University of Hertfordshire UH

School of Physics,
Engineering and
Computer Science

MSc Data Science Project 7PAM2002-0509-2024

Department of Physics, Astronomy and Mathematics

Data Science FINAL PROJECT REPORT

Project Title:

Time Series Forecasting of Stock Data

How effective is ARIMA for modelling stock prices, and what trends can be identified from its fitted components and how can these be compared to other machine learning models?

Student Name and SRN:

Ethan Deery, 19031484

GitHub Repository:

https://github.com/edeery3/Stock_prediction/tree/main

Supervisor: Robert Yates

Date Submitted: Enter the date you are submitting this report

Word Count: 5139

DECLARATION STATEMENT

This report is submitted in partial fulfilment of the requirement for the degree of Master of Science **in Data Science** at the University of Hertfordshire.

I have read the detailed guidance to students on academic integrity, misconduct and plagiarism information at Assessment Offences and Academic Misconduct and understand the University process of dealing with suspected cases of academic misconduct and the possible penalties, which could include failing the project or course.

I certify that the work submitted is my own and that any material derived or quoted from published or unpublished work of other persons has been duly acknowledged. (Ref. UPR AS/C/6.1, section 7 and UPR AS/C/5, section 3.6)

I did not use human participants in my MSc Project.

I hereby give permission for the report to be made available on module websites provided the source is acknowledged.

Student Name printed: Ethan Deery

Student Name signature Ethan Deery

Student SRN number: 19031484

UNIVERSITY OF HERTFORDSHIRE

SCHOOL OF PHYSICS, ENGINEERING AND COMPUTER SCIENCE

Abstract
This project evaluates ARIMA, rolling ARIMA, and LSTM models for forecasting FTSE 100 stock prices, including GSK, BAE Systems, and EasyJet, using daily closing data from 2015–2025. Exploratory analysis highlighted differences in stationarity and trends, guiding model selection. Forecasts were assessed using RMSE, AIC, and Ljung-Box statistics. Results indicate ARIMA performs well for short-term prediction of stationary stocks, while LSTM can capture trends in non-stationary stocks but requires optimisation. Rolling ARIMA improved accuracy but introduced lag. The findings highlight the importance of matching forecasting methods to data characteristics and suggest hybrid approaches and external factors for future research.

Introduction

Time series analysis is the study of data observed at regular intervals to identify patterns, uncover trends, and generate forecasts of future outcomes. Applications are wide-ranging, including climate modelling, healthcare monitoring, and anomaly detection. This project focuses specifically on financial forecasting, with the aim of predicting the behaviour of stocks and shares. Accurate forecasts can provide insights into how financial variables evolve over time, highlight anomalies, and support informed decisions on risk assessment and investment strategies.

Financial forecasting offers multiple benefits and can be applied in both long term and short term, to generate potential returns. It can also be useful for predicting how a market may behave to potential shocks and changes, making a greater stability in a volatile environment. However, forecasting financial data is inherently challenging due to volatility, noise, and non-stationarity. These complexities motivate the exploration and comparison of different predictive approaches. These insights can help benefit investors to optimise investments and time trades and companies make more informed decisions when planning.

The stocks focused on in this project come from the FTSE 100 which represents the 100 biggest companies in the UK market. The three companies come from different sectors: GSK (pharmaceuticals), BAE Systems (defence), and EasyJet (aviation). These firms were selected to represent distinct sectors with varying levels of volatility, sensitivity to global events, and long-term growth prospect. GSK represents a relatively stable, defensive stock from healthcare whereas BAE Systems reflects the influence of geopolitical factors on defence industry performance. By examining these diverse companies, the study not only evaluates forecasting accuracy but also investigates how model performance may vary across industries with different risk and return profiles.

The research question addressed in this study focuses on evaluating the performance of the ARIMA (Autoregressive Integrated Moving Average) model for both short-term and long-term stock price predictions. ARIMA has historically been one of the most widely used approaches to financial forecasting due to its simplicity, interpretability, and ability to capture linear trends. However, its effectiveness is limited when dealing with highly volatile, non-linear data such as stock prices. In contrast, machine learning models such as Random Forests and Neural Networks have become increasingly common, offering improved predictive performance but at the cost of greater complexity, reduced interpretability, and higher data requirements. This project therefore aims to assess whether ARIMA remains a competitive and practical forecasting tool compared to these newer approaches.

Literature review

Various approaches have been developed to optimise stock market prediction, each with different objectives. The methods vary in success rates and usability and have developed over time.
ARIMA (Autoregressive Integrated Moving Average) is widely used for forecasting due to its simplicity and acceptability of the model. Mondel, (2014) used ARIMA as a method to examine data from 56 companies, using 23 months of data to predict the outcome of the next month. The focus was on smaller ARIMA values, aided by AIC (Akaike information criterion) values to determine the best model. Mean absolute error (MAE) was used to measure success and achieved an 85% accuracy over the different sectors.

Auto ARIMA is a variation that has been used to automate the process of selecting the optimal ARIMA model. Khan, (2020) used this method however also benefitted from trialling several different models for optimisation. In this case, the auto ARIMA underperformed the trialled model showing there are benefits in following this process.

Whilst ARIMA represents a strong level for stock market prediction, non-linearity is not handled well which motivates researchers to explore other methods within machine learning.

Random forests have been used to predict stock movement and can focus more on technical indicators within the stock market Khaidem, (2016). Measurements in this case were a binary classification of whether the stock increased or decreased achieved. A 94% accuracy on one of the stocks showing it is robust at predicting these changes. However, this model benefited from a look ahead bias leading to an unfair advantage.

Neural networks have also been used, more specifically LSTM (Long Short-term memory) to predict future stock prices due to their ability to model sequential data and nonlinear data. Roondiwala, (2017) used this method on 5 years' worth of NIFTY 50 stocks. Key parts involved preprocessing of data such as transforming so models could be built as well as using strong feature extraction. 2 LSTM layers were used, a singular dense layer and linear output. Overall, a low RMSE value was found showing its strength as a model. This model is stronger at dealing with non-linearity within the data but does come with a greater computational cost and a risk of overfitting.

Pre-built models have also been created and one of the biggest is Facebook's Prophet model. Prophet considers yearly, monthly, and weekly trends and therefore is strong model when there are clear seasonal patterns (Pant, 2024). This research also showed that Prophet improved nonlinearity handling based upon a years' worth of Indian stock data. Overall, an RMSE value of 15.97, demonstrating strong predictive performance in data with seasonal patterns.

To combine the benefits of these models, we can create hybrid models to help enhance the predictions made and the robustness of models built. Choi, (2018) proposed using a combination of an ARIMA and LSTM models. The research was completed on 9 years' worth of stock data and a 100-day time window was used for the LSTM. The process first used the ARIMA to help filter out linearity where these inputs were fed into an LSTM. Overall, this model outperformed the individual counterparts showing this combination was beneficial on their data as ARIMA captured linear trends while LSTM modelled nonlinear residuals.

Despite these advances, comparisons between studies are challenging due to inconsistent evaluation metrics. This underscores the importance of careful experimental design and the selection of appropriate performance measures when developing forecasting models.

Dataset

The dataset was obtained from Investing.com (2025), a website providing data of prices from numerous stocks and shares. The data was taken for the period 1st April 2015 to 1st April 2025 at daily intervals. The values consist of an open, close, max and min price plus the volume of shares traded that day. As stock prices fluctuate frequently, the closing price was selected as the target variable.

The stocks collected are companies within the FTSE 100. I decided to collect a range of different industry classifications to compare how models may generalise certain stocks better or whether models predict more effectively in different areas, which may have various levels of volatility. The main shares I focused on were GSK, BAE Systems and EasyJet as these are a range of different stocks and well renowned within their industry.

This dataset is ideal for my research due to its simplicity and accessibility of the data. This would allow any models I build to be easily transferred and used on new data for future projects.

Ownership of the data is a key consideration as online sites can use second hand data meaning the rights aren't completely theirs. This means referencing needs to be done correctly to highlight the correct source and used in the correct manner. Another key point is that analysis must be completed in a safe way and therefore avoiding misleading others into false information or bias within the data. In a project such as this, future readers must also know how to access the data used within the project.

