction between "informed" and "uninformed" traders, showing that an efficient market will reveal the data that the "informed" traders have, by converging to a certain price. However, this removes the advantages of informed traders, as uninformed traders know that prices will reflect the knowledge of informed traders. Consequently, the market collapses as informed traders are not rewarded for the costly information gathered. It is the uncertainty brought by noise traders that allows informed traders to make a profit. Thus, "noise" is essential for informed traders to reap the benefits of acquiring information, even though the "noise" is not strictly caused by uninformed traders, but also by "many other factors" (S. J. Grossman 1977, 431).

Kyle (1985) later tests the idea of "uninformed noise traders who trade randomly"(p.1315), that is, he tests the idea of a class of traders who act in a way which introduces "noise" into the market. However, this is swiftly rebutted by Gorton and Dow (2006) who state that "noise traders in this new type of model were not well-motivated. In fact, their motives are not explained."(p.3)

In summary, the understanding of the identity and behavior of noise traders is of major importance since their irrational behavior should be taken advantage of, leading to their eradication (Gorton and Dow 2006). However, this remains a widely discussed topic and for simplicity we will assume that noise traders buy and sell randomly in our market.

2.2. Momentum Traders

The next essential type of trader in our market are momentum traders, whose strategy consists of buying past winners and selling or short selling the losers. These traders try to capitalize on market trends, by following the general direction of the market. In a simple way, momentum refers to the tendency of prices to keep dropping or rising.

Momentum has been studied for decades now, with Jegadeesh and Titman (1993) concluding that buying the best performing stocks, while shorting the worst performing ones, yielded abnormal profits of 1% when holding this portfolio for 3, 6, 9 and 12 months. These positive results have since been corroborated by various studies in different markets, with Asness, Moskowitz, and Pedersen (2013) finding a consistent premium for momentum strategies when jointly evaluating momentum across eight different markets.

However, as initially suggested by Jegadeesh and Titman (1993), when factoring in trading costs, these strategies become unprofitable, a result later corroborated by Carhart (1997). Even so, because the specific environment created for this project assumes no trading costs, momentum traders remain relevant in the market and will constitute its majority as they are relatively simple to implement and yield realistic market movement.

2.3. Agent-Based Models

As previously stated, Agent Based Modelling consists of programming different agents and their environment, as well as the interactions between them. In this case, we will be building an agent-based model of the financial market.

ABMs bring the advantages of flexibility and customization. This facilitates and expands our possibilities in economic research, especially in chaotic and dynamic systems such as the stock market, where agents have different information, mindset, and goals. Unlike human-based experiments, Agent-Based Modeling allows for the representation of diverse behaviors, information asymmetry, and varying preferences (Poggio et al. 2001). Consequently, they have become a useful tool to study the financial market.

The use of these type of models in financial markets was led by Garman (1976) and later adopted by Cohen et al. (1983), and Hakansson, Beja, and Kale (1985), the early pioneers of these methods. Later, Gode and Sunder (1993) presented a zero-intelligence model, where traders place buy and sell orders randomly. He uses it to demonstrate that, as long as budget constraints are followed, even with randomly generated orders, this model could help reconcile utility maximization and its criticisms. These critiques arise from the fact that this maximization is not always consistent with observations, contradicting the idea that traders always maximize utility.

Zero-intelligence models were improved by adding different degrees of intelligence to agents, reducing the randomness of their orders. Usually this is done by restricting the range of possible orders, based on factors such as recently observed trades and other external factors (Jamal and Sunder 1996).

More complex models which try to better understand the market are reviewed by LeBaron (2000). He concludes that, while these types of models provide a new valuable approach to economic research, they still require significant progress to become reliable theoretical structures for policy makers.

2.4. ARIMA Predictive model

The Autoregressive Integrating Moving Average (ARIMA) model was created by Box and Jenkins in their 1970 book: "Time Series Analysis: Forecasting and Control". It is a time series predictive model, widely used across all areas that benefit from time series prediction, including financial markets.

The various applications of such a predictive model allow its use across all types of studies in different areas. For instance, Contreras et al. (2003) used the ARIMA model to forecast electricity prices, while Lee and Ko (2011) write about "Short-term load forecasting using lifting scheme and ARIMA model".

Because this model takes 3 inputs, p, d and q, it is usually denoted as ARIMA(p,d,q). The variable p refers to the number of autoregressive terms, that is the number of lagged terms of the autoregression, while q refers to the number of lagged error terms of the moving average model. The term d is the degree of subtracting performed on the data and is better explained in the following examples. The general formula goes as follows (Nau 2020):

$$\hat{y}_t = \mu + \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} - \theta_1 e_{t-1} - \cdots - \theta_q e_{t-q} \qquad (1)$$

We denote y as the d^{th} difference of T:

$$d = 0 \qquad y = Y_t \qquad (2)$$

$$d = 1 \qquad y = Y_t - Y_{t-1} \qquad (3)$$

$$d = 2 \qquad y = (Y_t - Y_{t-1}) - (Y_{t-1} - Y_{t-2}) \qquad (4)$$

…

Therefore, ARIMA(1,0,0) is expressed as follows:

$$\hat{Y}_t = \mu + \phi_1 Y_t \quad (5)$$

ARIMA(1,1,0):

$$\hat{Y}_t = \mu + Y_{t-1} + \phi_1 (Y_{t-1} - Y_{t-2}) \quad (6)$$

The robustness and efficiency of this model in stock price forecasting, especially in the short term, has proven to have satisfactory results compared to other models (Merh, Saxena, and Pardasani 2010). Ariyo, Adewumi, and Ayo (2014) concluded that ARIMA models provide adequate predictions in the short term and considered that this could potentially guide investors toward profitable strategies and trades.

3. Methodology

3.1. Building an order Book.

To build this ABM, an order book was implemented using a python notebook. An order book is a memoir of all the desired transactions by all traders. Whenever a trader wishes to sell a stock, he places an "ask" order, with the corresponding desired sell price. Conversely, if said trader wishes to buy a stock, he places a "bid" order. An order book keeps an account of all "bid" and "ask" prices, that is, the price at which each trader intends to buy or sell a given stock, respectively.

