

stock market price prediction using machine learning

Anza Raja & Nimita Ankireddypalli



November 26, 2024

AI FInal paper

Table of Contents

[Introduction 2](#_Toc184162417)

[Methods 3](#_Toc184162418)

[a. Data Collection and Preprocessing 3](#_Toc184162419)

[*Data Collection:* 3](#_Toc184162420)

[*Preprocessing:* 4](#_Toc184162421)

[b. Models 4](#_Toc184162422)

[Results 6](#_Toc184162423)

[Model Evaluation Metrics 6](#_Toc184162424)

[Linear Regression: 6](#_Toc184162425)

[ARIMA: 6](#_Toc184162426)

[LSTM: 6](#_Toc184162427)

[Visual Comparisons 7](#_Toc184162428)

[Conclusions 8](#_Toc184162429)

[Figures 10](#_Toc184162430)

[Linear regression model plots for various stocks 10](#_Toc184162431)

[ARIMA model plots for various stocks 11](#_Toc184162432)

[LSTM model plots for various stocks 13](#_Toc184162433)

[Works Cited 16](#_Toc184162434)

# Introduction

The prediction of stock market prices has long been a field of great interest in finance and data science. Accurate forecasts can aid investors and financial analysts in decision-making, potentially leading to significant economic advantages. However, stock price prediction is an inherently difficult problem due to the volatile, non-linear, and time-dependent nature of financial data (Brownlee, 2020).

Our interest in pursuing this project stems from our fascination with the stock market, and specifically the stocks we chose to analyze which are Apple, Microsoft, Johnson & Johnson, Pfizer, JP Morgan, and the Bank of America. We chose to analyze two technology companies, two healthcare companies, and two banking companies to get a variety of results and to compare the stock data across different industries.

In this project, we compared the performance of three distinct models—Linear Regression, ARIMA, and LSTM—in predicting stock prices. By using historical data sourced from Yahoo Finance, our goal was to determine the strengths, weaknesses, and overall suitability of these models for forecasting time series data (Chollet, 2021). This study further underscores the trade-offs between the simplicity of statistical methods and the complexity of deep learning approaches in solving real-world forecasting problems.

We formulated our project based on the models and statistical analyses discussed in Shah, Vaidya, and Shah (2022) which outlined the model’s strengths and weaknesses in handling non-linear, volatile financial data. The article was especially useful in understanding the methodologies and capabilities of the models, specifically LSTM and ARIMA prior to our implementation of them. We learned what to look for in our results which enabled us to properly and accurately analyze the outputs of our code.

## Methods

## a. Data Collection and Preprocessing

*Data Collection:*  
We gathered historical stock data for Apple, Microsoft, Johnson & Johnson, Pfizer, JP Morgan, and the Bank of America using the yfinance library (Ran, 2019). The dataset included daily records of open, close, high, low, and volume spanning over twenty-four years from the beginning of 2010 to the end of 2023. The analysis focused on the stock's closing price, which is often a key indicator of both short and long-term stock growth for investors.

We utilized the pandas library to convert our stock data into a DataFrame for ease of use and analysis throughout the rest of our code (The Pandas Development Team, 2024).

We began by inspecting the data to ensure that we had what we needed by plotting all six stocks’ closing prices over the time frame on the graph displayed below.

**A graph of stock closing prices

Description automatically generated**

### *Preprocessing:*

1. **Stationarity Check:**  
   We began by conducting a stationarity check on each of the stocks. Time series data must be stationary for models like ARIMA to work effectively. Stationarity implies that the statistical properties (mean, variance, autocorrelation) of the dataset remain constant over time (Wei, 2006).
   * Non-stationary data can lead to spurious results, where the model falsely interprets trends as predictive information.
   * We employed the Augmented Dickey-Fuller (ADF) test to assess stationarity. Initial tests revealed that the data was not stationary due to a clear upward trend over time. To address this, we applied differencing, a common transformation technique (Brownlee, 2020). We used the pandas library .diff() method to do so (The Pandas Development Team, 2024).
2. **Autocorrelation and Partial Autocorrelation (ACF and PACF):**  
   After differencing, we analyzed ACF and PACF plots to identify significant lags in the data. These plots helped us determine the order parameters for the ARIMA model. The ACF revealed correlations between the time series and its lagged values, while the PACF helped isolate the direct effects of individual lags. We were able to determine that we should use a p value of 2, a d value of 0, and a q value of 1.
3. **Normalization:**  
   For the LSTM model, we normalized the data to ensure faster convergence and reduced the risk of exploding or vanishing gradients during training (Chollet, 2021) We wanted to ensure that our data was as consistent as possible without a big risk of overfitting.
4. **Splitting the Dataset:**  
   The dataset was divided into training (80%) and testing (20%) sets to evaluate model performance on unseen data.