Exploratory data analysis

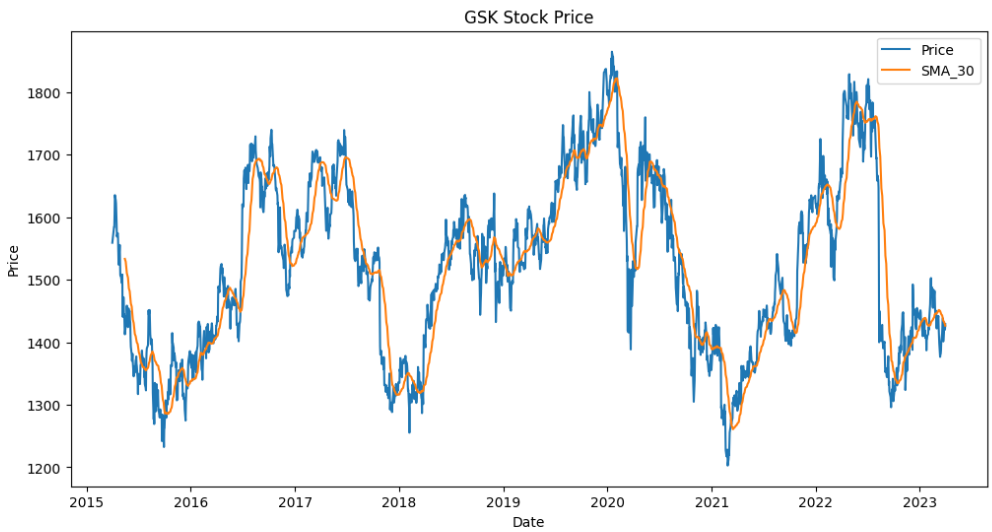

Figure 1 – GSK Price

Figure 1 shows the plot for the GSK closing price. The raw price data shows the clear short-term fluctuations that are typical in the stock market. The plot also involves a simple 30-day moving average, which is useful to understand the longer trends and highlight the noise in the individual data points.

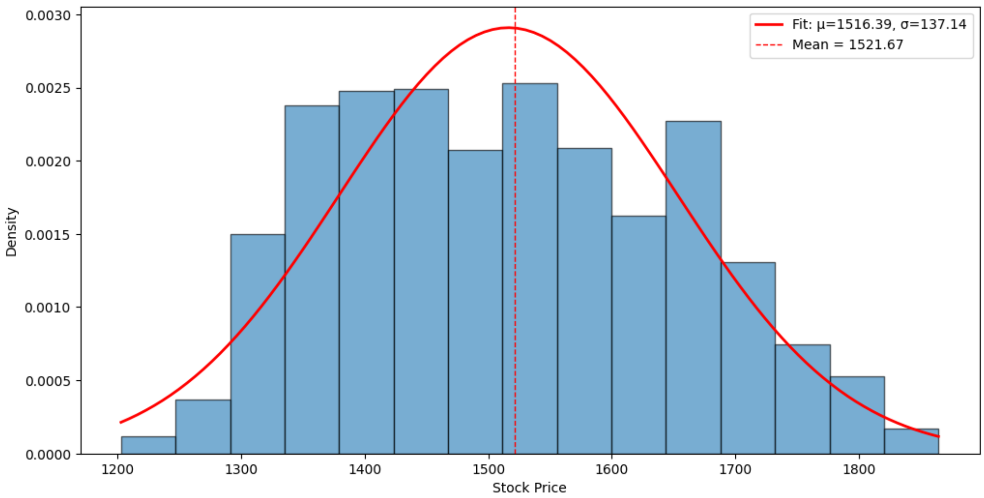

Figure 2 – GSK Histogram

Figure 2 shows the distance of the GSK price data, which appears to follow a bell-shaped curve such as the fitted gaussian distribution, where the majority of the datapoints fall between 1300 – 1700 with some extremes. This is highlighted by a mean of 1521.67 and standard deviation of 140.94. This shows whilst there is volatility, the values tend to remain close to the mean over time, as we can see shown in figure 1.

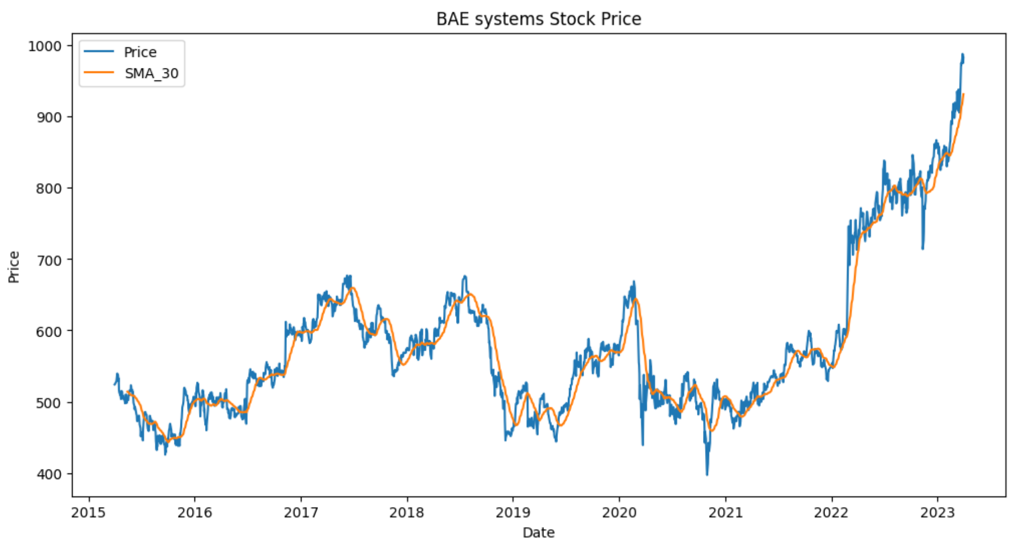

Figure 3 – BAE Price

Figure 3 shows a plot of the BAE Systems closing price and how this changes. This dataset differs to the GSK data as there is a clear long term upward trend. The moving average once again smooths the value of the stock, allowing the overall trajectory to be observed more easily.

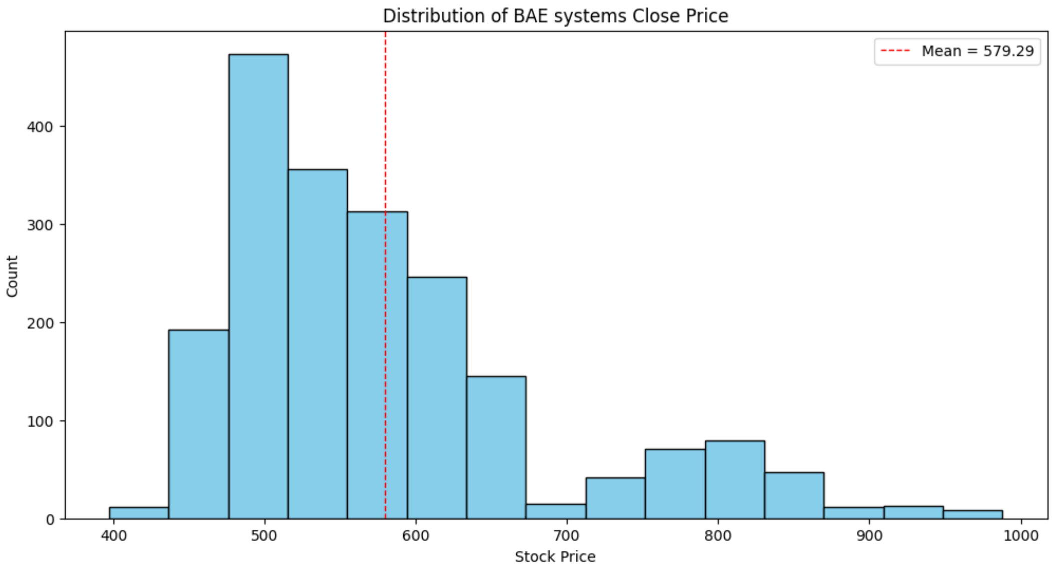

Figure 4 – BAE histogram

Figure 4 shows the distribution of the BAE Systems data, and it is clearly positively skewed, due to the early prices being much lower. The mean closing price of 579.29 lies to the right of the main peak of the distribution, further reflecting the upward trend in the series. This asymmetry in the distribution highlights the presence of long-term growth rather than reversion to a stable average, distinguishing BAE from GSK. This skew could highlight a need for differencing.