To achieve this, a python class was created, as we will need to use a series of functions and constants related to this notebook. By creating an OrderBook class, we can create all the attributes required for this class in order to facilitate coding.

In our order book, all bids must be lower or equal to the lowest "ask" price. Conversely, all "ask" prices must be higher or equal to the highest "bid" price. This is because, realistically, no trader would buy an asset at a higher price than the lowest one available, and vice versa. Following this logic, a transaction occurs every time the lowest "ask" price is equal to the highest "bid". In other words, we register a transaction when two traders agree to trade the asset at the same price. Figure 1 illustrates how a typical order book works:

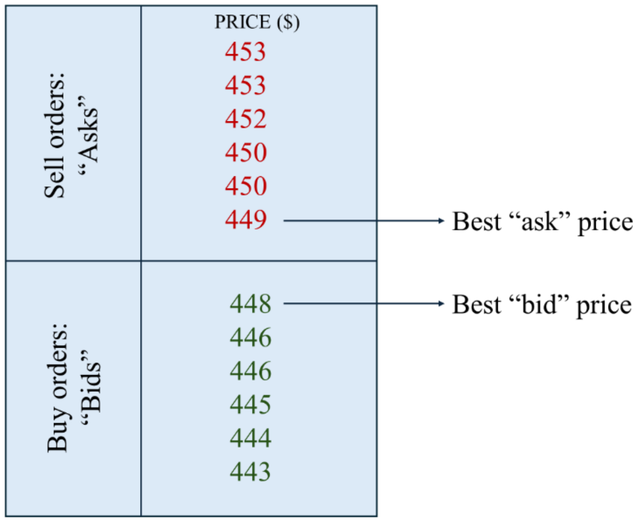

Figure 1: Order Book Example

To achieve this, we create two functions, "match_order" and "simulation". They allow us to simulate the behavior of an order book. Each time an order is registered, "match_order" checks the market for any possible transactions. To do so, it first sorts the bid and ask lists. Then, it checks if the "best bid" is equal to the "best ask" and, if it is, it processes the transaction.

The "simulation" function, as the name suggests, is responsible for simulating the market. That is, it runs a loop in which each trader places an order per day. With the help of the "match_order" function, it makes sure that all transactions are recorded, and the asset price is readjusted accordingly.

3.2. Trader Class

After creating the order book, the class Trader was defined. This class has many subclasses, each representing a different type of trader, with distinct behaviors and trading strategies. These subclasses will facilitate and organize the interactions between themselves and the order book. The two trader types that were implemented were: noise and momentum traders. When combined, these two trader types are able to create a sufficiently dynamic and realistic financial market. Importantly, we left out some relevant types of traders such as the fundamental traders, as this was not compatible with the type of order book and market we were creating. Fundamental traders base their trading strategies on information about the companies, such as financial statements. This is information that we do not have in the current environment. In this setting, agents do not learn, and act randomly or with very incomplete information.

A crucial step in this code will be the implementation of one or several news-sensitive traders, which are traders who react to news announcements. Such traders will be the center piece of this project, allowing us to simulate the impact of news in the market. However, we should expect non-news-sensitive traders to be impacted by news announcements, even if indirectly. For example, a momentum or volatility timing trader might be indirectly affected, since their trading signal could shift based on the reaction of the news-sensitive traders.

3.3. The ARIMA prediction model

Firstly, we separate the data into training and testing lists. We attribute 80% of the historic values to the training dataset, and the remaining 20% to the testing dataset. The former will help us train the data and adjust our model accordingly, allowing us to choose the appropriate previously mentioned values for p, d, and q. After that, we will use the latter to make stock price predictions.

We implement this model mainly using two functions. The first is the "arima_forecast", which, given a list (in this case a list of returns) and a tuple (in this case composed by the three inputs (p,d,q) of an ARIMA model), returns a forecast for the next day's price. The other is "arima_forecast_list", which, by using arima_forecast recursively, provides a list of all the predictions we need. This function uses our training dataset to predict the following day's price by using arima_forecast. After that, it adds the prediction to a list of predicted values, while adding the following day's real price to the training set. It does this cycle until we run out of real prices in the testing dataset (Kanaskar 2023).

We are then able to plot the two separate lists of the actual prices and the predicted prices, which helps us visualize the behavior and accuracy of the model. This will be useful to study the impact that a large shock in the market has on the model.

To precisely measure the effectiveness and accuracy of the model through our time period, we will use other mathematical tools such as the Root Mean Squared Error (RMSE), the Mean Absolute Error (MAE) and the Bias (refer to the appendix for formulas). With these metrics, we will test the accuracy of the ARIMA in our environment and check for any interesting biases.

4. Our market's characteristics

Using the proposed methodology, we obtain a fully functional market composed of noise, momentum and news traders. However, as expected, this simulation does not share the exact same characteristics and properties of a real financial market, as many factors and types of traders were not included in this simulation. Because of this, it is very important to keep these differences in mind when analyzing our outputs and drawing conclusions. As such, in this section we look at the properties of this particular environment and try to rationalize and explain them.

After experimenting with various values, it was decided that there should be 10 traders of each category in the market. This means that there are 10 noise traders and 60 momentum traders in total, as we implemented 6 different types of momentum traders. This proportion of noise traders is close to the estimate of 9,3% made by Ahmed (2019), while maintaining our market functional and volatile. Firstly, we look at our market without the intervention of news traders, and later look at their impact on the market.

In Figure 2 we can observe three distinct examples of the market movement we obtain with this model. As expected, every simulation yielded very different results as there is a significant randomness factor in the model, coming from the noise trader. These traders impact the market directly while also indirectly affecting momentum traders and their decisions.

