TechJapan: Deepcraft Trainee Assignment

NTT Stock Price Prediction Model

Shashank Shekhar Asthana[1]

[1]**Department of Computer Science and Engineering**
[2]**Indian Institute of Technology, Jodhpur**

October 9, 2024

Abstract

In the volatile realm of financial markets, accurate prediction of stock prices represents a quintessential challenge and opportunity for both academics and practitioners. This research delves into the predictive capabilities of several advanced machine learning models ARIMA, LSTM, GRU, and XGBoost within the context of the Japanese stock market. Initially, comprehensive exploratory data analysis (EDA) was conducted to discern underlying patterns and inform the feature engineering process. Subsequent model development focused on optimizing and comparing each model's performance based on their ability to forecast stock prices accurately. The study's findings reveal that while traditional models like ARIMA provide a baseline, ensemble methods such as XGBoost significantly outperform in terms of prediction accuracy, as evidenced by the lowest Root Mean Squared Error (RMSE). These results underscore the potential of using advanced machine learning techniques in enhancing stock market analysis and investment strategies. The implications extend to the development of more robust trading systems and improved economic forecasting models, demonstrating a clear benefit to financial market analytics.

1 Problem Statement

Accurate stock price prediction remains a cornerstone challenge in financial economics, crucial for investment strategy formulation and risk management. Traditional forecasting methods often fall short in capturing the complex, nonlinear patterns inherent in stock price movements, particularly in dynamic markets like Japan's. This study addresses the need for more sophisticated, data-driven approaches by examining the efficacy of several machine learning models.The goal is to identify a model that not only outperforms traditional benchmarks in terms of accuracy but also adapts to the volatility and unpredictability typical of financial markets.

1.1 Data Acquisition and Analysis

Data for this study was sourced from the Tokyo Stock Exchange which is the NTT stock price data, featuring a decade of daily stock price metrics such as open, close, highs, lows, and volume. Initial data preprocessing included normalization, handling of missing values, and temporal formatting. Extensive exploratory data analysis helped uncover trends, seasonal patterns, and inform feature engineering for predictive modeling.

1.2 Strategy Development

The strategy for developing predictive models involved a systematic approach, starting with the selection of robust machine learning techniques suited for time-series forecasting. ARIMA provided a statistical baseline, while advanced models like LSTM, GRU, and XGBoost were implemented to harness their respective strengths in handling complex dependencies and non-linear patterns. The development process included tuning hyperparameters, employing cross-validation to prevent overfitting, and leveraging feature importance analysis to refine model inputs. The ultimate aim was to develop a strategy that not only enhanced prediction accuracy but also provided actionable insights into market dynamics.

1.3 Backtesting and Evaluation

Backtesting was meticulously conducted to assess the models against historical stock price data, ensuring their robustness and reliability in forecasting. The models were evaluated using

Root Mean Squared Error (RMSE), with **XGBoost demonstrating outstanding performance, achieving an RMSE of 1.061**. Additional metrics, Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE), were also employed to provide a multidimensional view of model accuracy. These evaluations confirmed XGBoost's superior capability in managing the complexities and volatilities inherent in financial markets.

1.4 Optimization

In the last stage, we optimize and refine our strategies based on backtesting results to properly balance the future predictions.

2 Data Analysis

2.1 Statistical Analysis

Statistical analysis was integral to the exploratory data analysis phase of the project, ensuring a thorough understanding of the dataset's characteristics before model development. Descriptive statistics provided initial insights into the central tendencies, variability, and distribution of stock prices and trading volumes. Histograms and box plots were utilized to visualize the data distributions and identify any outliers or anomalies. Correlation matrices were generated to examine the relationships between different financial indicators, highlighting potential predictors for stock price movements. Additionally, time-series decomposition techniques were applied to observe underlying trends, seasonal patterns, and residuals within the stock price data. This comprehensive statistical examination helped in crafting a robust feature set and informed the subsequent modeling strategy, ensuring that the data-driven insights were grounded in statistical rigor.

2.2 Volatility Analysis

Volatility is defined as how much variance is there in values of a particular quantity which has been displayed in Figure 1.