Methodology

Firstly, I pre-processed the data so that my models would be able to work effectively and correctly. This involved converting values to integers rather than strings, examining null values and interpolating the data to fill in gaps shown by non-working days. This was performed on the GSK and BAE datasets.
I split the respective datasets into a training and testing dataset which consisted of the first 8 years (01/04/2015 – 01/04/2023) being used to train my model to predict the following 2 years.

ARIMA

ARIMA is made up of 3 different terms, Auto Regressive (AR) Integrated (Differencing) and Moving Average (MA) which are used in different calculations and are often displayed as (p, d, q). AR represents the lag of the prior p terms which implies that what has happened recently will continue to happen for the next term. Differencing represents the order of differencing, d. This value is used to make the data stationary as ARIMA requires a constant mean and standard deviation to maintain a valid relationship over time. This is commonly done by subtracting the current term from the prior term in the equation:

$$z_i = y_i - y_{i-1}$$

This equation highlights 1st order differencing, which can be extended for 2nd order. The final term MA accounts for the errors made in the previous q timepoints. This helps to make the prediction smoother and less reactionary. If we underpredict, the value at the next point will be corrected upwards and vice versa.

Previous research has created ideas to help select which values offer the best models based upon training data. One way is using autocorrelation and partial autocorrelation plots (ACF and PACF). If the ACF declines gradually, we expect the series to be autoregressive with the order shown on the lag of the PACF (Idrees, 2019). This reverses for the MA term where the PACF will decline gradually with the order shown on the lag of the ACF. These plots can also be used to deduce the differencing term, if the ACF or PACF has a slow decline (Dong, 2017).

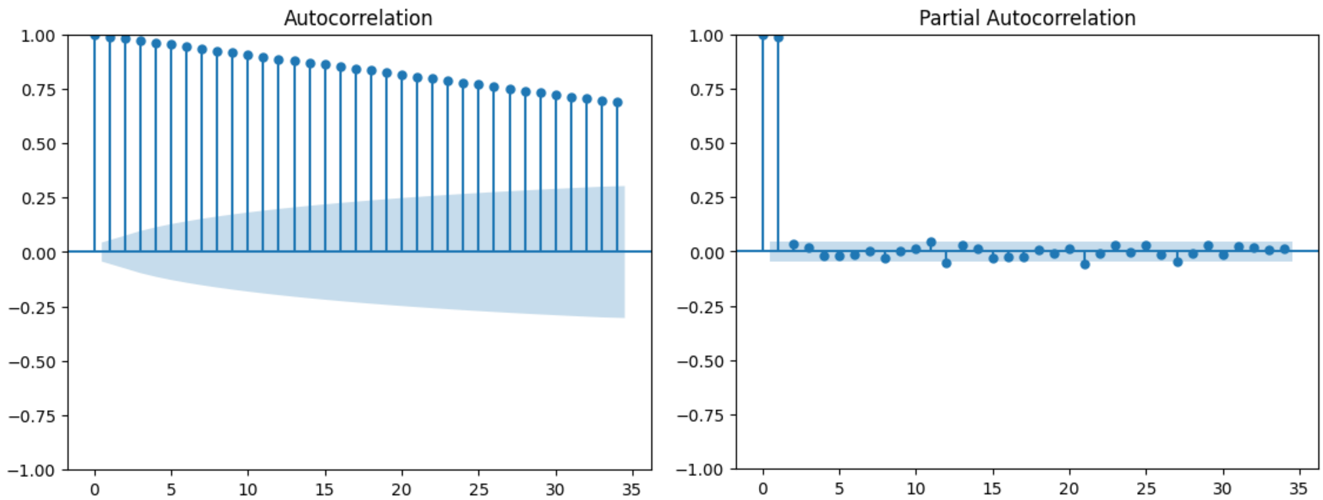

Figure 5 – ACF and PACF plots for GSK data

Figure 5 shows these plots for the GSK data. We can deduce that this should require a 2nd order autoregressive model based upon the rules above. However, the ACF has a gradual decline so it may mean differencing is required. Another test we can use to deduce if the series is stationary is the augmented Dickey Fuller test (ADF test). This statistical test gives us a value and if the p value is greater than 0.05, further differencing is required (Analytics Vidhya, 2025). For our data, the Adfuller test gave a value of -3.0220 and a p value of 0.0329 meaning differencing is probably not needed.

We can use a visual experimentation to deduce whether differencing is needed as data may have a clear trend. This can be seen below as we compare the GSK and BAE data and see how the BAE in figure 3 has constant increase whereas the GSK data in figure 1 fluctuates far more often and appears to be distributed normally around a mean.

While ARIMA provides a strong statistical baseline, its assumptions of linearity and stationarity limit performance in more complex datasets. To address this, I also implemented rolling ARIMA and neural network approaches.

Rolling ARIMA

ARIMA models struggle when handling nonlinearity, which may lead to important relationships being missed. One way we can manage this is by using rolling forecasting which works by rather than training data in a fixed dataset, the training window is continuously updated as time progresses. Therefore, the most recent data points are incorporated allowing complex relationships to be learnt and more adaptive forecasts. This continuous updating allows the model to adapt to structural changes in the data such as shifts in trends or seasonal patterns.

This method does come with limitations. The time that can be predicted in advance is limited to the time step given. For example, I chose a model which is retrained every 5 days to align with the start of a week. This prediction is only valid for that brief period, making it less beneficial in long term forecasting. The method also has greater computational cost due to more ARIMA models being built.

LSTM

As part of the project, I wanted to be able to compare how ARIMA stacks up compared to another forecasting method such as LSTM. LSTMs are a type of recurrent neural network designed to learn from sequential data. LSTMs help with handling the long-term dependency problem (Olah, 2015) which can be experienced by ARIMA. LSTM add and remove information using three different memory cells:

- Forget gate – this determines the information that gets thrown away and not used in future predictions.
- Input gate – made up of two separate nodes – one which decides which values will be updated and the next which creates new candidate values to use in the model.

- Output gate – this helps to filter how much of the stored information is passed on.

I performed extra preprocessing to enable this model to be used. Firstly, the use of a MinMax scaler to scale the values between 0 and 1 to improve the gradient stability in backpropagation. I also selected a look back window of 100 days to be used to predict the following days, based upon experimentation. LSTMs can also be for long-term and short-term forecasting and therefore can be compared in both these ways with the overall ARIMA models.

```
model = Sequential([
    Input(shape=(X_train.shape[1], X_train.shape[2])),
    LSTM(128, return_sequences=True),
    Dropout(0.2),
    LSTM(64, return_sequences=True),
    Dropout(0.2),
    LSTM(32),
    Dropout(0.2),
    Dense(1)
])

model.compile(optimizer='adam', loss='mean_squared_error')
```

Figure 6 – LSTM code

Above shows the architecture of the LSTM models I used. There are 3 LSTM layers with varying units to capture some complexity. I have also used a dropout layer of 0.2 to help reduce the risk of overfitting and finally a dense layer to give the output value. These decisions were made through experimentation to build the best model. The model was then compiled using the adam optimiser and the mean square error loss function to measure training success.

I trained the model for up to 20 epochs with a batch size of 32. I also implemented an early stopping callback to monitor validation loss and stop training if improvements were not made for 5 epochs to maintain the best weights.

Results

Model evaluation metrics

We can use numerous metrics to examine whether the model was successful or. The first of these is an RMSE (Root Mean squared error). This value is used to measure the difference between our predicted values and the real values.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(X_i - \hat{x}_i)^2}$$

The equation above shows how we calculate this by taking the difference and squaring this value at each timepoint, finding the mean of these values, and taking the square root of this value. The lower the error value, the closer the predictions will be to the real data.

We can also use an AIC (Akaike information criterion) which is a statistical value to measure the overall fit and complexity of an ARIMA model.

$$AIC = 2k - 2\ln(L)$$

AIC is calculated as in the equation above where k is the number of parameters and L represents the maximised value of the likelihood of the model. This value punishes models for overfitting and a high number of parameters. This value ideally should be as low as possible to represent a good fit.

