Northwestern
SCHOOL OF
PROFESSIONAL STUDIES

Stock Price Trend Prediction:

Predicting Stock Price for Next Day Using Time Series and Machine Learning

Authors:

Eswarankarthik Paranthaman,

Amish Jain,

Mahmoud Mackawy,

Rakib Ahsan

Professors:

Dr. Donald Wedding & Dr. Srabashi Basu

MS DSP 422 Practical Machine Learning

March 17, 2025

Table of Contents

Executive Summary 3
Problem Statement 4
Exploratory Data Analysis 5
Data Preparation 10
Methodology 11
Findings and Conclusion 13
Lessons Learned 14
References 17

Executive Summary

Stock price prediction is a crucial yet challenging task in financial market analysis, as stock prices are influenced by multiple factors, including economic conditions, investor sentiment, and company performance. This project aims to predict the next day's stock price using machine learning techniques, leveraging historical stock data from Yahoo Finance.

Our study evaluates four different models: ARIMA, Linear Regression, LSTM, and a hybrid LSTM-XGBoost model. Each model was selected based on its ability to analyze time series data and capture patterns in stock price movements. The dataset consists of Apple Inc. (AAPL) stock prices from 1980 to 2025, including key features such as opening price, closing price, volume, and engineered variables like lag features and Fourier transformations to capture trends and seasonality.

The evaluation of these models was conducted using performance metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE). Our initial results indicate that ARIMA and Linear Regression performed significantly better than deep learning models like LSTM and LSTM-XGBoost, which exhibited high error rates. This suggests that for short-term stock price prediction, traditional statistical models may still provide better accuracy compared to complex deep learning approaches.

Future work will focus on improving model performance through hyperparameter tuning, alternative feature engineering techniques, and testing on multiple stocks to assess generalizability. Additionally, we plan to explore more advanced deep learning architectures, such as Transformer-based models, to enhance predictive accuracy.

This project provides valuable insights into the effectiveness of different machine learning models in financial forecasting, helping investors and analysts make data-driven decisions in an unpredictable stock market.

Problem Statement

Stock price prediction is a fundamental yet highly complex challenge in financial markets. Prices fluctuate due to various factors, including economic conditions, market sentiment, company performance, and geopolitical events. Accurately forecasting stock prices can help investors, traders, and financial analysts make informed decisions, minimize risks, and identify profitable opportunities.

Traditional approaches to stock prediction, such as fundamental and technical analysis, have been widely used but often fail to capture the dynamic and nonlinear nature of stock price movements. With advancements in artificial intelligence and machine learning, new data-driven methods have emerged, offering improved predictive capabilities. However, selecting the most effective model remains a challenge, as different algorithms have varying strengths in handling time series data, feature interactions, and market volatility.

This project aims to develop and compare machine learning models to predict the next day's stock price using historical data and technical indicators. We evaluate the performance of ARIMA, Linear Regression, LSTM, and an LSTM-XGBoost hybrid model to determine the most effective approach for short-term stock price forecasting. By analyzing prediction accuracy using Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE), this study seeks to provide insights into the best-performing methodologies for financial market predictions.

Understanding which machine learning models work best for stock price forecasting will help improve risk management, enhance algorithmic trading strategies, and optimize decision-making for investors in an ever-changing stock market.

Exploratory Data Analysis

In our research paper, we conducted an extensive Exploratory Data Analysis (EDA) to understand the characteristics of our dataset. We first imported the dataset from Yahoo finance and performed data cleaning, ensuring consistency in column headings and converting datatypes to appropriate formats such as datetime and float. To explore overall trends, we plotted the target variable, Close Price, against the date as shown in figure 1.1.