## b. Models

**Linear Regression:**  
A baseline model that uses the past 30 days of stock prices to predict the next day’s closing price. It provides a straightforward approach, ideal for comparison with more advanced models. As its name implies, Linear regression assumes that there is a strictly linear relationship between the independent and dependent variable, which holds true to some extent for stock data.

**ARIMA (Autoregressive Integrated Moving Average):**  
ARIMA is a statistical model optimized for time series forecasting. It uses three parameters:

* p (autoregression): Determines the number of lag observations to include. Based on PACF, we set this to 2 to avoid overfitting.
* d (differencing order): Set this to 0 because we had already differenced our data prior to running the ARIMA test on it.
* q (moving average): Determined using ACF to be 1.

ARIMA captures linear trends and seasonality but struggles with sudden changes in stock prices. It is beneficial for short-term forecasting but does not hold much accuracy when used for long-term forecasting. ARIMA can also struggle with predicting turning points in the data which can be problematic when working with stock data which tends to have a substantial amount of turning points.

**LSTM (Long Short-Term Memory):**  
LSTMs are recurrent neural networks designed to handle sequential data. By using memory cells, they retain long-term dependencies, which are critical for predicting stock prices influenced by extended historical trends. The LSTM model was trained on sliding windows of the past 60 days of stock prices. As the name implies, LSTMs are good for handling long-range data by remembering data from previous time steps and utilizing that to make predictions on future data while continuing to retain the previous data. That being said, LSTM is a more complex model, and this is reflected in its runtime as it has many more layers to it than the other two models used for our project.

# Results

## Model Evaluation Metrics

Each model's performance was evaluated using Mean Squared Error (MSE) and Root Mean Squared Error (RMSE). These metrics quantify the average squared and root-squared differences, respectively, between predicted and actual prices, as discussed in Shah, Vaidya, and Shah (2022).

### Linear Regression:

* **MSE:** 0.015
* **RMSE:** 0.122
* **Insights:** Linear Regression struggled to capture the non-linear nature of stock price movements. The model performed poorly during periods of high volatility, leading to large prediction errors. This was to be expected with stock data as it does not usually tend to be linear, instead presenting a lot of turning points and complex trends. Linear Regression will sometimes underfit the data as it is not an advanced enough model to handle more complex data, which seems to have happened in our case.

### ARIMA:

* **MSE:** 0.008
* **RMSE:** 0.089
* **Insights:** After differencing the data, ARIMA captured linear trends and seasonal patterns effectively. However, the model showed limitations in adapting to abrupt price changes and displayed slight lag in its predictions. While better than Linear Regression in that it knows how to handle seasonality and trends, ARIMA still struggles with analyzing non-linear data which is shown by the MSE and RMSE still being higher than LSTM’s.

### LSTM:

* **MSE:** 0.003
* **RMSE:** 0.055
* **Insights:** The LSTM model outperformed both Linear Regression and ARIMA by capturing non-linear dependencies and longer-term trends. It provided predictions that closely followed actual prices, though at the expense of computational resources and increased training time, which was as we had hypothesized. Hence, LSTM was the best model for the purposes of our project but the most significant disadvantage we observed was the training time compared to the other two models.

### Visual Comparisons

* Predicted vs. actual prices highlighted the superiority of LSTM in closely tracking the data.
* ACF and PACF plots validated the stationarity of the ARIMA data and informed the selection of parameters, demonstrating their critical role in time series modeling.
* The plots and graphs can be seen after the Conclusions section.

# Conclusions

This project demonstrated the importance of model selection and data preparation in time series forecasting, specifically with financial data such as stocks. Our key takeaways are as follows:

1. **Stationarity and Model Accuracy:**  
   The necessity of stationarity for ARIMA was evident. Without differencing, the model would have failed to produce meaningful forecasts, emphasizing the importance of preprocessing in time series analysis. However, this was not necessary for the Linear Regression and LSTM analyses and these models performed relatively well without needing to be differenced.
2. **Model Strengths and Weaknesses:**
   * **Linear Regression:** While simple and interpretable, this model failed to account for the complex, non-linear nature of stock prices, making it unsuitable for practical forecasting. This model was easy to implement but we feel that instead of using Linear Regression, we should have used a hybridized model of ARIMA and LSTM to see if both models working together, in a sense, was the most accurate model.
   * **ARIMA:** The ARIMA model was effective for short-term forecasts and seasonal trends. ARIMA remains a reliable statistical approach but requires careful parameter tuning and struggles with volatile data. In the future, we would like to experiment with different p, d, and q values and see if we would be able to produce more accurate results from our model.
   * **LSTM:** Despite its computational cost, LSTM excelled by capturing intricate patterns in the data, making it ideal for dynamic and long-term predictions. This model was easily the most accurate what with its black-box like structure and complex neural network. We would like to increase the dataset size as a future enhancement to see if this further improves the model’s accuracy and to ensure that we did not overfit the model without realizing it.
3. **Role of ACF and PACF:**  
   These tools provided critical insights into lag dependencies, helping fine-tune ARIMA parameters and ensuring the model captured meaningful trends. We were able to identify what p, d, and q parameters to use by running the Augmented Dickey-Fuller test to determine stationarity.
4. **Accuracy and Implications:**  
   The LSTM model's superior accuracy highlights the potential of deep learning in stock market forecasting. However, its complexity demands substantial computational resources and expertise, which may limit its accessibility to non-specialists. Additionally, with it being a complex black-box model, it lacks explainability which can make it more difficult to troubleshoot should issues present and to understand what exactly is happening within the model that produces the outputs we are seeing.