Another test I used was the Ljung-Box score which measured whether residuals were autocorrelated. This is important to show to make sure we are gathering all relevant information in the model.

GSK results

I first evaluated the (2,0,0) devised by the rules in my methodology. One way we can examine the quality of the model is the strength of the forecast, which can be seen in the plot below.

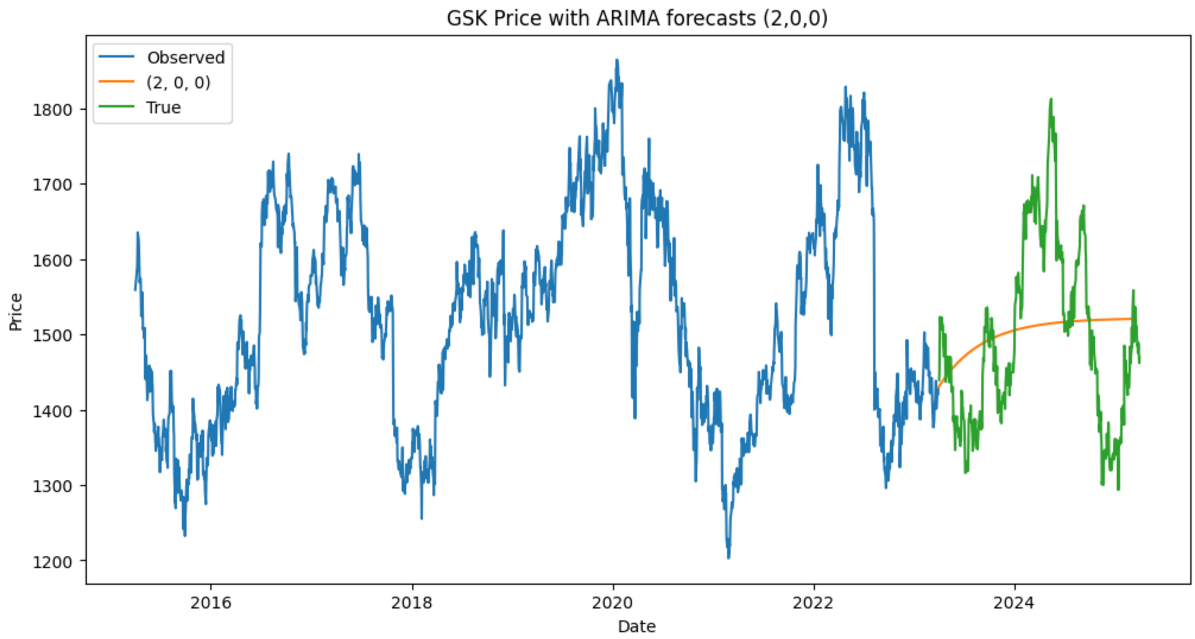

Figure 7 – (2,0,0) GSK ARIMA

Overall, the model is very stale, where the prediction converges to near a mean value. This can be a drawback of ARIMA as it can fail long term to observe these complex relationships on noisy data. However, as a short-term prediction the first few timepoints show a strong prediction.

Model Order	AIC	RMSE of Predictions	Ljungbox score
(2,0,0)	24261.65	114.2	0.866
(3,0,5)	24271.59	113.8	0.969
(2,1,2)	24248.37	135.73	0.099
(0,0,2)	31006.61	121.21	0
(2,2,0)	25087.37	1198.86	0

Table 1 – GSK Model Metrics

Using code I tested, values of up to 5 for AR and MA values and 2 differencing were tested and table 1 shows an example of some of the stronger and weaker models. Firstly, we can see how the high differencing model (2,2,0) and a moving average model (0,0,2) perform worse showing there is no reason to explore these further. Therefore, we can compare the remaining 3 models.

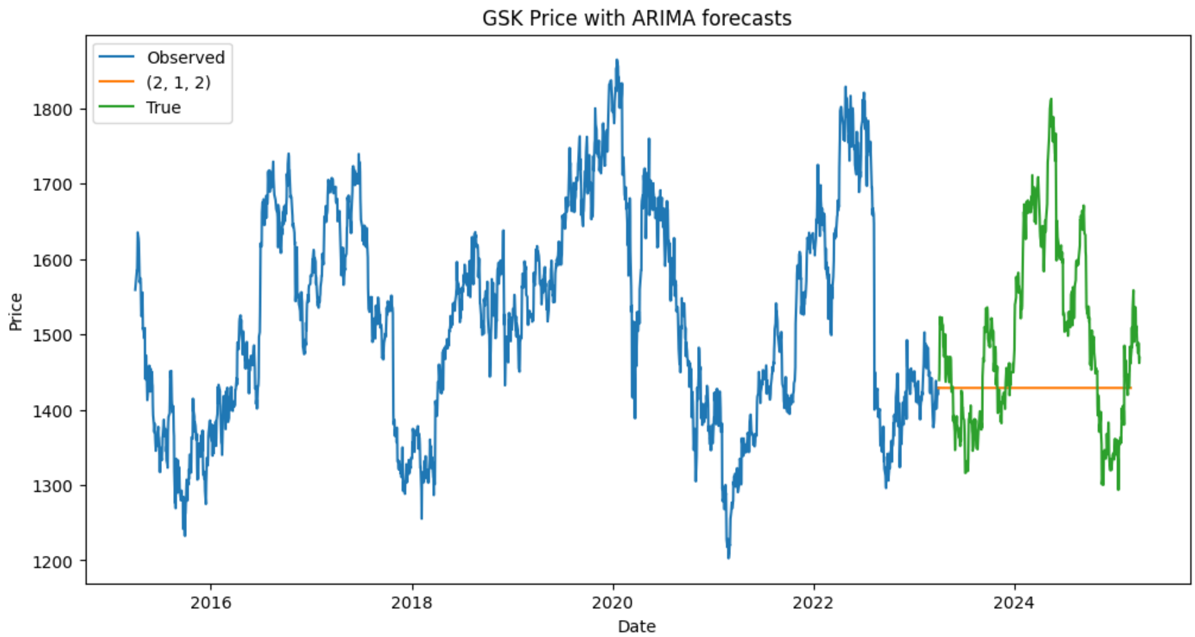

Figure 8 – (2,1,2) GSK ARIMA

Figure 8 show the plot using the (2,1,2) model which offered the best AIC value. The forecast predicts the same value for the entirety of the forecast and the main issue is caused by differencing as the data is already stationary. The Ljung-Box value also suggest that the residuals are correlated meaning this is a poor model.

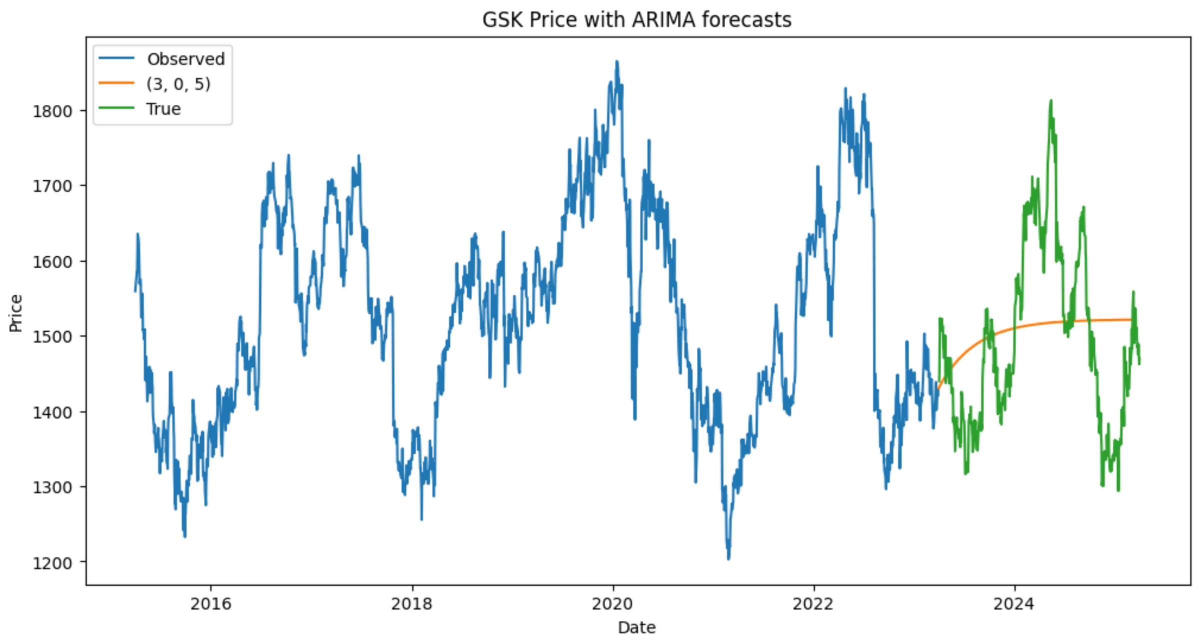

Figure 9 – (3,0,5) GSK ARIMA

Finally, we can see the (3,0,5) which shows a similar forecast to our original model. The RMSE is only marginally better suggesting that we can use either model. Due to the better AIC model, I felt the (2,0,0) model was stronger due to simplicity.