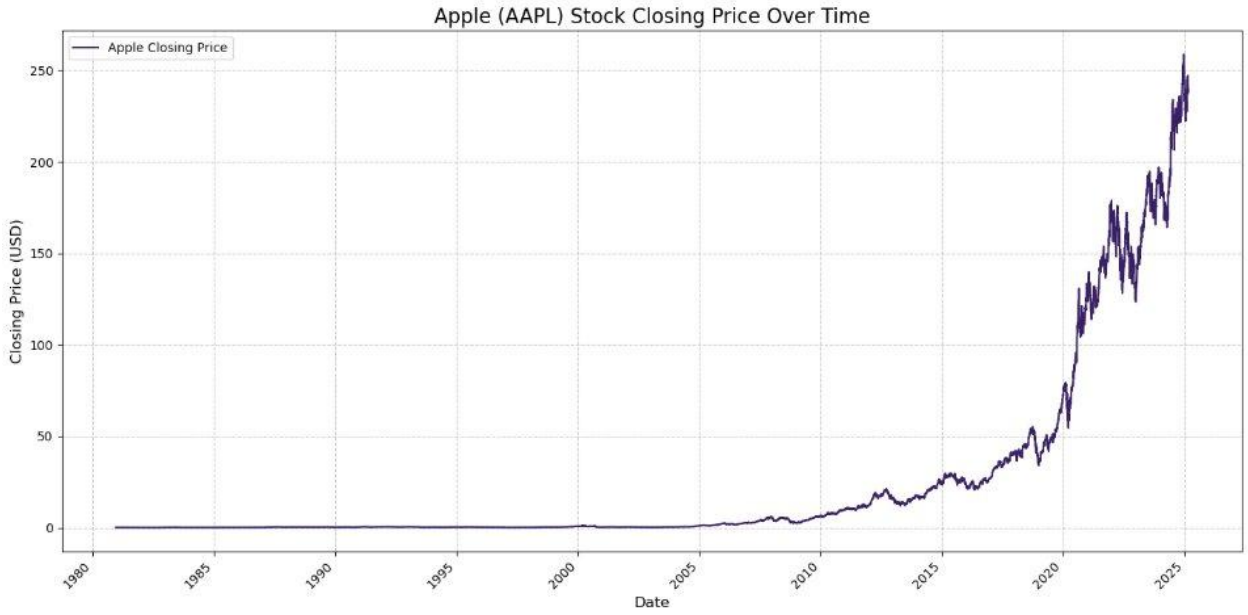

Figure 1.1. Close Price Over Time

We then decomposed the Close Price to observe its underlying components—trend, cyclicity, seasonality, and residual variations as seen in figure 1.2.—providing insights into its behaviour over time. To assess stationarity, we performed the Augmented Dickey-Fuller (ADF) test, which

yielded a high p-value (0.96), indicating non-stationarity. To address this, we applied differencing to stabilize the mean.