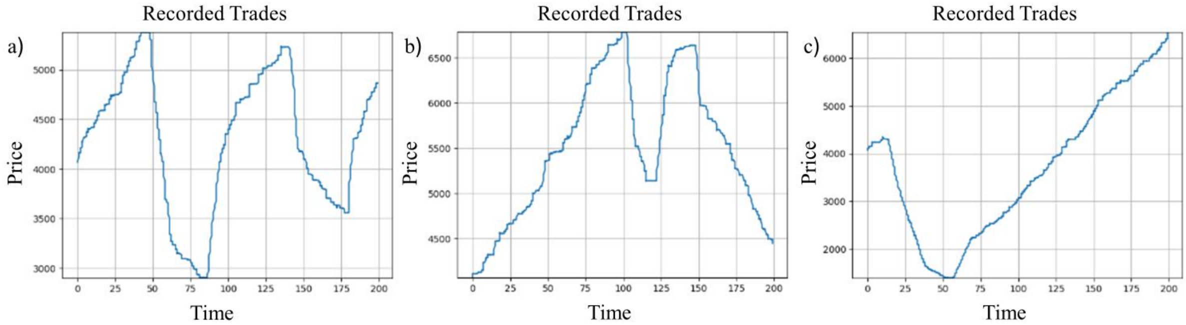

Figure 2. Market simulations. a, b and c show three distinct runs, illustrating the randomness of the model.

The observed disparity in the of results we obtain in our simulations is caused by the randomness factor and the overall simplicity of the environment created. Furthermore, we confirm that the simulations sometimes yield rather unrealistic stock values and price changes. This makes it essential to run several simulations to draw conclusions with this ecosystem. As a reference we observe in figure 5 the S&P500's performance in 2023:

Figure 3. S&P500 2023 plot.

An important factor to consider in any market is its volatility, a metric which tells us what type of movement to expect, and the amount of risk we are exposed to. In this case, we look at daily volatility, that is, we compute returns by looking at the daily closing price. We run 1000 simulations, and we get an average standard deviation of 0,94%, which is still way lower than the SP500's volatility of 15% in the last year, according to Financial Times. Figures 2 and 3 provide an explanation for this. Our market tends to follow a relatively steady and predictable trajectory, unlike the SP500, which has unpredictable drawdowns and unsteady movement. This is mainly due to the fact that we mostly use momentum traders, which trade in the same direction.

Additionally, we look at the Maximum Drawdown, an important measure of risk of any asset, which calculates the maximum loss incurred in a certain time period. After simulating 1000 instances of this market, we obtain an average maximum drawdown of 58% which is significantly higher than the SP500's usual drawdown. For context, in the last 20 years the biggest drawdown was 49%, in 2008. Once again, we credit this property to the herding behavior of momentum traders, which trade in the same direction most of the time.

Now we look at the impact of news traders in our market. At the middle point of our time period, a news announcement was "planted". We expect "good" news to shock the market positively and "bad" news to do the opposite. Later we will see if they impact our predictive model differently. In Figures 4 and 5 we can visualize some examples of their impact:

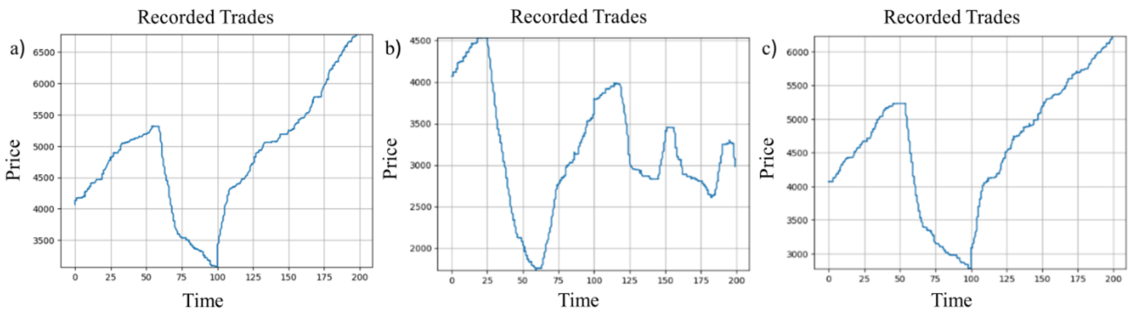

Figure 4. Market simulations. a, b and c show three distinct runs, with the announcement of good news.

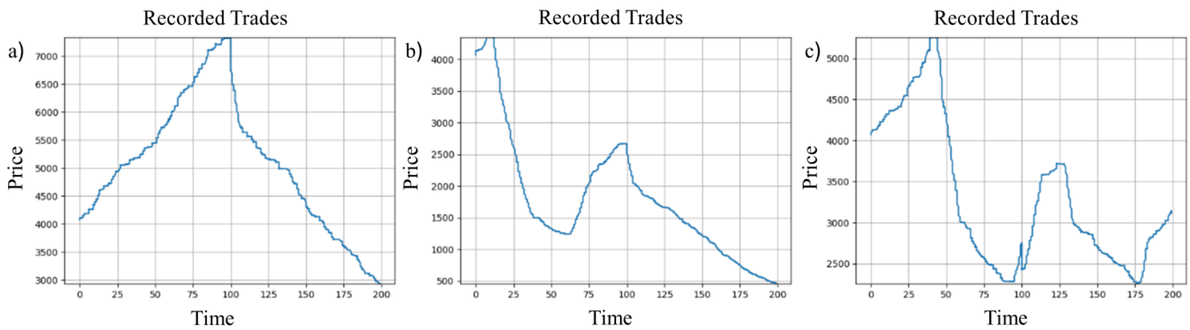

Figure 5. Market simulations. a, b and c show three distinct runs, with the announcement of bad news.

One can observe examples of good and bad news, in Figures 4 and 5, respectively. As expected, on this day, good news shock the market in around 15,5%. On the other hand, bad news create an 11.8% drop in stock price.

With news traders in our market, the volatility goes up slightly, reaching an average of 1,43%, which is expected as they cause a substantial shock in the market that impacts traders in the days following the announcement. Additionally, in the second half of this period of time (after the shock) the volatility goes up to 1,77%, confirming that the news actually create considerable volatility in our market. This could be impactful to our predictive model, as an unexpected shock in volatility is sure to cause some disturbance in its ability to predict stock prices. On the other hand, both types of news seem to impact the maximum drawdown equally, as they both average 60% for this metric.

Because our main goal is to study the impact of news on the ARIMA model, it is relevant to first study their impact on the market itself. Consequently, we analyze how long it takes for volatility to go back to our initial values, by looking at different 20-day intervals after the announcement. Table 1 shows the evolution of volatility over time, before and after the news announcement:

	0 days later	2 days later	5 days later
Before	0.76%	0.71%	0.72%
After	3.28%	0.81%	0.77%

Table 1. Volatility over time. Evolution of volatility in the days following a shock in stock price.