In conclusion, while all three models offered valuable insights, LSTM emerged as the most robust tool for stock price prediction. Future work could explore hybrid models that combine the strengths of ARIMA's interpretability and LSTM's adaptability. Additional factors such as economic indicators or sentiment analysis could further enhance the predictive accuracy of the models. We hypothesize that with such additions/changes, the accuracy of our models would improve.

# Figures

## A graph of red and blue dots Description automatically generatedLinear regression model plots for various stocks

A graph with red and blue dots

Description automatically generatedA graph with red and blue dots

Description automatically generatedA screen shot of a graph

Description automatically generatedA graph of red and blue dots

Description automatically generatedA graph of red and blue dots

Description automatically generated

## ARIMA model plots for various stocks

**A graph showing a line

Description automatically generated with medium confidenceA graph with purple line

Description automatically generatedA graph showing a line of growth

Description automatically generated with medium confidence**

A graph showing a line graph

Description automatically generated with medium confidenceA graph showing a line graph

Description automatically generated with medium confidenceA graph showing a line of growth

Description automatically generated with medium confidence

## LSTM model plots for various stocks

**A graph showing a line

Description automatically generated with medium confidenceA graph showing a line

Description automatically generated**

A graph showing a line

Description automatically generated with medium confidenceA graph with blue lines

Description automatically generated

A graph showing a line

Description automatically generated with medium confidenceA graph showing a line graph

Description automatically generated with medium confidence

# Works Cited

# Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jozefowicz, R., Jia, Y., Kaiser, L., Kudlur, M., ... Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems [Software]. Retrieved from <https://www.tensorflow.org>

# ​Brownlee, J. (2020). Introduction to Time Series Forecasting with Python. Retrieved from [www.inf.szte.hu: https://www.inf.szte.hu/~korosig/teach/books/Jason%20Brownlee%20-%20Introduction%20to%20Time%20Series%20Forecasting%20with%20Python%20-%20How%20to%20Prepare%20Data%20and%20Develop%20Models%20to%20Predict%20the%20Future-v1.9%20(2020).pdf](https://www.inf.szte.hu/~korosig/teach/books/Jason%20Brownlee%20-%20Introduction%20to%20Time%20Series%20Forecasting%20with%20Python%20-%20How%20to%20Prepare%20Data%20and%20Develop%20Models%20to%20Predict%20the%20Future-v1.9%20(2020).pdf)

# ​Chollet, F. (2021). Deep learning with Python. Retrieved from [www.papiro-bookstore.com: https://www.papiro-bookstore.com/wp-content/uploads/2021/12/Deep-Learning-with-Python.pdf](https://www.papiro-bookstore.com/wp-content/uploads/2021/12/Deep-Learning-with-Python.pdf)

# Harris, C.R., Millman, K.J., van der Walt, S.J. et al. Array programming with NumPy. Nature 585, 357–362 (2020). DOI: 10.1038/s41586-020-2649-2. ([Publisher link](https://www.nature.com/articles/s41586-020-2649-2)).

# J. D. Hunter, "Matplotlib: A 2D Graphics Environment", Computing in Science & Engineering, vol. 9, no. 3, pp. 90-95, 2007

# Microsoft. (2024). Visual Studio Code (Version 1.93.1) [Software]. <https://code.visualstudio.com>

# Ran, A. (2019). yfinance: Download market data from Yahoo! Finance's API (Version 0.2.49) [Python package]. Retrieved from <https://pypi.org/project/yfinance/>

# Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

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

​​Shah, J., Vaidya, D., & Shah, M. (2022, 08 14). A comprehensive review on multiple hybrid deep learning approaches for stock prediction. Retrieved from www.sciencedirect.com: <https://www.sciencedirect.com/science/article/pii/S2667305322000497>

# The Pandas Development Team. (2024). pandas: Powerful Python data analysis toolkit (Version 2.2.3) [Python package]. Retrieved from <https://pandas.pydata.org>

​Wei, W. W. (2006). Time series analysis (Second edition). Retrieved from civil.colorado.edu: <https://civil.colorado.edu/~balajir/CVEN6833/lectures/wwts-book.pdf>