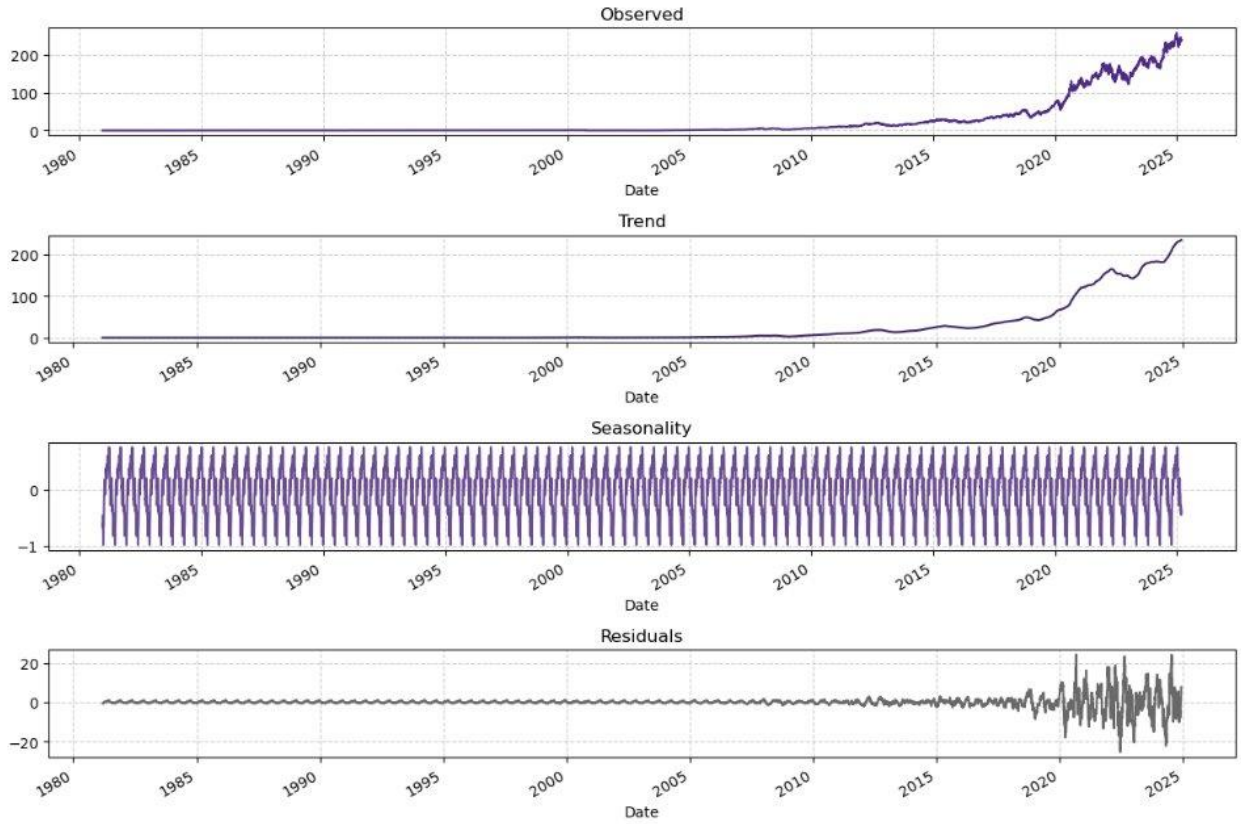

Figure 1.2. Close Price Trend, Seasonality, and Residuals Analysis

We further analyzed Autocorrelation (ACF) and Partial Autocorrelation (PACF) plots as seen in figure 1.3. to identify dependencies in lagged values, aiding in model selection.

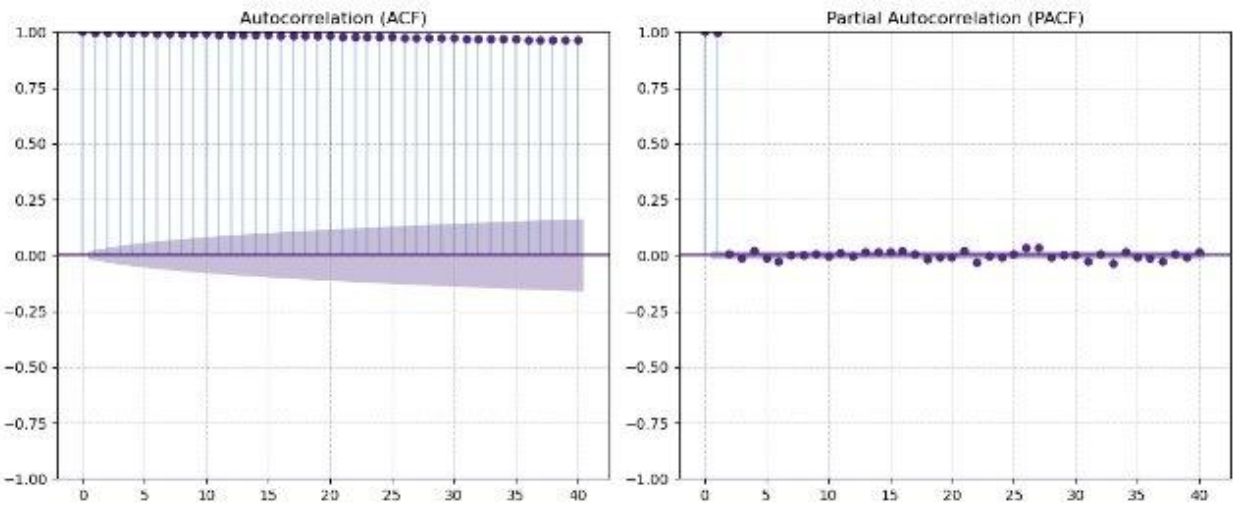

Figure 1.3. Autocorrelation (ACF) and Partial Autocorrelation (PACF) plots

ACF Plot (Left Side): The autocorrelation values are very high and persist across many lags (i.e: almost 40 lags), showing a slow decay. The presence of strong correlations at all lags suggests that the time series is non-stationary. The shaded confidence interval indicates that most autocorrelations are significant.