Table 1 shows that after only 5 days the effects of the market shock are negligible when looking at volatility. This is important to note as, if the impact was longer-lasting, the added volatility could potentially affect the model's accuracy. This way, we can be sure that any change in the model's behavior is strictly attributed to its capabilities.

5. Results

5.1. Pre-News ARIMA model

To start our analysis and evaluate the impact of a news announcement on the ARIMA model, we first need to look at how the model behaves without news and establish this as our baseline. As previously stated, the first 80% of prices generated by our model will serve as a training dataset and the remaining 20% will be used as a testing dataset. This means that if we run a 200-day simulation, we will have 40 days of predictions made by the autoregressive model.

As shown in Figure 6, the model seems relatively accurate at predicting prices, justifying its use in this thesis. The model is capable of following new trends, usually with a small delay as expected. In Figure 6, "Real Prices" are the prices generated by our market.

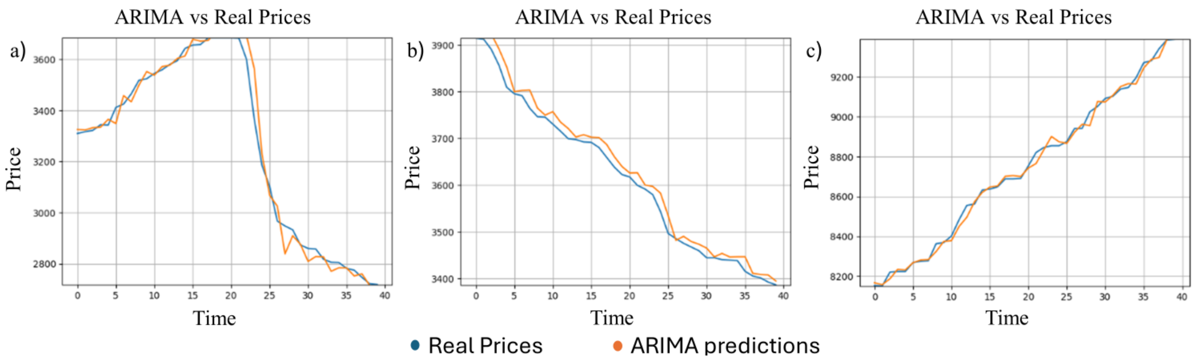

Figure 6. ARIMA *versus* Real Prices. a, b and c show three examples of the ARIMA model predictions *versus* prices generated by our ABM.

The next step in evaluating the model is obtaining its RMSE, MAE and Bias metrics. These serve as a measure of how "distant" from the actual prices the predictive model is, and as a baseline for the forthcoming introduction of news in our market. The Bias metric will inform as to the tendency of the model to over or undershoot the stock price. This analysis will demand the use of less iterations than before, as predictive modelling is a computationally heavy task, requiring fitting the model and several iterations every time we wish to simulate results. In this case, we will use 100 iterations, so we need to take into consideration that these results may not be as accurate as we would like. For further information, we will calculate the maximum difference between the predicted prices and the actual price generated by our market. That is, we will observe the day on which the prediction was furthest away from the real price. For these 100 iterations such means for RMSE, MAE and Bias metrics were 41.38, 25.07 and -1.19, respectively, with an average maximum difference between the real and the predicted price of 157.54. Again, we will later use these values to compare the data in a market that includes news traders. However, we can already note a slight tendency for the model to underestimate stock prices, as we observe an average negative bias.

One last interesting metric to analyze is the volatility of these predictions, since a shock in the market could potentially increase this metric. In these 40 days, we can observe an average market volatility of 0.88%, while the predictive model gives us a volatility of 1,34% (a 52% increase). We will subsequently test if these values go up relative to each other when news are added to the market.

Once again, the results in this section will serve as a baseline and will allow us to analyze the results we get from the ARIMA when we introduce a shock in the market in the next section.

5.2. Effect of good news on the ARIMA model

After gathering information on how effective the model is in "normal" conditions, we can now analyze the impact of an event that produces a shock in the stock market. We will start by analyzing the cases in which this impact is positive and then compare it with negative news. The goal is to determine if there are any relevant differences between the two. In Figure 7, one can observe some examples of the behavior of the model after we plant a news announcement midway through our testing dataset.

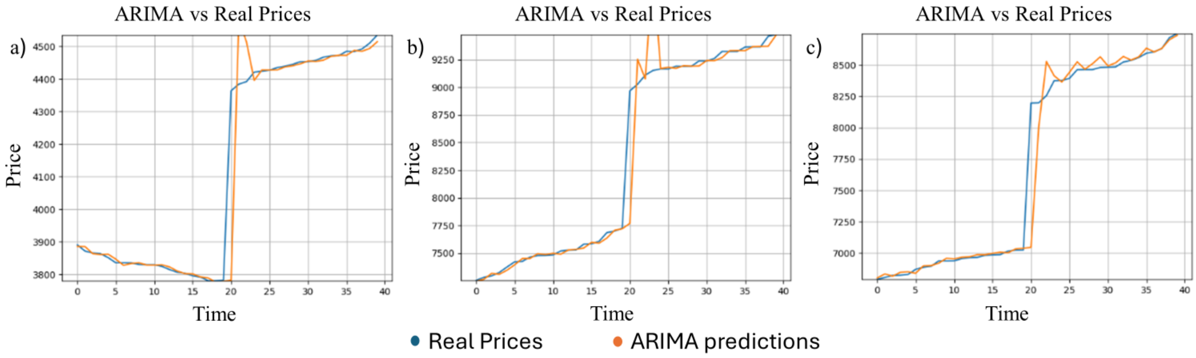

Figure 7. ARIMA *versus* Real Prices: Good News. a, b and c show three examples of the ARIMA model predictions *versus* prices generated by our ABM, when good news are introduced into the market.

By merely observing Figure 7, we can clearly see that, after the news announcement, the model seems to lose accuracy. Before the shock, the model appears to correctly predict the movement of the market. However, once news traders correct the stock price, the ARIMA model makes some predictions which are unprecedently distant from reality.