One way this model can be used is when we may look to invest in a stock. One potential application of such forecasts is to establish trading thresholds, where values below the forecast line may indicate a buy opportunity, and values above may indicate a sell trigger. This comes with its own difficulties but as long-term investment, the data follows a noisy pattern fluctuating above and below its mean.

BAE results

I examined further how this method would work on the BAE Systems dataset, as there were signs of trend and need for differencing. By using the rules for the ACF plots and Ad fuller test, a (2,1,0) model is advised.

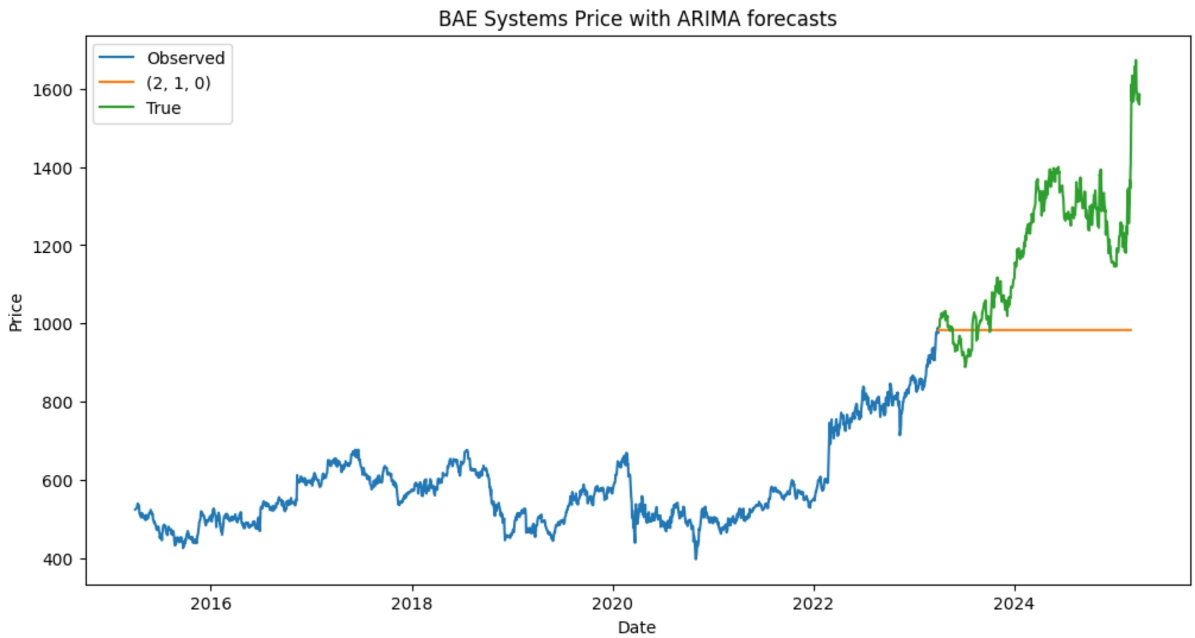

Figure 10 – (2,1,0) BAE ARIMA

This plot shows this model and its limitations. Although a first-order difference was applied, the model still failed to capture the upward trend, suggesting that higher-order differencing may be required. This does come with risk as research states that values above one are rarely used due to over differencing issues (Hyndman, 2008).

Model Order	AIC	RMSE of Predictions	Ljungbox score
(5,2,0)	19742.14	116.01	0
(3,1,2)	19305.20	268.26	0.99
(2,1,0)	19311.83	268.29	0.99
(5,0,0)	19325.85	342.22	0.88

Table 2 – BAE model metrics

The (5,2,0) model with the highest differencing has the lowest RMSE value. However, when examining the model, we need to be wary of the fact the residual may be correlated due to the Ljungbox score.

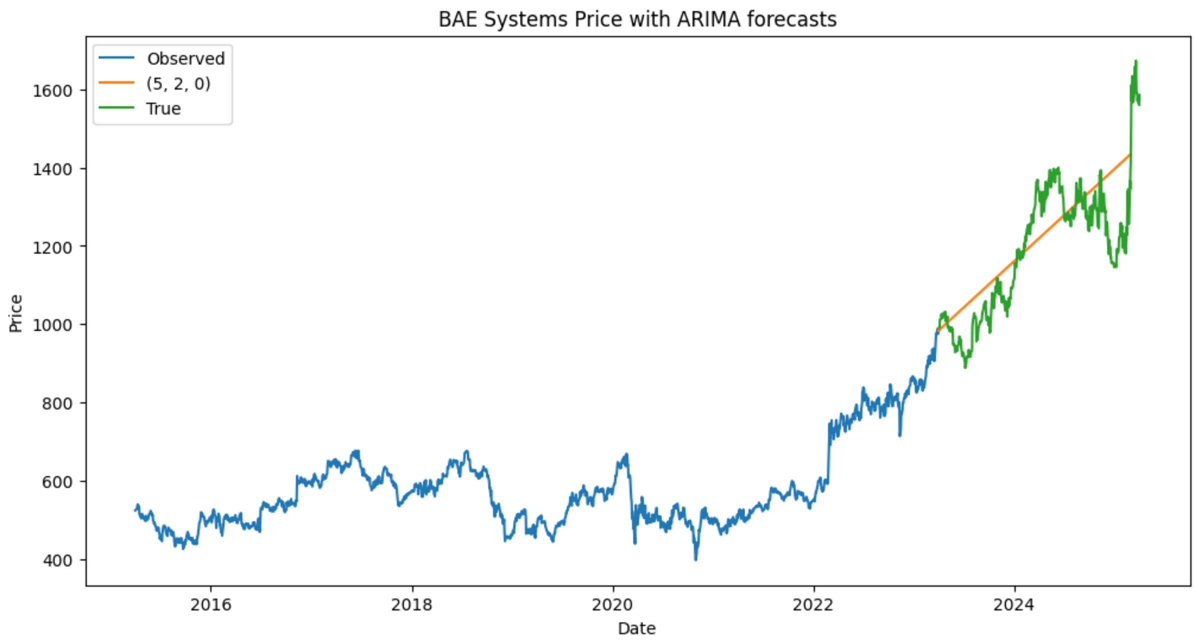

Figure 11 – (5,2,0) BAE ARIMA

The forecast for the (5,2,0) model appears to follow the trend being shown by the data strongly. This is a bold prediction, however in this we can see how it works effectively in this case due to the severity of increase in the data. In comparison to the GSK dataset, this model does show a good long-term prediction of the BAE stock price.