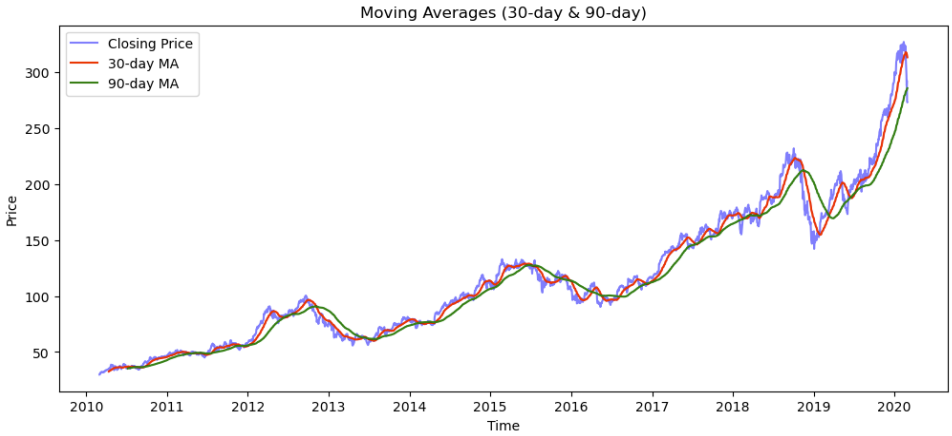

Figure 1.4. Moving Averages (MA30 and MA90)

PACF Plot (Right Side): The first lag shows significant spikes, while other lags remain close to zero. This suggests that the time series follows an AR(1) process.To reduce seasonality, we

smoothed the dataset using Moving Averages (MA30 and MA90) as seen figure 1.4., providing a clearer trend analysis.

Figure 1.5 illustrates Apple's daily stock return fluctuations from 2010 to 2020, highlighting the inherent volatility of Apple's stock prices, with numerous rapid swings in both positive and negative directions. The 2013 period shows a sharp drop, while mid-2014 exhibits the highest spike in return gains. Periods of heightened volatility, where return fluctuations are more frequent and pronounced, are noticeable throughout the decade. While daily returns vary significantly, no clear long-term upward or downward trend is visually apparent. Several distinct spikes and dips represent days with particularly large changes in stock price, suggesting that impactful events influenced Apple's value on those dates. The overall distribution of these daily returns appears somewhat symmetrical around zero, implying that large gains and large losses are roughly equally likely.

A notable positive spike occurred in early 2012, which can be attributed to Apple's strong financial position at the time. The company held a substantial cash reserve of $97.6 billion, making it highly attractive to investors and boosting market confidence (Carlson 2012). Conversely, the deepest drop in early 2013 was driven by a combination of factors, including a 12% decline in Apple's stock in January due to slowing consumer demand for iPhones. Increased competition, particularly from Samsung, further pressured Apple's prices and market share, leading to investor concerns and a sharp decline in stock returns (Gandel 2012).

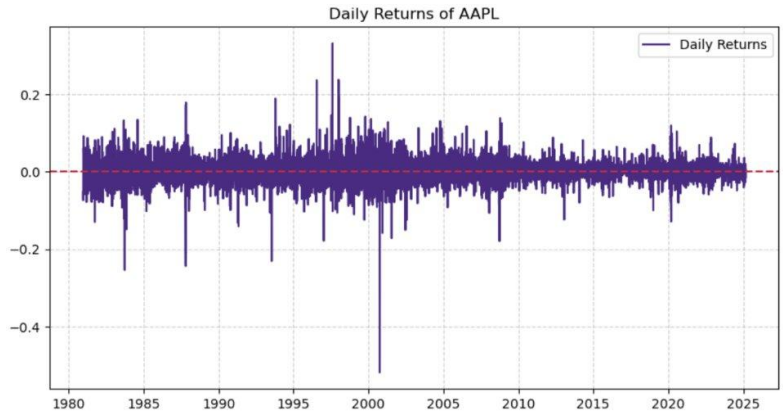

Figure 1.5. Daily Returns Over Time

The correlation heatmap presented below in figure 1.6. provides insights into the relationships among features, enabling a deeper understanding of their interdependencies and potential influence on the model. The analysis reveals that the target variable exhibits a nearly perfect correlation (~100%) with most features, except for volume, which demonstrates a moderate correlation (~60%). In contrast, close price differences and daily returns exhibit minimal correlation (~1%) with the target variable, indicating limited direct influence.