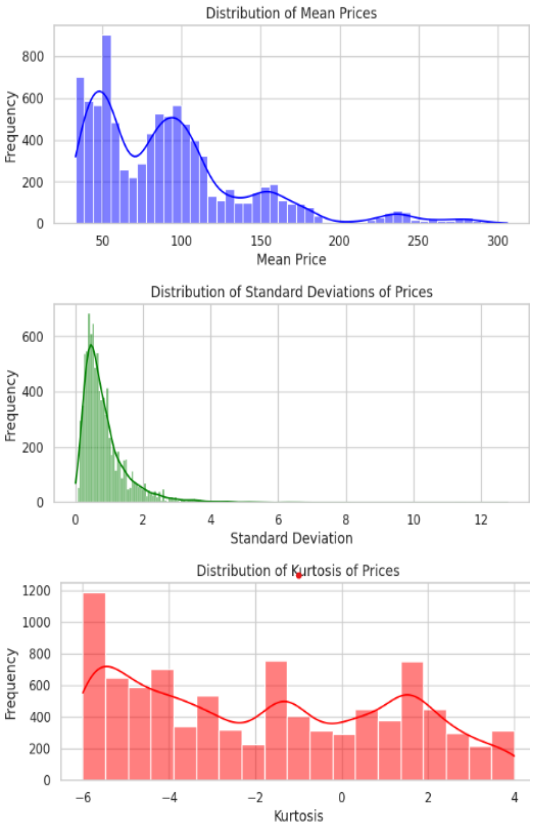

Figure 1: Statistical Analysis

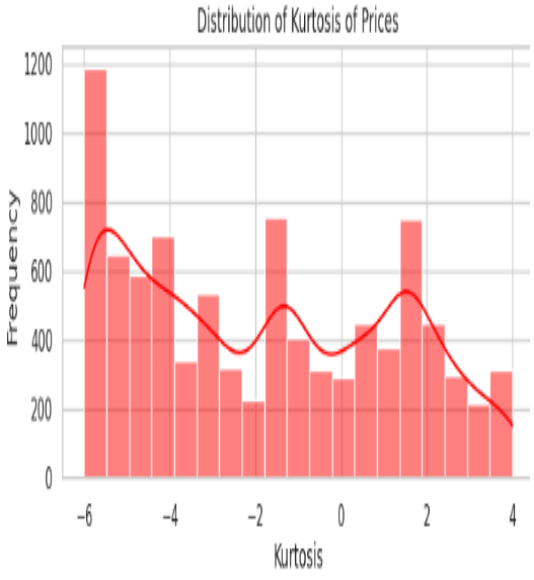

Figure 2: Kurtosis of Prices

3 Strategy Design

3.1 ARIMA

In our comprehensive strategy to predict stock prices, the ARIMA model was selected as the initial approach due to its proven track record in time-series analysis. ARIMA, which stands for Autoregressive Integrated Moving Average, models time series data by explaining a given time series based on its own past values, which includes lagged variables (AR), the differencing of raw observations to make the series stationary (I), and the lagged forecast errors (MA).

To enhance the model's capability, we incor-

porated lagging and moving average features during the statistical analysis phase. **Specifically, we created several lagged features (e.g., lag of 1 day, lag of 2 days) to capture the immediate past influences, and moving averages (e.g., 7-day, 14-day moving averages) to smooth out short-term fluctuations and reveal underlying trends in the stock prices.** These features are crucial for capturing temporal dynamics, which the ARIMA model could leverage for more accurate forecasting.

The suitability of ARIMA for our dataset was confirmed through the Augmented Dickey-Fuller (ADF) test, which ensured the stationarity of the time series—a critical assumption for the effective application of ARIMA. The results indicated that the stock prices were already stationary, hence no differencing was required (d=0). Additionally, the Autocorrelation Function (ACF) and the Partial Autocorrelation Function (PACF) plots were used to determine the order of the ARIMA model, leading to a configuration of (p=1, d=0, q=0).

Thought Process - The decision to use the ARIMA model as a starting point in our stock price prediction strategy stemmed from its reputation as a robust statistical method for time series forecasting. Understanding the linear relationships and seasonal patterns within the financial data was critical, and ARIMA offered a straightforward approach. We focused on employing ARIMA's components—autoregression to capture the influence of previous time steps, integration to ensure stationarity, and moving averages to smooth out past noise. By integrating lagged and moving average features, we aimed to enhance the model's sensitivity to recent trends and reduce the noise in daily price movements, setting a solid foundation for preliminary forecasts.

Results : Upon implementing the ARIMA model with parameters (p=1, d=0, q=0), based on our detailed ACF and PACF analysis, the model achieved an **RMSE of 80.44.** This result provided a valuable benchmark for evaluating its effectiveness compared to more complex models. It demonstrated a moderate capability in capturing the underlying patterns in stock price movements, successfully reflecting simpler trends and cyclicality in the data.