After running 100 simulations, we reach the following average values: RMSE = 167.77 (305% increase); MAE = 64.14 (155% increase); Bias = -5.51; maximum difference between prediction and reality = 865.53 (449% increase). The increase in these metrics confirms our hypothesis that the model becomes less accurate when significant shocks occur in the market. This increase shows that the predicted values are more distant to reality than in our previous analysis. Furthermore, we note that a more negative bias indicates that the model underestimates the price even more. This is expected as, on the day of the shock, the model is bound to undershoot, because the price unexpectedly goes up. This value could be even higher, but we notice that the model tends to greatly overestimate the stock price in the days that follow the shock. Regarding the maximum difference of 865.53, we expect this major difference to be on the day that the news come out, since, by looking at the graphs, we can confirm that, there is a delay in predicting this anomaly.

We also observe a major increase in the volatility of predictions, going from around 1% before the news announcement to 5,6% after. Meanwhile, the real volatility goes from 0,7% to 3,8%. This suggests that, while the predictions do become a lot more volatile, this increase is proportional to the market itself (they both increase close to 450%). Therefore, in this regard, the model seems to yield reasonable results compared to the actual market.

5.3. Effect of bad news on the ARIMA model

We previously calculated that bad news in our market create a less aggressive shock when compared to good news, so it might be relevant to analyze this case separately, as they seem to affect the market differently. In Figure 8 we observe 3 cases of the ARIMA model predicting prices in a market affected by bad news.

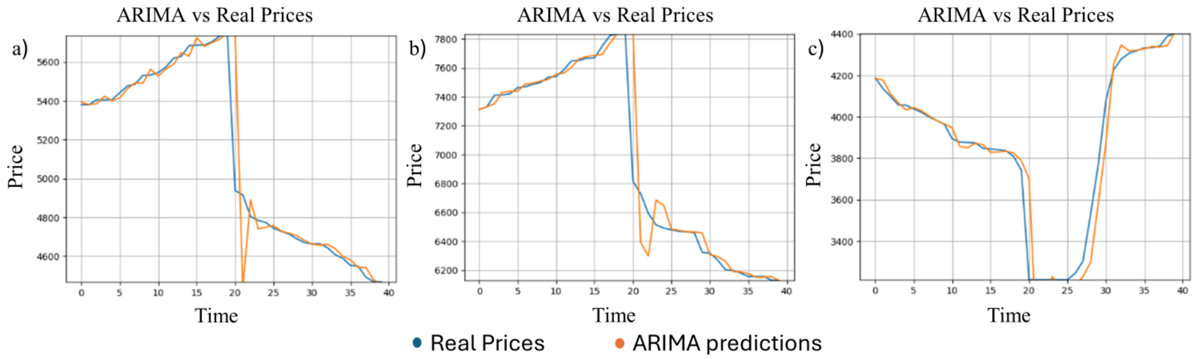

Figure 8. ARIMA *versus* Real Prices: Bad News. a, b and c show three examples of the ARIMA model predictions *versus* prices generated by our ABM, when bad news are introduced into the market.

Figure 8 shows similar results to our previous conditions; however, we still need to run precise measurements of RMSE and volatility to accurately test for differences.

Running 100 simulations as before, we obtain the following average results RMSE =144.82 (250% increase); MAE = 54.84 (118% increase); Bias = 8.2; and average maximum distance to reality = 773.75 (391% increase). This difference could be attributed to the previously mentioned less aggressive shock caused by bad news compared to good news. A bigger price change is more likely to create more inaccurate results in our model.

Previously we observed that good news have no negative effect on the model's predictions in regard to volatility. That is, volatility of predictions after the shock increased by the same factor as the real market volatility. Bad news seem to have a strange effect on the predictions' volatility, since the increase in volatility is higher in the real market than in the predictions. It increases 420% while the predictions give us a 200% increase. However, these values could be caused by a low number of iterations, since the pre-news volatility predictions are abnormally high, compared to our previous values.

In the last two sections, we confirmed our hypothesis that the ARIMA model's accuracy would be significantly affected by strong shock in the stock price. However, we still don't know how long lasting these effects are. Does the model quickly regain its accuracy? Or are there persistent side effects from this anomaly? We will attempt to answer these questions in the following sections.

5.4. How long lasting are these effects?

In this section, we analyze the persistence of the inaccuracy caused by a shock in the stock market. With this goal in mind, we will first look at our metrics before the shock. Then we will look at them again 5, 10, 15. 20 and 25 days after the shock. If we get it, we can conclude that the model has readjusted to the shock. Our measures will all be calculated using 20-day intervals, so that we obtain a fair comparison. To do so, we will need to expand our test dataset to 60 days. We will place the news announcement on day 20 of this dataset. In this chapter and the ones that follow, we will use 200 iterations, as the added accuracy is essential for clear results.

We start by analyzing the effect of good news, since they seemed to cause bigger inaccuracies in our model. Iterating 200 times, we get the following average values for the 20 days prior to the shock: RMSE = 34.27, MAE = 22.37, Bias = -1.04 and maximum distance between reality and predictions = 99.91. In the days immediately after, the following averages values were registered: RMSE = 197.47 (476% increase), MAE = 89.74 (301% increase), Bias = -13.57 and maximum distance between reality and predictions = 741.86 (643% increase). As expected, we obtain very significant increases in our metrics, as previously studied. The shock creates immediate and very substantial inaccuracies, at least in the first days after it occurs.

We then look at how the model behaves 5 days after the shock. Again, we use a 20-day time frame, which means we will be looking at days 5 to 25 after the shock. In this interval we still observe significant jumps in our metrics, indicating a decrease in accuracy by our model. However, these values represent a huge improvement when compared to the 20 days immediately after the market shock. We obtained the following average values 5 days after the announcement, with respective percentage increases compared to our baseline values: RMSE = 50.03 (46% increase), MAE = 32.54 (45% increase), Bias = 6.8 and maximum distance between reality and predictions = 144 (44% increase). Moreover, we note that our Bias metric shifted its signal (from negative to positive), indicating that all the effects on this metric are gone, or that, 5 days later, the model overcompensates for its previous negative bias.