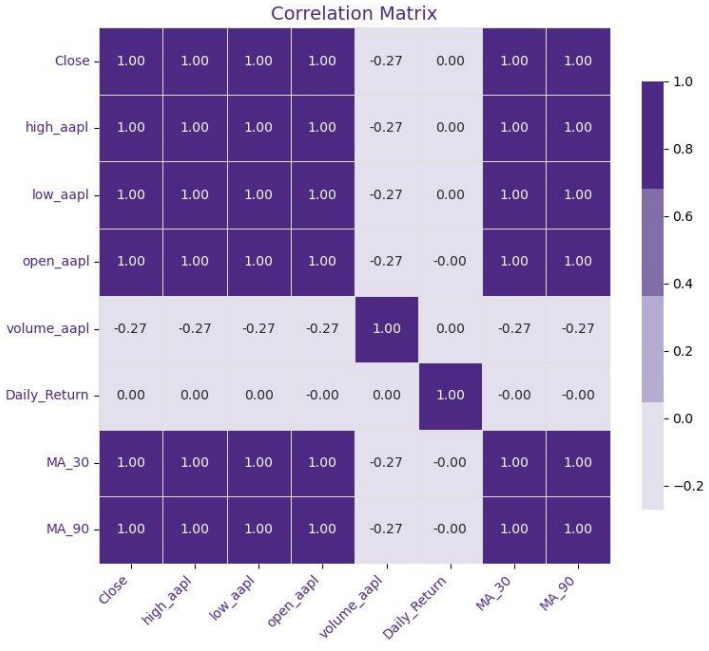

Figure 1.6. Correlation Heatmap

This comprehensive EDA allowed us to refine our dataset for robust time series modelling and forecasting.

Data Preparation

For this machine learning project, stock market data for Apple Inc. (AAPL) was sourced from Yahoo Finance. The dataset spans from December 12, 1980, to present date, and was downloaded using the yfinance library in Python.

The dataset consists of the following columns:

- Close: The closing price of the stock on each trading day.
- Volume: The total number of shares traded on that day.
- Open: The opening price of the stock on each trading day.
- High: The highest price the stock reached on each trading day.
- Low: The lowest price the stock reached on each trading day.

Feature Engineering

To enhance the dataset for machine learning applications, several additional features were engineered:

- **Lag Features**: Created to capture short-term trends in stock prices:
    - Lag_1: Previous day's closing price.
    - Lag_2: Closing price two days prior.
    - Lag_3: Closing price three days prior.

These features help capture short-term trends and autocorrelation in stock prices, which are important for time-series forecasting models.

- For LSTM we structured data so that each input sample contained past 10 days of stock prices to predict the next day's price.

Variable Transformations, Data Scaling & Assumptions

To prepare the data for modelling, we applied several transformations and scaling methods:

- **Feature Scaling:**
  - **Min-Max Scaling or Standardization:** Scaling improves model stability and predictive power. However, LSTM models generally perform better with standard scaling (zero mean, unit variance).
- **Differencing:**
  - For ARIMA, we used the Close_diff feature, representing the difference between the current and previous day's closing prices. This transformation helps ensure stationarity, a key assumption for ARIMA models.
- **Assumptions:**
  - Time-Series Models (ARIMA, LSTM): We assume that stock price data exhibits a temporal correlation structure.
- **LSTM-XGBoost Hybrid Model:** The assumption is that LSTM captures the sequential dependencies, while XGBoost models non-linear relationships, making them complementary.

Methodology

To predict the next day's stock price, we implemented a structured approach that involved data collection, feature engineering, model selection, and performance evaluation. Our methodology

ensures that the models are trained and tested on reliable financial data while using appropriate evaluation metrics to measure accuracy.

1. **Data Collection**

- The dataset was sourced from Yahoo Finance using the yfinance Python library.
- We focused on Apple Inc. (AAPL) stock, collecting daily historical data from December 12, 1980, to February 28, 2025.
- The dataset includes essential financial metrics such as opening price, closing price, high, low, and trading volume.