Shortcomings: ARIMA inherently assumes that the future values of a series are a linear function of past data points and residual errors. This assumption fails to capture the non-linear relationships that are often present in financial markets, where stock prices are influenced by a myriad of interdependent factors. While the RMSE value also is not as per the mark and can be displayed in the figure.

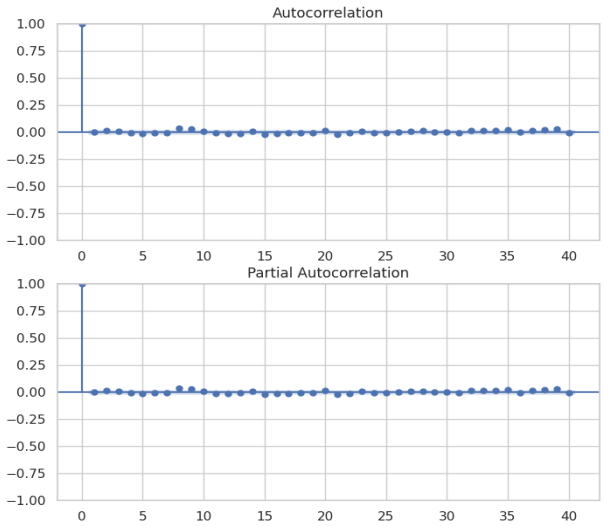

Figure 3: Displaying the Autocorrelation and Partial Autocorrelation using ACF and PACF

3.2 LSTM

Long Short-Term Memory (LSTM) networks, a type of recurrent neural network (RNN), were chosen to address the shortcomings of the ARIMA model, particularly its inability to capture long-term dependencies and non-linear patterns in stock price data. LSTMs are well-suited for financial time series analysis due to their architecture, which allows them to remember information for long periods, an essential feature for predicting the sequential data of stock prices.

The LSTM model was designed with multiple layers to enhance its ability to learn complex patterns:

- **Data Preparation**: The stock price data was first normalized to aid in the training process and reshaped into sequences that match the LSTM's input structure, fostering the model's ability to learn from sequences effectively.
- **Model Architecture**: The model consisted of several LSTM layers stacked together to deepen the learning capability, interspersed with Dropout layers to prevent overfitting. This setup was aimed at enhancing the model's ability to understand deeper patterns without memorizing the noise in the training data.

- **Training**: The LSTM model was trained using a backpropagation through time (BPTT) algorithm, with a mean squared error loss function and an Adam optimizer for efficient convergence.

Results :
The LSTM model demonstrated a significant improvement over ARIMA, evidenced by a lower RMSE value of 4.3412, which indicated enhanced predictive accuracy. This model was particularly effective in capturing the temporal dynamics and volatility of stock prices, adapting well to the inherent nonlinearities of financial markets.

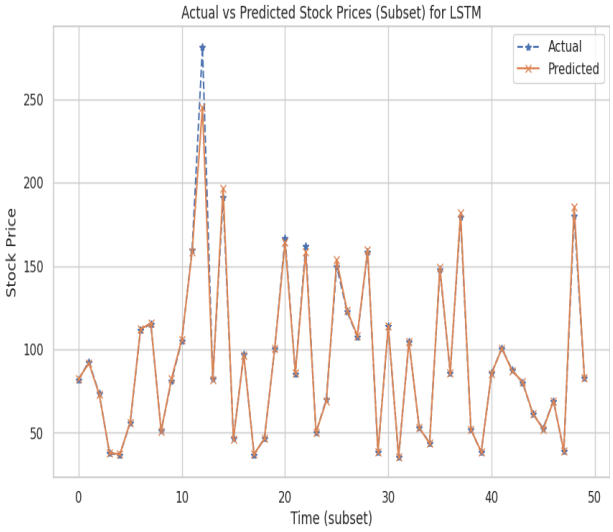

Figure 4: Actual v/s Predicted Stock prices using LSTM for Subset

Shortcomings: While the LSTM model marked a substantial advancement in our predictive capabilities, it also came with its set of challenges:

- **Complexity and Computation**: LSTMs are computationally intensive due to their complex structures and the large number of parameters that need to be trained. This complexity can lead to longer training times and demands more computational resources.
- **Overfitting Risks**: Despite the use of Dropout layers, the risk of overfitting remains if the model is excessively trained on the historical data, which can degrade its performance on unseen data.
- **Sensitivity to Parameter Settings**: The performance of LSTM models is highly sensitive to the configuration of hyperparameters such as the number of layers, the number of neurons per layer, and the learning rate. Finding the optimal set of parameters requires extensive testing and validation.

We can observe that LSTM is performing far better than ARIMA with RMSE value of 4.3412 while the graph plot displays that on few values actual deviates from the predicted.