We take this one step further and try to find evidence of the impact of the shock on the ARIMA model ten days after. Our 200 simulations show that 10 days after the news announcement the results are similar to 5 days after. The average of our metrics are as follows: RMSE = 49.36 (44% increase); MAE = 31.43 (41% increase); Bias = 4.97; Max. Distance 143.14 (43% increase). Once again, the ARIMA model is affected by the shock in the stock market, leading to inaccurate results. The Bias metric tells us that the ARIMA model continues to overestimate prices. We therefore continue to look further away from the news announcement.

Even 15 days after the shock there is around a 33% increase in the relevant metrics (excluding bias), but we are close to achieving the baseline result. To confirm this, we look 20 days after the announcement and observe that the drop in accuracy persists. Therefore, we finally look at the 25-day mark, where we still see some damage caused by the shock. However, at this stage this process was halted, as it is computationally heavy and the percentage increases are clearly slowly declining.

Table 2 summarizes the information in the previous paragraph, while Table 3 provides the values obtained from performing the same analysis with bad news.

	Before	0 days after	%	5 days after	%	10 days after	%
RMSE	34,27	197,47	476%	50,03	46%	49,36	44%
MAE	22,37	89,74	301%	32,54	45%	31,43	41%
BIAS	-1,04	-13,57		6,8		4,97	
MAX_DIFF	99,91	741,86	643%	144	44%	143,14	43%

	Before	15 days after	%	20 days after	%	25 days after	%
RMSE	34,68	46,22	35%	44,54	30%	41,4	21%
MAE	22,63	29,4	31%	28,26	26%	27,42	23%
BIAS	-0,48	3,87		1,25		0,68	
MAX_DIFF	101,91	136,19	36%	133,77	34%	110,94	11%

Table 2. Accuracy of ARIMA before and after good news. Table of relevant metrics in the days following the shock in stock price.

	Before	0 days after	%	5 days after	%	10 days after	%
RMSE	35,17	159,45	353%	40,38	15%	35,12	0%
MAE	22,93	71,47	212%	25,03	9%	22,52	-2%
BIAS	-0,78	16,96		-4,07		-1,6	
MAX_DIFF	100,56	606,17	503%	122,74	22%	101,77	1%

Table 3. Accuracy of ARIMA before and after good news. Table of relevant metrics in the days following the shock in stock price.

We observe a very significant difference in the duration of these effects when comparing good and bad news. In the case of the latter, after only 10 days, the model is reproducing very similar values to baseline, while it takes more than 25 days for this to happen in the case of good news. This is an interesting finding, given that in the previous section we saw that our metrics during the testing period were similar in both cases. However, the previous test dataset was smaller, analyzing only 20 days after the shock. With a larger test dataset, we can detect the difference in results more clearly.

We previously saw that the impact of this shock on the market's volatility is mostly gone after 2 to 5 days. As such, what could be driving the model to make such bad predictions 20 to 25 days after? As we saw earlier, the ARIMA is an auto regressive model, which means it uses its previous iterations to determine the next prediction. Consequently, the news announcement creates a chain reaction of "bad" predictions, as they all depend on the previous one. That is, if one prediction is unusually wrong, it is sure to impact the next, which in turn will impact the next one, and so on.

In the next section we investigate the effect of adding a similar shock into the training dataset, aiming to obtain improved results in our testing dataset. Training datasets are used to help the model choose the best possible parameters for each scenario. By adding a similar event to it, we expect the model to choose more adequate parameters to fit these types of market shocks.

5.5. Adding a shock to the training dataset.

The training dataset allows us to fit the model according to the usual market characteristics. That is, we use a training dataset in order to choose the most appropriate parameters (p,d,q) for our model. As such, we expect that, by adding a shock in the market to that dataset, the model chooses parameters that more accurately predict the shock in our testing dataset, as seen in Figure 9:

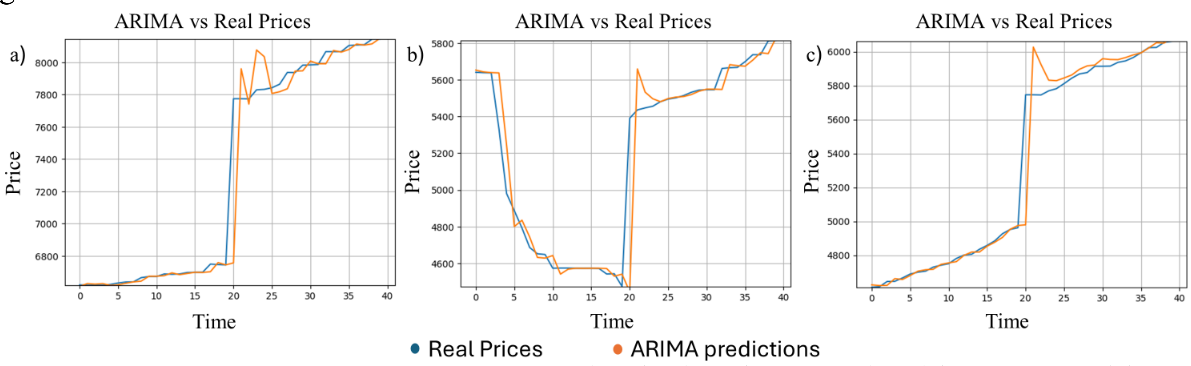

Figure 9. ARIMA with shock in training dataset. a, b and c show three examples of the ARIMA model's predictions when good news are introduced in the training and testing datasets.

Comparing these graphs with those in Figure 7, where there was no shock in the training dataset, we note that they seem similar. We therefore run a more thorough analysis to test for changes in accuracy. In this new scenario, the model averages the following metrics (with respective increase from baseline): RMSE = 164.42 (297% increase), MAE = 57.38 (128% increase), Bias = -17.14 and maximum difference = 939.07 (496% increase). These values are lower than in our previous analysis, making us hypothesize that the addition of a large shock in the training dataset may allow the model to pick more adequate parameters to help predict stock prices after a news announcement.