2. **Model Selection**

Different machine learning models were implemented and tested using slightly varying features. Here are the models used:

- **ARIMA** (AutoRegressive Integrated Moving Average):
  - **Features**: Close_Diff (the difference between consecutive closing prices).
  - ARIMA is suitable for capturing linear trends and seasonality in time-series data. We tested ARIMA using different orders (p, d, q) for model optimization.
- **Linear Regression**:
  - **Features**: Lag_1, Lag_2, Lag_3
  - A simple model that captures linear relationships between the target variable (next day's closing price) and the lagged features. Linear regression was used as a baseline to compare performance with more complex models.
- **Timeseries LSTM (Long Short-Term Memory)**:
  - **Features**: Series Sequence of past 10 days.
  - LSTM is a type of recurrent neural network (RNN) that is designed to model sequential data. We used this model to capture long-term dependencies and temporal patterns in the data.
- **LSTM-XGBoost Hybrid Model**:
  - **Features**: Series Sequence of past 10 days.

- This hybrid model combines LSTM's ability to capture sequential patterns with XGBoost's strength in handling non-linear relationships. The idea was to leverage the strengths of both models to improve accuracy.

3. Model Training and Evaluation

- The dataset was split into training and testing sets to validate model performance.
- We performed 80:20 split for the data. Training data span from 17-12-1980 to 19-04-2016 and Testing data spans from 20-04-2016 to 28-02-2025.
- We evaluated each model using the following metrics:
  - **Mean Squared Error (MSE)**: Measures the average squared difference between actual and predicted prices.
  - **Root Mean Squared Error (RMSE)**: Provides an interpretable measure of error magnitude.
  - **Mean Absolute Percentage Error (MAPE)**: Expresses prediction error as a percentage for better comparability.
- Model performance was analyzed to determine which approach yielded the most accurate predictions.

4. Tooling Setup

To develop and evaluate our models, we used the following tools:

- **Programming Language**: Python
- **Development Environment**: Anaconda Jupyter Notebook & Google Colab
- **Machine Learning Libraries**:
  - scikit-learn for data preprocessing and traditional models.
  - statsmodels for ARIMA implementation.
  - tensorflow/keras for deep learning models (LSTM).
  - xgboost for gradient boosting implementation.
  - matplotlib & seaborn for data visualization.

Findings and Conclusion

Models	Test		
	MSE	**RMSE**	**MAPE**
ARIMA	7.113	2.667	1.16%
Linear Regression	8.632	2.938	1.03%

Time Series LSTM	979.96	31.30	13.03%
Time Series LSTM_XGBoost	3,123.133	55.885	28.53%

Table 1. Models Evaluation

The table above shows the performance comparison between four models-ARIMA, Linear Regression, LSTM, and LSTM and XGBoost hybrid. ARIMA achieved the lowest MSE at 7.113 and RMSE at 2.667, with a MAPE of 1.16%, indicating strong predictive performance. Linear Regression followed closely with an MSE of 8.632, RMSE of 2.938, and the lowest MAPE at 1.03%.

In contrast, Time Series LSTM and Time Series LSTM_XGBoost exhibited significantly higher errors, with MSE values exceeding 900 and RMSE around 30. Notably, their MAPE values were extremely high at 13.03% and 28.53%, respectively, indicating severe prediction inaccuracies. These results suggest that traditional statistical models, particularly ARIMA and Linear Regression, outperformed deep learning-based approaches, which struggled with accuracy in this context.

In our effort to predict Apple's next-day closing price using a linear regression model, we initially observed that our predictions were not aligning well with actual market values. A thorough analysis, including visualizations, revealed that the discrepancies in critical scenarios were significant. This indicated that our model was not yet production-ready for real-world applications.

Lessons Learned

Future Work

1. **Incorporating advanced technical indicators**:
    a. Intoruduicng technical indicators like RSI, MACD, Bollinger Bands, Moving Average Convergence Divergence (MACD) Histograms, or Stochastic Oscillators

could provide more insight into stock price movements, especially in volatile markets.