3.3 GRU

Gated Recurrent Unit (GRU) models were selected to build upon the advancements made with LSTMs, aiming to further refine the prediction of stock prices with a simpler and potentially more efficient architecture. GRUs, similar to LSTMs, are designed to solve the vanishing gradient problem in traditional RNNs but with fewer parameters and a more streamlined architecture that can lead to faster training times and better generalization under certain conditions.

The implementation of the GRU model was carefully planned:

- **Data Preparation**: As with LSTM, the data for the GRU model was normalized and reshaped into an appropriate format that facilitates learning from sequences, which is critical for capturing temporal dependencies in stock price movements.
- **Model Architecture**: The GRU network included multiple **GRU layers with 64 units neurons in the first layer** which allow the model to effectively capture information across many time steps without the need for the memory cell used in LSTMs. This simplification potentially reduces the risk of overfitting while maintaining the ability to process long sequences of data.
- **Training**: The model was trained with a focus on optimizing performance through hyperparameter tuning, using grid search techniques to find the ideal settings for parameters such as the number of GRU units, batch size, and learning rate.

Results: : The GRU model demonstrated strong performance, **often comparable to or slightly better than the LSTM in certain metrics, with a notably lower RMSE of 2.2471 indicating improved prediction accuracy.** Its streamlined architecture allowed for quicker training cycles, which was beneficial given the extensive dataset. **The result obtained is highly accurate and performing**

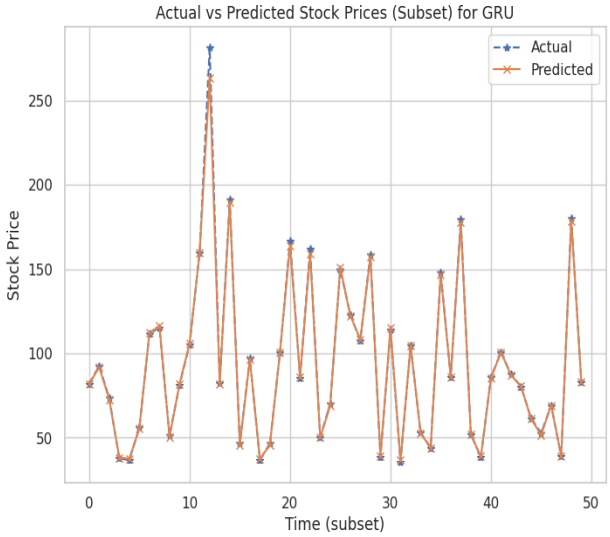

Figure 5: Actual v/s Predicted Stock Prices for GRU on a subset of prices

really good on the testing data while is only deviating on few points.

3.4 XGBOOST

XGBoost (eXtreme Gradient Boosting) was incorporated into our predictive modeling strategy as a powerful ensemble machine learning algorithm that uses a gradient boosting framework. Recognized for its efficiency, scalability, and performance, XGBoost is particularly adept at handling non-linear relationships and interactions between variables, which are common in financial markets. This model was chosen to complement the sequential models (LSTM and GRU) by offering a different approach that focuses on structured learning from tabular data.

The implementation involved several key steps:

- **Data Preparation**: Unlike the sequence-based models, XGBoost required a tabular format with engineered features. We derived a set of features based on historical price data, including lagged values, moving averages, and volatility measures, all designed to encapsulate the market dynamics effectively.
- **Model Architecture**: XGBoost was configured with multiple trees (boosted trees), where each tree was built in a way that it learned from the errors of its predecessors, thus improving the model's accuracy incrementally with each iteration.
- **Hyperparameter Tuning**: To optimize the XGBoost model, extensive hyperparameter tuning was performed. Parameters such as the learning rate, number of trees, depth of trees, and regularization terms were meticulously adjusted through a grid search approach to find the best combination that minimizes prediction error.