Following the same process for a bad news announcement we get the charts in Figure 10:

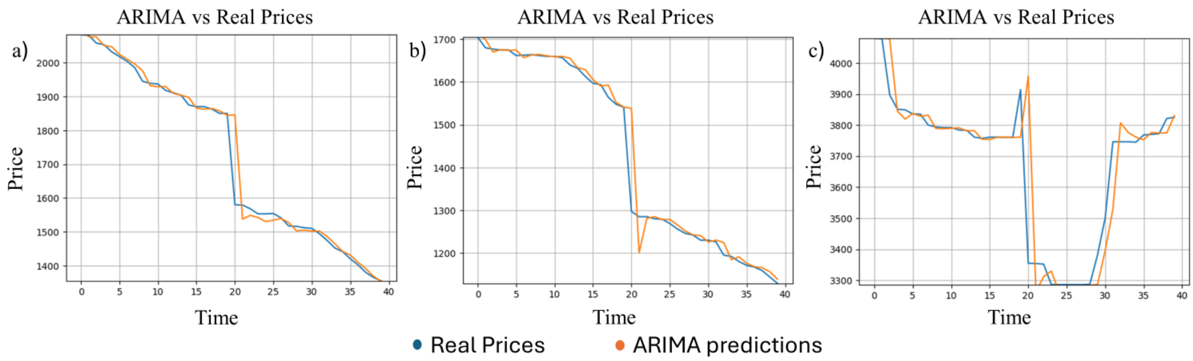

Figure 10. ARIMA with shock in training dataset. a, b and c show three examples of the ARIMA model's predictions when bad news are introduced in the training and testing datasets.

The data obtained seems to show slightly better results, leading us to run a more thorough analysis. After which we obtained the following results: RMSE = 97.39 (135% increase), 36.65 (46% increase), Bias = 6.9 and maximum distance = 557.87 (254% increase). Once again, we notice significant improvements when compared to our previous results. This supports the idea that adding a shock to the training dataset helps the model to choose better parameters for this type of event. Moreover, the values for bad news are significantly more positive than for good news, once more adding to the idea that our model is better at predicting this type of shock.

We have previously observed the benefit of looking at different time intervals after the shock to better understand and confirm our analysis. Therefore, we will do the same in the next section, to draw stronger and more interesting conclusions about the ARIMA model.

5.6. Are the effects shorter lasting?

We ran an analysis similar to the one in section 5.4. As summarized in Table 4, adding a shock to the train dataset set seems to yield significantly better results with the ARIMA model. After only 5 days, the results are better than at almost any interval of our previous analysis (except 25 days after). The values of the relevant metrics maintain a slow reversion to the pre-news value and after 25 days our RMSE is only 14% higher than baseline.

	Before	0 days after	%	5 days after	%	10 days after	%
RMSE	45,28	207,76	359%	55,97	24%	58,56	29%
MAE	27,88	82,29	195%	34,4	23%	35,08	26%
BIAS	-1,12	-39,55		-2,6		-0,67	
MAX_DIFF	133,88	860,32	543%	163,94	22%	175,66	31%

	Before	15 days after	%	20 days after	%	25 days after	%
RMSE	45,28	55,9	23%	53,59	18%	51,75	14%
MAE	27,88	32,89	18%	32,04	15%	31,81	14%
BIAS	-1,12	-0,61		-3,03		-2,7	
MAX_DIFF	133,88	172,1	29%	163,71	22%	143,22	7%

Table 4. Accuracy of ARIMA before and after good news, with shock in the training dataset. Table of relevant metrics in the days following the shock in stock price.

Regarding bad news, the data summarized in Table 5 shows significantly more positive results than before, with all traces of the shock being eliminated after only 5 days. Additionally, in the first 5 days after the shock, we can observe that our metrics suffer a much less intensive shock than before, confirming the success of adding one of these anomalies to the train dataset.

	Before	0 days after	%	5 days after	%
RMSE	30,64	118,31	286%	30,71	0%
MAE	20,48	50,7	148%	20,99	2%
BIAS	-1,78	13,43		-4,28	
MAX_DIFF	88,76	488,72	451%	82,99	-7%

Table 5. Accuracy of ARIMA before and after bad news, with shock in the training dataset. Table of relevant metrics in the days following the shock in stock price.

6. Conclusions

After analyzing all these results, we reach some interesting conclusions regarding the ARIMA model through the use of our ABM. Firstly, we conclude that, as expected, the model becomes very inaccurate in the days immediately after our simulated news announcement, yielding an average increase of around 400% in some of our metrics in the days that follow. Secondly, we observe that, in our market, good news have a much stronger impact on the accuracy of the model, leading to accentuated and long-lasting poor predictions. On the one hand, the impact of bad news is negligible after just a few days (around 5 to 10). On the other hand, good news create an impact that can be seen even upwards of 25 days after the market shock. If we were to use this model to predict prices in our day-to-day, we would be significantly more impacted by positive shocks in the market. However, this could be partially caused by how our market works, or the types of traders involved, since this difference is very unexpected and rather disproportionate.

Additionally, we find some interesting results when we add a similar event to the training dataset. When this shock is added, the model seems to perform better than before, especially in the case of bad news. In this case, the effects of the second shock (the one in the testing dataset), seem to wear off after only 5 days, which is a clear improvement over our previous values of 10 days. Regarding good news, the improvement is less significant but still relevant, yielding lower percentage increases in our metrics.

With this in mind, we can hypothesize that in the real world, the ARIMA model could yield acceptable results after a similar shock, since these types of announcements are more frequent. Therefore, we expect the model to pick the most adequate parameters, as the training data set in question has many instances of this type of shock. According to our analysis, this would increase the model's accuracy in such a scenario.

7. Limitations and Future Research

This study has several limitations, which inhibit us from drawing more significant and relevant conclusions about the ARIMA model. Some of them arise from the characteristics of our market, and others by the nature of the study itself.

Firstly, we only use two types of traders (besides the news traders) and, even though they can create significant movement in our market, the volatility and the type of movement are far from our real-life market. Furthermore, the choice and implementation of these traders is rather arbitrary and strictly with the goal of mimicking the real-life financial market. Nevertheless, we are able to draw some significant conclusions on how the model behaves once we establish a baseline, despite the fact that results can often be far from reality.

Additionally, this type of study requires a lot of computational power for fitting the ARIMA model and using it to predict values. We were often limited to around 200 iterations to get our mean results, which is not ideal to draw strong conclusions. This could have forced us to speculate a lot on the meaning of these results and how they compare with one another.