2. **Incorporating External Data Sources**:
    a. **Sentiment Analysis**: Including sentiment analysis data from news sources or social media (Twitter, Reddit) could improve predictive accuracy, especially around key events like earnings reports or market shifts.
    b. **Economic Indicators**: Adding external macroeconomic indicators (e.g., GDP growth, unemployment rates, interest rates, inflation rates) could help the models understand broader economic trends and how they influence stock prices.
    c. **Alternative Data**: Using alternative data, such as satellite imagery of stores, supply chain data, or website traffic, could provide valuable predictive signals that are not fully reflected in historical stock prices alone.
3. **Incorporating News and Event Data**:
    a. Financial news and event data such as company earnings reports, product launches, or geopolitical events could be incorporated using natural language processing (NLP) models. These events often lead to price spikes or drops and could improve forecasting accuracy.
4. **Expanding to Multiple Stocks**:
    a. The current model is focused on Apple Inc. (AAPL), but future work could expand this approach to other stocks or even a portfolio of stocks. This would require adjusting the models to handle multi-stock datasets and possibly implementing multi-output regression tasks.
5. **Ensemble Methods**:
    a. Experimenting with ensemble learning methods, such as stacking models or blending predictions from different models (e.g., ARIMA, LSTM, XGBoost), could improve performance by leveraging the strengths of multiple models.
6. **Improving Model Interpretability**:
    a. While deep learning models like LSTM and LSTM-XGBoost can provide strong performance, they are often considered black-box models. Future work could explore ways to make these models more interpretable, such as using SHAP values or LIME (Local Interpretable Model-Agnostic Explanations) to understand the feature importance and model decisions.

Other Data Points to Include

1. **Volume Data**:
    a. Including volume data as an additional feature might improve predictive performance, especially in capturing price movements based on trading activity. Volume often correlates with market volatility and price reversals.
2. **Intraday Data**:
    a. While daily closing prices were used, intraday data (hourly or minute-level data) could capture more detailed trends and short-term price movements. This would require resampling the data and adjusting models to handle high-frequency data.
3. **Macroeconomic Data**:
    a. Incorporating macroeconomic data, such as interest rates, inflation rates, and GDP growth can help provide context for price movements, especially during periods of economic turbulence.
4. **Currency Exchange Rates**:
    a. For multinational companies like Apple, currency exchange rates can influence stock prices, especially when dealing with international revenue. Incorporating exchange rate data could provide additional signals for predicting stock movements.

References

[1] Atsalakis, George S., and Konstantinos P. Valavanis. Surveying Stock Market Forecasting Techniques – Part II: Soft Computing Methods. Expert Systems with Applications 36, no. 3 (2009): 5932–5941. https://doi.org/10.1016/j.eswa.2008.07.006.

[2] Chen, Ming-Hsiu, Wei-Chiang Hong, and Chein-Chiang Lin. "A Hybrid Machine Learning Framework for Stock Market Forecasting." *Neural Computing and Applications* 31, no. 10 (2019): 7071–7085. https://doi.org/10.1007/s00521-018-3568-5.

[3] Fischer, Thomas, and Christopher Krauss. "Deep Learning with Long Short-Term Memory Networks for Financial Market Predictions." *European Journal of Operational Research* 270, no. 2 (2018): 654–669. https://doi.org/10.1016/j.ejor.2017.11.054.

[4] Hoseinzade, Esmaeil, and Saeed Haratizadeh. "CNNpred: CNN-Based Stock Market Prediction Using Several Data Sources." *Expert Systems with Applications* 127 (2019): 272–285. https://doi.org/10.1016/j.eswa.2019.03.029

[5] Kim, Kyoung-jae. "Financial Time Series Forecasting Using Support Vector Machines." *Neurocomputing* 55, no. 1–2 (2003): 307–319. https://doi.org/10.1016/S0925-2312(03)00372-2.

[6] Murphy, John J. *Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications.* New York: New York Institute of Finance, 1999.

[7] Patel, Jigar, Sahil Shah, Priyank Thakkar, and K.K. Kotecha. "Predicting Stock Market Index Using Fusion of Machine Learning Techniques." *Expert Systems with Applications* 42, no. 4 (2015): 2162–2172. https://doi.org/10.1016/j.eswa.2014.10.031.

[8] Tsai, Chih-Fong, and Yu-Hsin Hsiao. "Combining Multiple Feature Selection Methods for Stock Prediction: Union, Intersection, and Multi-Intersection Approaches." *Decision Support Systems* 50, no. 1 (2010): 258–269. https://doi.org/10.1016/j.dss.2010.08.028.

[9] Carlson, Nicholas. 2012. "Two Charts Show Why Apple's Stock Dropped." *Business Insider*, February 14, 2012. https://www.businessinsider.com.

[10] Gandel, Stephen. 2012. "Even at $500, Apple Is Still Cheap." CNN Money: The Buzz, February 14, 2012. https://money.cnn.com.