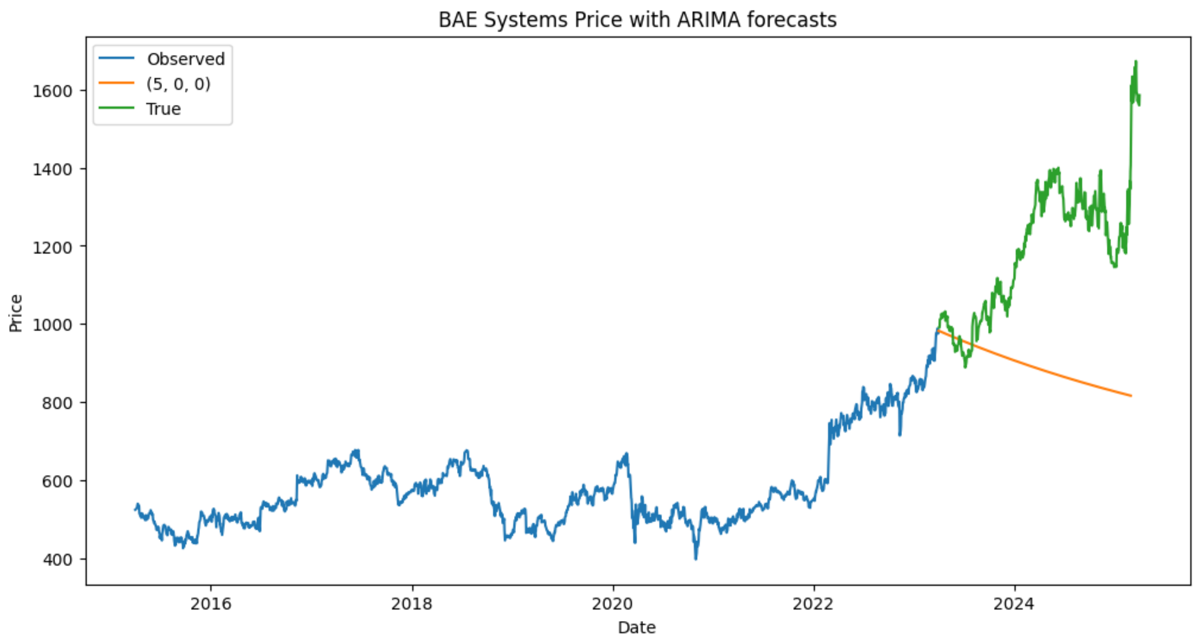

Figure 12 – (5,0,0) BAE ARIMA

We can compare this to a plot which doesn't use differencing and despite having a lower AIC and better residuals, we can see how crucial the RMSE and visualisation becomes to evaluating these models' success. In this model, we can see convergence back to the mean as trend isn't accounted for which means our forecast predicts the opposite direction.

<u>GSK Rolling ARIMA</u>

We can compare these results to the rolling ARIMA models in an aim to prevent stale models and more accurate prediction.

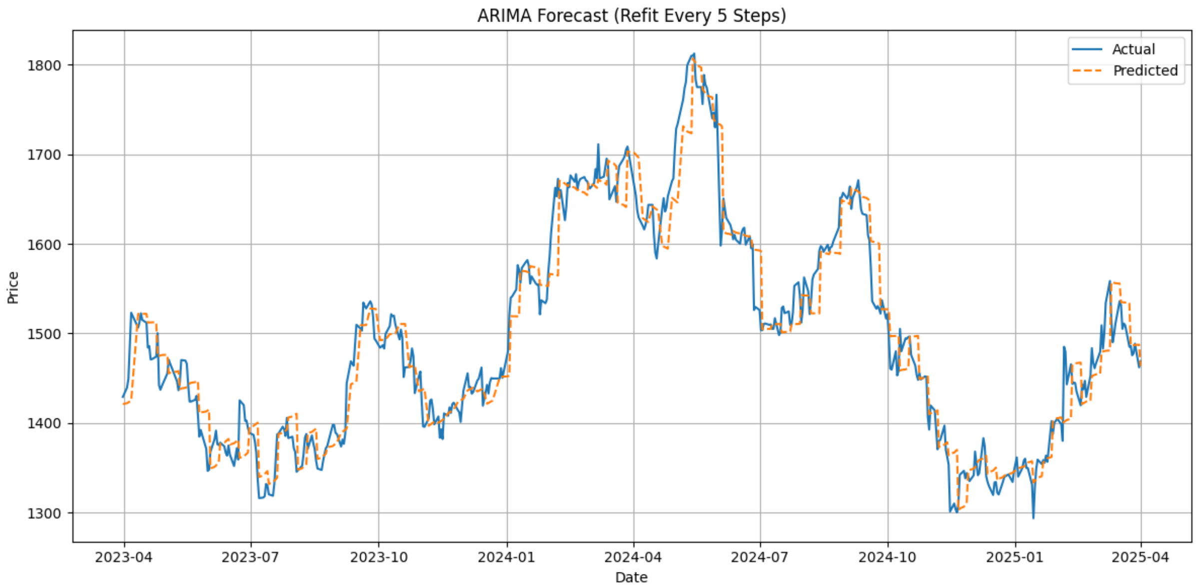

Figure 13 – GSK Rolling ARIMA

Above shows the forecast of the rolling ARIMA for the GSK dataset. As the data is given latest information every 5 steps, the forecast can pick up noise and follow the overall flow. However, as there is a considerable lag, it may not perform long or short term as the prediction may be sufficiently behind to predict what is happening that day. I tried ways to deal with this but was unsuccessful showing the limitations of the model. Although rolling ARIMA reduces RMSE to 31.13 by incorporating updated information, the lag in predictions reduces its practical applicability for real-time forecasting.

<u>LSTM results</u>

For the LSTM, the first model I built was based upon recurring updates, similar to the way a rolling ARIMA works.

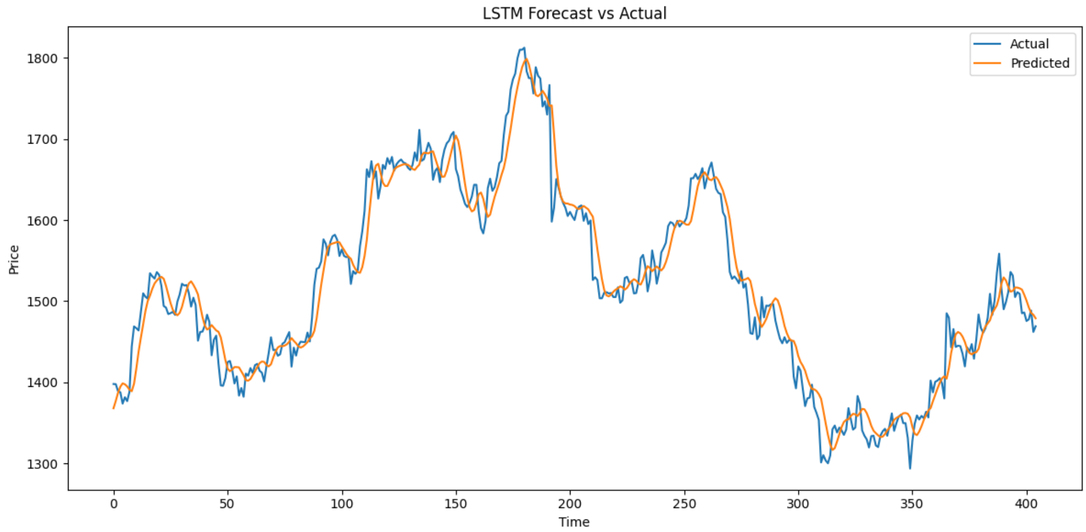

Figure 14 – GSK LSTM forecast

Figure 14 shows an example of the forecast against the actual GSK data. we can see yet again the improved predictions that are made and the ability to learn the intricate patterns learnt by the model. However, when reviewing this prediction, we can again see a lag, preventing its usability. We achieve an RMSE value of 26.57 which shows an improvement in the prediction over rolling ARIMA, but more exploration is needed to improve the model for longer term forecasting.

```python
# Forecasting
def recursive_forecast(model, initial_sequence, n_steps_ahead):
    """
    Use predictions to create values at next time point
    """
    sequence = initial_sequence.copy()
    predictions = []

    for _ in range(n_steps_ahead):
        seq_reshaped = sequence.reshape((1, sequence.shape[0], sequence.shape[1]))
        pred_scaled = model.predict(seq_reshaped, verbose=0)
        pred = scaler.inverse_transform(pred_scaled)
        predictions.append(pred[0, 0])
        sequence = np.vstack((sequence[1:], pred_scaled))

    return np.array(predictions)
```

Figure 15 – forecasts using predictions code

Figure 15 shows the code I used to use the predictions made at each timepoint to build an LSTM model. Therefore, after every prediction is made this value then becomes the end of the 100-day sequence, allowing us to forecast further in advance. This can be effective if there is a clear pattern in the data that can be learned such as seasonality.