Regarding future research, I consider that there should be an attempt to better mimic the market and run more efficient simulations of this market. Additionally, more shocks of this type could be added, to analyze the progress of the model's accuracy. Is there a limit to how many shocks we can add until the model becomes futile? We could take a deeper dive into these sorts of questions in future research.

Finally, we could analyze the impact of less aggressive shocks, this would help us get a better understanding of the model's capabilities in a real-life scenario, where shocks of 15% don't happen often. Experimenting with different shock magnitudes would also allow us to better compare the impact of good and bad news. The differences we observed between them in this study could potentially be linked to the shock not being the same magnitude.

References

Ahmed, Mirza Faizan. 2019. “Estimating Proportion of Noise Traders and Asset Prices.” *Business Review* 14 (2): 1–12. https://doi.org/10.54784/1990-6587.1014.

Alanyali, Merve, Helen Susannah Moat, and Tobias Preis. 2013. “Quantifying the Relationship Between Financial News and the Stock Market.” *Scientific Reports* 3 (1): 3578. https://doi.org/10.1038/srep03578.

Ariyo, Adebiyi A., Adewumi O. Adewumi, and Charles K. Ayo. 2014. “Stock Price Prediction Using the ARIMA Model.” In *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*, 106–12. IEEE. https://doi.org/10.1109/UKSim.2014.67.

Asness, Clifford S., Tobias J. Moskowitz, and Lasse Heje Pedersen. 2013. “Value and Momentum Everywhere.” *The Journal of Finance* 68 (3): 929–85. https://doi.org/10.1111/jofi.12021.

Carhart, Mark M. 1997. “On Persistence in Mutual Fund Performance.” *The Journal of Finance* 52 (1): 57–82. https://doi.org/10.1111/j.1540-6261.1997.tb03808.x.

Cohen, Kalman J., Steven F. Maier, Robert A. Schwartz, and David K. Whitcomb. 1983. “A Simulation Model of Stock Exchange Trading.” *SIMULATION* 41 (5): 181–91. https://doi.org/10.1177/003754978304100502.

Contreras, J., R. Espinola, F.J. Nogales, and A.J. Conejo. 2003. “ARIMA Models to Predict Next-Day Electricity Prices.” *IEEE Transactions on Power Systems* 18 (3): 1014–20. https://doi.org/10.1109/TPWRS.2002.804943.

Foltice, Bryan, and Thomas Langer. 2015. “Profitable Momentum Trading Strategies for Individual Investors.” *Financial Markets and Portfolio Management* 29 (2): 85–113. https://doi.org/10.1007/s11408-015-0246-4.

Garman, Mark B. 1976. “Market Microstructure.” *Journal of Financial Economics* 3 (3): 257–75. https://doi.org/10.1016/0304-405X(76)90006-4.

Gode, Dhananjay K., and Shyam Sunder. 1993. “Allocative Efficiency of Markets with Zero-Intelligence Traders: Market as a Partial Substitute for Individual Rationality.” *Journal of Political Economy* 101 (1): 119–37. https://doi.org/10.1086/261868.

Gorton, Gary B., and James Dow. 2006. “Noise Traders.” *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.901857.

Grossman, Sanford. 1976. “On the Efficiency of Competitive Stock Markets Where Trades Have Diverse Information.” *The Journal of Finance* 31 (2): 573. https://doi.org/10.2307/2326627.

Grossman, Sanford J. 1977. “The Existence of Futures Markets, Noisy Rational Expectations and Informational Externalities.” *The Review of Economic Studies* 44 (3): 431–49. https://doi.org/10.2307/2296900.

Hakansson, Nils H., Avraham Beja, and Jivendra Kale. 1985. “On the Feasibility of Automated Market Making by a Programmed Specialist.” *The Journal of Finance* 40 (1): 1–20. https://doi.org/10.2307/2328045.

Jamal, Karim, and Shyam Sunder. 1996. “Bayesian Equilibrium in Double Auctions Populated by Biased Heuristic Traders.” *Journal of Economic Behavior & Organization* 31 (2): 273–91. https://doi.org/10.1016/S0167-2681(96)00892-X.

Jegadeesh, Narasimhan, and Sheridan Titman. 1993. “Returns to Buying Winners and Selling Losers: Implications for Stock Market Efficiency.” *The Journal of Finance* 48 (1): 65–91. https://doi.org/10.2307/2328882.

Kanaskar, Shikhar. 2023. “Predicting Stock Prices: Using ARIMA, Fourier Transformation and Deep Learning.” Shikhar’s Data Science Projects. March 12, 2023. https://medium.com/shikhars-data-science-projects/predicting-stock-prices-using-arima-fourier-transformation-and-deep-learning-e5fb4f693c85.

Kyle, Albert S. 1985. “Continuous Auctions and Insider Trading.” *Econometrica* 53 (6): 1315–35. https://doi.org/10.2307/1913210.

LeBaron, Blake. 2000. “Agent-Based Computational Finance: Suggested Readings and Early Research.” *Journal of Economic Dynamics and Control* 24 (5–7): 679–702. https://doi.org/10.1016/S0165-1889(99)00022-6.

Lee, Cheng-Ming, and Chia-Nan Ko. 2011. “Short-Term Load Forecasting Using Lifting Scheme and ARIMA Models.” *Expert Systems with Applications* 38 (5): 5902–11. https://doi.org/10.1016/j.eswa.2010.11.033.

Merh, Nitin, Vinod Saxena, and Kamal Pardasani. 2010. “A Comparison between Hybrid Approaches of ANN and ARIMA for Indian Stock Trend Forecasting.” *Business Intelligence Journal* 3:23–44.

Nau, Robert. 2020. “Introduction to ARIMA: Nonseasonal Models.” Duke University. August 23, 2020. https://people.duke.edu/~rnau/411arim.htm.

Poggio, Tomaso, Andrew W. Lo, Blake D. LeBaron, and Nicholas T. Chan. 2001. “Agent-Based Models of Financial Markets: A Comparison with Experimental Markets.” *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.290140.

Appendix

Root Mean Squared Error (RMSE):

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2} \tag{7}$$

Mean Absolute Error (MAE):

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |\hat{y}_i - y_i| \tag{8}$$

Bias:

$$Bias = \frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i) \tag{9}$$