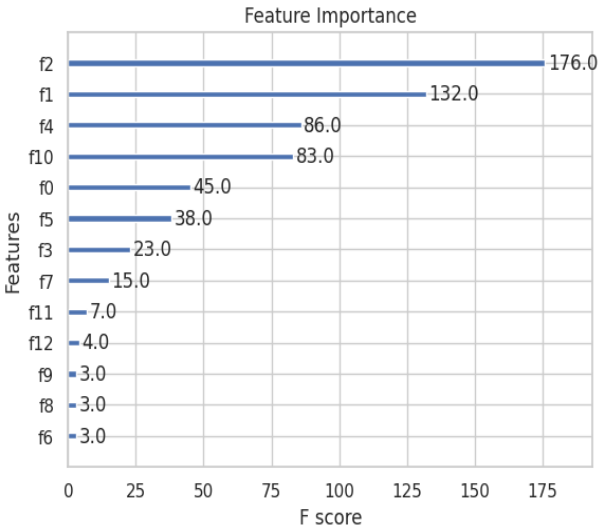

Figure 6: Feature Importance plot for XGBOOST model

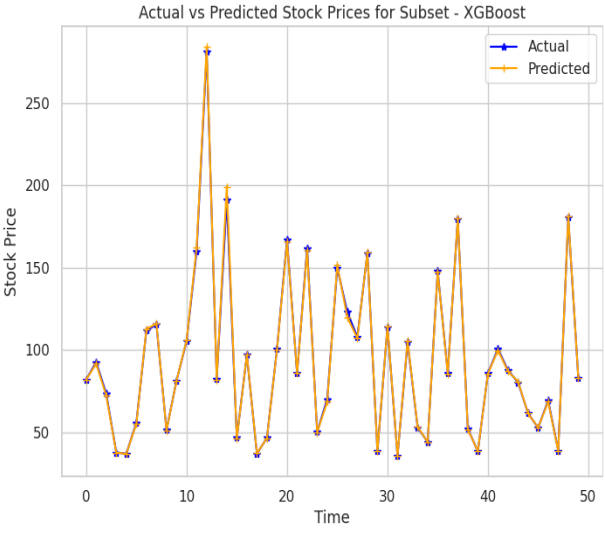

Figure 7: Actual v/s Predicted Stock Prices for XGBOOST on a subset of prices

Results: : **XGBoost demonstrated exceptional performance, achieving the lowest RMSE value of 1.0617 all the models tested,** which was a significant improvement over both traditional models like ARIMA and more complex neural network models like LSTM and GRU. This highlighted its robustness in handling the complexities of stock price prediction. **While the graph plot for XGBOOST displays that stock prices of ac-**

tual v/s predicted are nearly same, hence the model has displayed the best performance.

Insights and Shortcomings: **Model Strengths**: The strength of XGBoost in this project lay in its ability to model complex patterns through its ensemble approach, effectively reducing bias and variance. **Computational Efficiency**: Despite its powerful capabilities, XGBoost is relatively efficient computationally, especially compared to deep learning models, making it suitable for scenarios where computational resources are a concern. **Shortcomings**: While XGBoost performed exceedingly well, it requires careful handling of overfitting, especially as the number of trees increases. Unlike LSTM and GRU, XGBoost does not inherently process temporal dependencies unless specifically engineered within the features.

4 Conclusion

4.1 Comprehensive Overview of Prediction Strategies

This research project undertook a comprehensive analysis of several advanced predictive models to determine the most effective approach for forecasting stock prices in the financial markets, specifically focusing on the Japanese stock market. Here's a summary of each strategy and the overall conclusion of the study:

ARIMA : The ARIMA model served as a baseline, utilizing its traditional time-series forecasting capabilities to provide initial insights into stock price trends. While it effectively captured linear relationships and some seasonal patterns, its RMSE of 80.44 highlighted limitations, particularly in handling non-linear complexities and volatile market conditions inherent in financial data.

LSTM: The LSTM model introduced a more sophisticated approach to capture temporal dependencies and non-linear patterns that ARIMA could not. It significantly improved the prediction accuracy, lowering the RMSE and demonstrating the importance of considering long-term dependencies in stock price movements. However, its computational demands and sensitivity to parameter settings posed challenges.

GRU: GRU further refined the LSTM approach by simplifying the model architecture, which enhanced training efficiency without a substantial sacrifice in performance. It proved slightly better or comparable to LSTM in certain scenarios, especially where quicker model training was beneficial. Like LSTM, GRU was effective in handling sequence prediction but required careful tuning and validation.

XGBOOST: XGBoost emerged as the top performer with the lowest RMSE, showcasing its robustness in handling diverse and complex datasets. Its ability to capture intricate patterns through an ensemble of decision trees and provide interpretable results on feature importance made it exceptionally valuable for this application. Despite its strengths, XGBoost required diligent management of feature engineering and overfitting.

Overall Conclusion: The comparative analysis of these models revealed that no single model consistently outperformed others across all scenarios; instead, each had its strengths and areas of applicability. XGBoost, however, stood out for its efficiency and robustness, making it highly suitable for real-time stock price forecasting in volatile markets. This study recommends a hybrid approach or ensemble techniques that could leverage the strengths of sequential models like LSTM and GRU with the structured learning capabilities of XGBoost for enhanced prediction accuracy.