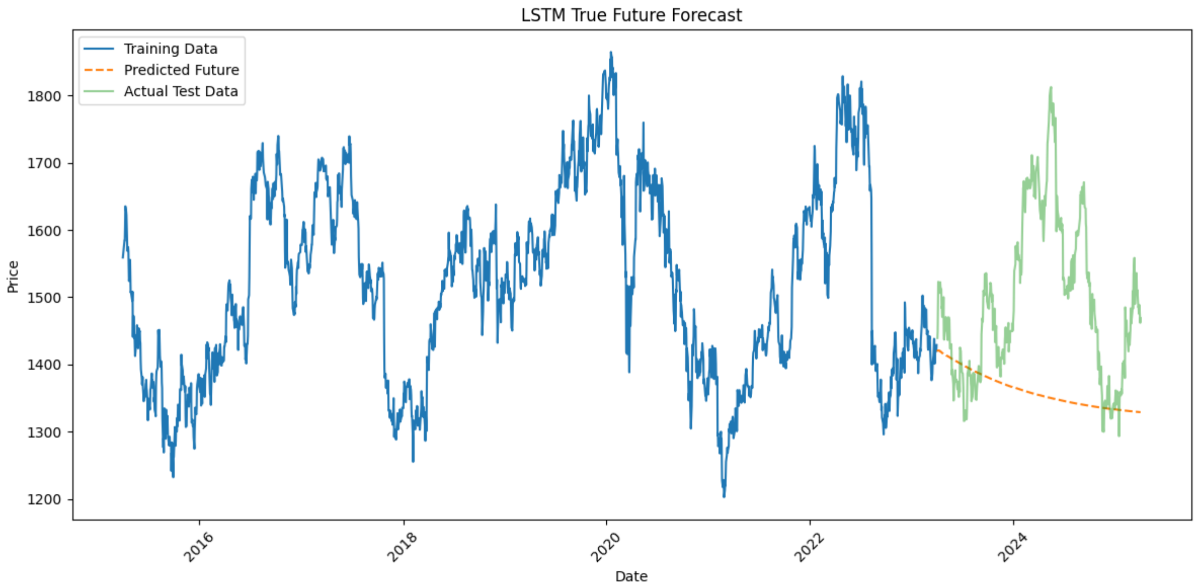

Figure 16 – GSK LSTM using predictions.

Figure 16 shows us that the LSTM model I built was unable to learn from the GSK dataset. This model shows a prime example of the negatives of using predictions as once it predicts a wrong value, the results tend to get worse, as error can become larger. Overall, this LSTM model had an RMSE of 184.5 which is higher than the ARIMA model built for the data. As the prediction is wrong, this shouldn't be used for long term investing on stationary data as the LSTM currently is built.

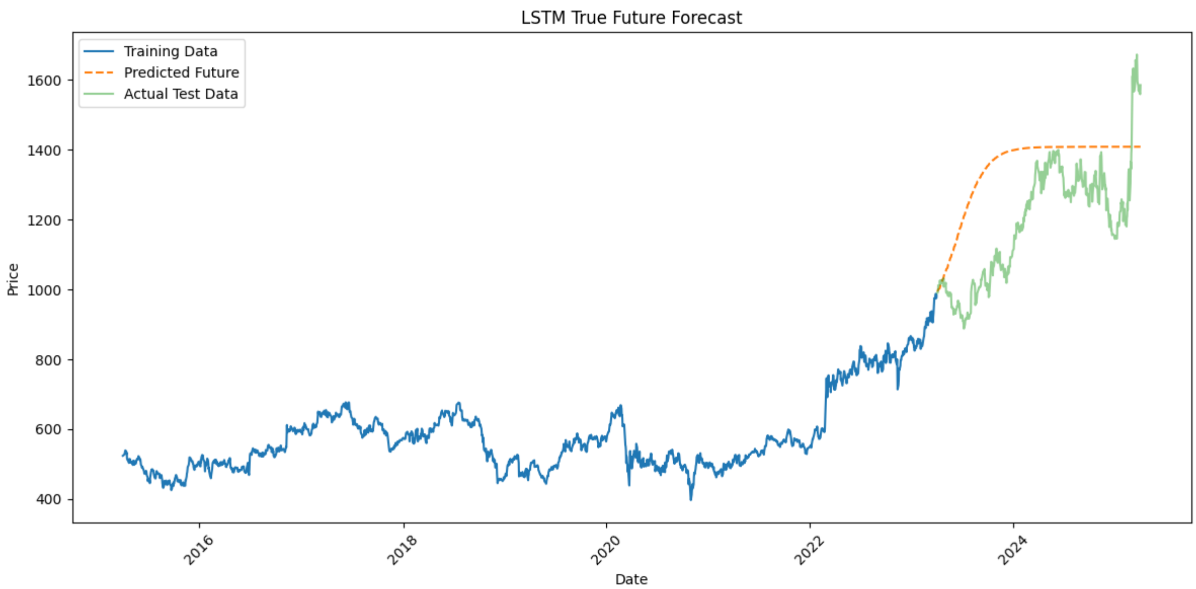

Figure 17 – BAE LSTM using predictions.

Figure 17 shows the LSTM model run on the BAE Systems data. Here we can again see the overreliance to rely on the predictions and continuously predict in that direction. This causes us to predict a much sharper increase than the true data, however it does plateau at a level the data later reaches, similar to the autoregressive drift suffered in an ARIMA. This model seems to behave sensitively to the previous few points meaning that a higher window of prediction could be beneficial.

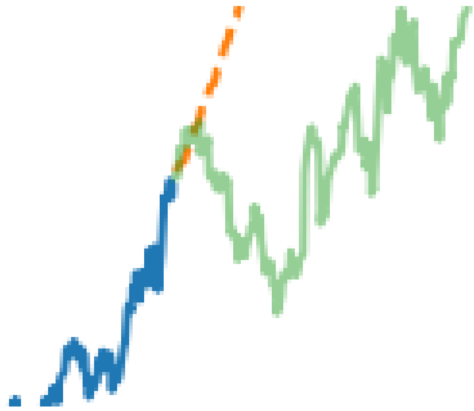

Figure 18 - Zoomed version of Figure 17

Figure 18 shows the point at where our predictions begin in the BAE LSTM. This shows that for the first few datapoints, we have a strong grasp of what will happen. In this case, the model would work well for short term trading however this may be different if we train from the wrong point. For example, about 10 timepoints later we see a drop which the model would struggle to predict, as it is clearly strong at learning the overall trend. This LSTM has a RMSE of 203.4, which is again worse performing than the best ARIMA effort.

Discussion
To summarise, ARIMA can work well for short term forecasting as it learns from previous data strongly. Potentially exploring higher values could be beneficial to see if we can gain further improvements. The main issue with my (2,0,0) model was the convergence to the mean long term.
When dealing with trend on the BAE Systems dataset, we can see how effective differencing is at improving the predictions we can make and capturing trend for long term forecasting. Differencing helps to make the mean constant and enables us to make the data stationary to allow ARIMA to work effectively.

When using these ARIMA predictions we can almost use the line prediction as a buy/sell line to make decisions, despite not picking up that nonlinearity. As for rolling ARIMA, despite being able to have a better accuracy, the lag makes it not useful as the predictions are delayed meaning the decisions are not being informed properly.

The LSTM model built were able to outperform the rolling ARIMA, however there was still concerns about lag within data. LSTMs did help the opportunity to learn these nonlinear relationships. Using prior predictions for LSTMs meant the predictions followed the direct of the first few predictions and therefore if the model was wrong, we predicted very wrong reducing the overall confidence in the model as errors were compounded. Going forward, I feel this model would benefit from optimisation to improve performance by looking at the layers used, number of units and the window being used to predict forwards. By using more tuning and training on more data such as (Roondiwala, 2017) this model could overall perform better.

Overall, we can see how these various models have positives and negatives in their usability. One major difficulty with the setup of my experiment could be the window I am aiming to predict. Prior researchers focused on a shorter window and issues can be seen in my ARIMA and LSTM as after a while, my models become stale and continuously predict in the same direction. This could be fixed by focusing on a short-term forecasting as the accuracy can be stronger and errors are not compiled.
This suggests that ARIMA remains the stronger choice for short-term prediction of stationary series such as GSK, while LSTM may offer advantages for trending stocks such as BAE, albeit requiring more optimisation.

One significant difference in the results is that we have 2 datasets that have varying characteristics. Firstly, the GSK data fluctuates around the mean, making it already stationary and ARIMA should work well. However, the randomness of direction can often make it tricky to learn patterns, especially if we start our prediction from the wrong point. Therefore, this dataset is better used as short-term tool, as long term it fluctuates too much.

In contrast, despite needing differencing the trend in the BAE Systems data makes it easier to apply either method as the relationship is clearer and noise is less influential. The pattern is much more stable, but it is important to remember that this won't stay the case. Overall, we can see how these two different industries behave differently showing a universal model may be more difficult to apply over all stocks.

When comparing to the literature, my results partly support existing findings. For example, Mondal (2014), found that ARIMA can be effective for short-term forecasting, but its performance deteriorates in longer horizons. Likewise, my results confirmed (Khan 2014)'s conclusion that model selection plays a key role, as trying different ARIMA structures significantly affected accuracy. However, unlike both studies, my models underperformed overall, which may be due to differences in dataset size, prediction horizon, and the limited number of ARIMA variations I evaluated. Going forward, I would use longer time intervals such as monthly data to improve long-term stability while continuing to refine short-term forecasts.

One big issue that the stock market data can provide is sudden shocks caused by external events. On a day-to-day basis, this can cause rapid changes which models can struggle to keep up with. An example of this can be found by looking at the EasyJet stock data.

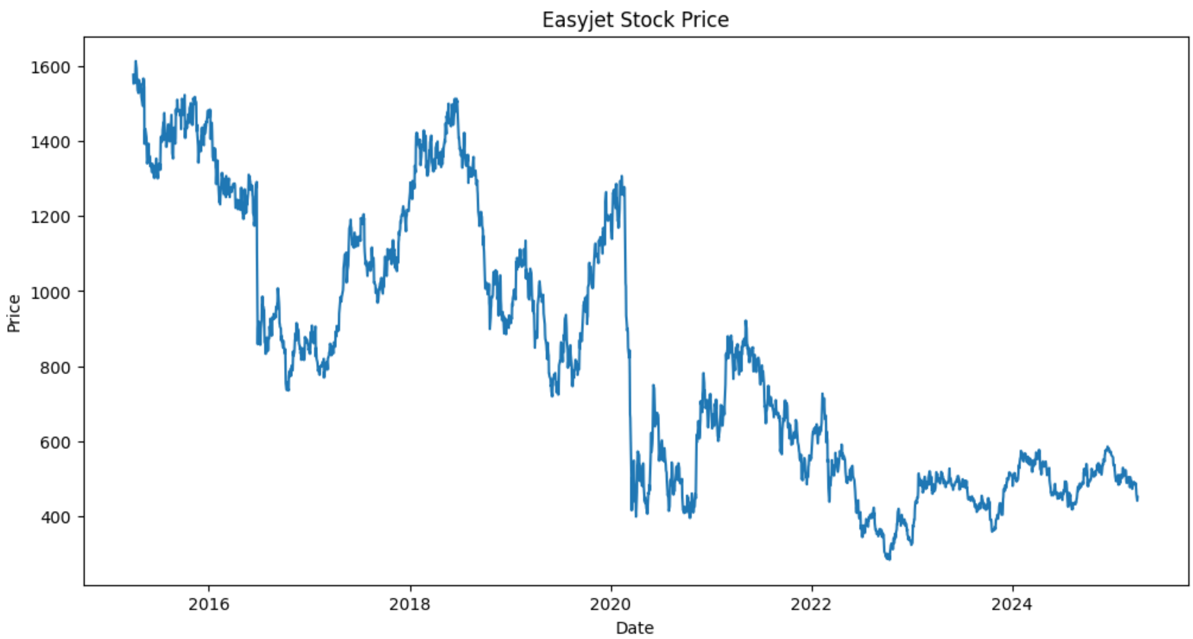

Figure 19 – EasyJet Price

Figure 22 shows a huge drop in early 2020 for the EasyJet share price, lining up with the start of the COVID-19 pandemic where air travel was reduced, directly impacting this industry.

Therefore, there is a scope for stock market prediction to work alongside news sentiment analysis to jointly with this research. This could provide enhanced predictive power that statistical models alone would not be able to achieve. This could include an additional of a score of how positive the news and how this may affect the stock. For example, if war is a prevalent factor in the news, aerospace companies such as BAE Systems may see influx in price due to demands.

Another way we could do this is by using other stocks to predict what may happen in a stock we try and predict. A known example of this is when expected or actual returns on bonds, equities, and real estate fall, the commodity of gold can see large investments (Investopedia, 2011). This is due to gold being seen as a safer investment as the value is less fragile to change.

Conclusion

This project has shown the process of how we can use ARIMA to predict the value of stocks and how it can be used. ARIMA is a strong method to use as a baseline and build a strong indication of what may happen especially for a limited number of timepoints. Despite not capturing nonlinearity it excelled over models built that tried to manage this by achieving a lower RMSE and an overall stronger forecast. Overall ARIMA was a much stronger model for short term forecasting bases upon the datasets used.

The project should therefore be used for short-term trading the models built have shown weaknesses when predicting long-term, mainly due to high volatility within the values. In future, we should aim to incorporate methods to improve the ability to use the models over a longer duration as well as looking to explore hybrid methods combine positives of a mixture of models as LSTM could help add nonlinearity to a strong ARIMA model.

References

Investing.com UK ,GSK share price – GSK shares, https://uk.investing.com/equities/glaxosmithkline

Investing.com UK ,BAE systems share price – BAE systems shares - https://uk.investing.com/equities/bae-systems

Investing.com UK ,EasyJet share price –EasyJet systems shares - https://uk.investing.com/equities/easyjet

Mondal, P., Shit, L. and Goswami, S., 2014. Study of effectiveness of time series modeling (ARIMA) in forecasting stock prices. International Journal of Computer Science, Engineering and Applications, 4(2), p.13.

Pant, J., Pant, H.K., Kotlia, P., Singh, D., Sharma, V.K. and Bhatt, J., 2024, August. Utilizing the prophet model for time series based stock price forecasting. In 2024 5th International Conference on Electronics and Sustainable Communication Systems (ICESC) (pp. 13-17). IEEE.

Choi, H.K., 2018. Stock price correlation coefficient prediction with ARIMA-LSTM hybrid model. arXiv preprint arXiv:1808.01560.

Khan, S., 2020. ARIMA model for accurate time series stocks forecasting. International Journal of Advanced Computer Science and Applications.

Pai, P.F. and Lin, C.S., 2005. A hybrid ARIMA and support vector machines model in stock price forecasting. Omega, 33(6), pp.497-505

Roondiwala, M., Patel, H. and Varma, S., 2017. Predicting stock prices using LSTM. International Journal of Science and Research (IJSR), 6(4), pp.1754-1756.

Khaidem, L., Saha, S. and Dey, S.R., 2016. Predicting the direction of stock market prices using random forest. arXiv preprint arXiv:1605.00003.

Idrees, S.M., Alam, M.A. and Agarwal, P., 2019. A prediction approach for stock market volatility based on time series data. IEEE Access, 7, pp.17287-17298.

Dong, Y., Li, S. and Gong, X., 2017, April. Time series analysis: An application of arima model in stock price forecasting. In 2017 International Conference on Innovations in Economic Management and Social Science (IEMSS 2017) (pp. 703-710). Atlantis Press.

Analytics Vidhya, 2025, Understanding Stationary Time Series Analysis, https://www.analyticsvidhya.com/blog/2021/06/statistical-tests-to-check-stationarity-in-time-series-part-1/#Augmented_Dickey-Fuller_Test

Olah, C., 2015. Understanding LSTM Networks. Colah's Blog, 27 August. Available at: https://colah.github.io/posts/2015-08-Understanding-LSTMs/.

Hyndman, R.J. and Khandakar, Y., 2008. Automatic time series forecasting: the forecast package for R. Journal of statistical software, 27, pp.1-22.

Utilizing predictive capabilities to anticipate the volatile stock market can introduce a sense of stability into an individual's life

Investopedia. (2011, February 22). What drives the price of gold?. Investopedia.: https://www.investopedia.com/financial-edge/0311/what-drives-the-price-of-gold.aspx