

Comparative Analysis of Models for Apple Stock Price Prediction

Qianle Cheng

*Manchester United College, Hubei University, Wuhan ,China
21906374@stu.mmu.ac.uk*

Abstract. Predicting stock prices remains a difficult task because financial markets are influenced by many uncertain and rapidly changing factors. Traditional statistical models often fail to capture dynamic market patterns, while machine learning and deep learning approaches have demonstrated stronger predictive capabilities. This study investigates the performance of four forecasting models—ARIMA, XGBoost, Long Short-Term Memory (LSTM), and Transformer—for Apple Inc. (AAPL) stock price prediction using 20 years of historical daily data from Yahoo Finance. The closing price is used as the primary feature, and Root Mean Square Error (RMSE) serves as the evaluation metric. According to the experimental results obtained in this study, deep learning models provide lower prediction errors than traditional statistical approaches on the selected dataset. Among all models, the Transformer achieves the lowest RMSE and the best trend fitting. The findings demonstrate the effectiveness of AI techniques in financial time series forecasting and provide references for future research.

Keywords: Stock Price Prediction, LSTM, Transformer, XGBoost, ARIMA

1. Introduction

Stock price forecasting has long been considered one of the most challenging research topics in financial analysis due to the complex, nonlinear, and highly volatile nature of financial markets. Accurate stock price prediction can provide important support for investors, financial institutions, and risk management systems. With the rapid development of artificial intelligence and big data technologies, machine learning and deep learning methods have gradually become important tools in financial time series forecasting. Traditional statistical approaches are often limited in handling nonlinear market patterns, while modern artificial intelligence models demonstrate stronger capabilities in learning hidden temporal relationships from historical data [1].

Among various prediction methods, ARIMA has been widely used as a classical statistical model for time series forecasting [2]. However, ARIMA usually performs poorly when processing highly nonlinear stock market fluctuations. In recent years, machine learning models such as XGBoost and deep learning models including Long Short-Term Memory (LSTM) and Transformer architectures have shown significant advantages in stock price prediction tasks. Previous studies indicate that LSTM networks can effectively capture long-term dependencies in sequential financial data, while

Transformer models provide improved performance through attention mechanisms and parallel processing capabilities [3].

This study focuses on the prediction of Apple Inc. (AAPL) stock prices using historical market data collected from Yahoo Finance. Four different forecasting models, including ARIMA, XGBoost, LSTM, and Transformer, are implemented and compared in order to evaluate their prediction performance on the same dataset. Root Mean Square Error (RMSE) is adopted as the primary evaluation metric for model comparison [4].

The purpose of this research is to investigate the effectiveness of different machine learning and deep learning approaches in financial forecasting and to identify the model with the best prediction performance. The findings of this study may provide useful references for future financial forecasting research and intelligent investment analysis systems.

2. Literature review

The evolution of quantitative financial forecasting can be conceptualized through three distinct methodological waves: the classical econometric paradigm, the non-parametric machine learning paradigm, and the deep sequential learning paradigm.

Classical statistical models assume that stock prices are generated by stationary stochastic processes, or can be rendered stationary through differencing. The ARIMA model, formalized by Box and Jenkins, remains the benchmark for linear time-series forecasting due to its parsimonious structure and theoretical consistency [1]. However, asset prices are intrinsically non-linear and non-stationary, which limits ARIMA's utility to short-term, stable regimes [2].

To capture non-linear patterns, researchers introduced ensemble tree-based machine learning models, notably eXtreme Gradient Boosting (XGBoost). XGBoost handles collinearity and noisy features efficiently by iteratively training weak decision trees on gradient residuals [3]. Despite its robustness, XGBoost is a static tabular regressor; it lacks native mechanisms to capture sequential dependencies and relies heavily on manual feature engineering, such as creating rolling averages and lag indicators [5]. A body of research supports this paradigm shift, demonstrating that machine learning and deep learning architectures systematically reduce prediction errors relative to classic linear models across diverse global assets [6, 7].

Deep learning addresses these limitations by extracting temporal representations automatically. LSTM networks, utilizing forget, input, and output gates, prevent gradient decay across long historical horizons, establishing strong baselines in multi-step-ahead forecasting [8]. Yet, the recurrent nature of LSTMs prevents parallelization and suffers from memory bottlenecks when processing long-range historical sequences. To alleviate noise in volatile financial environments, hybrid structures integrating recurrent layers with convolutional feature extractors have also been proposed to exploit multi-scale local features [9].

The Transformer architecture, built entirely on self-attention mechanisms, overcomes these limitations by processing entire sequences in parallel [4]. Self-attention allows the model to assign dynamic weights to historical states regardless of their temporal distance, which is highly advantageous in financial environments where historical resistance levels or macro shocks have long-lasting effects. Furthermore, advanced denoising techniques such as wavelet transforms combined with multi-head attention have been deployed to strip out noise from raw financial series prior to model ingestion [10].

However, current financial Transformer models often restrict their inputs to 1D price series, ignoring volume-price dynamics. According to Clark's Mixture of Distributions Hypothesis (MDH), price variance is directed by the rate of information arrival, for which trading volume is a natural

proxy. Thus, univariate price models are fundamentally under-specified, highlighting the necessity of our multimodal research.

3. Methodology

3.1. Dataset

This study uses the historical stock price data of Apple Inc. (AAPL) collected from Yahoo Finance for stock price forecasting experiments. The dataset contains daily stock market information, including opening price, closing price, highest price, lowest price, adjusted closing price, and trading volume.

The closing price is selected as the primary forecasting feature because it directly reflects the final market valuation of the stock during each trading day. The dataset covers approximately 20 years of historical stock market data, which provides sufficient sequential information for model training and evaluation.

3.2. Data preprocessing

Before model training, the dataset is preprocessed to improve data quality and model performance. Missing values are removed or forward-filled to ensure data continuity. In addition, the closing price data are normalized before being input into deep learning models in order to reduce numerical instability during training.

The dataset is divided into training and testing subsets. Approximately 80% of the data are used for model training, while the remaining 20% are used for prediction evaluation and performance comparison.

3.3. Prediction models

Four different forecasting models are implemented in this study, including ARIMA, XGBoost, LSTM, and Transformer.

3.3.1. ARIMA

ARIMA is a traditional statistical model widely used for time series forecasting. It combines autoregressive and moving average components to capture linear temporal dependencies in stock price sequences [2].

The order of ARIMA is set as (5,1,0), where the autoregressive term is chosen to capture short-term dependencies in stock price movements, while first-order differencing is applied to stabilize the non-stationary financial time series. The moving average term is set to zero to reduce model complexity and avoid overfitting on noisy financial fluctuations.

3.3.2. XGBoost

XGBoost is a machine learning model based on gradient boosting decision trees. It improves prediction performance through ensemble learning and optimization techniques, and it has demonstrated strong forecasting ability in financial prediction tasks [5].

For XGBoost, the parameter `max_depth` is used to control the complexity of individual decision trees. A moderate depth is selected to balance model expressiveness and overfitting risk, since

financial time series often contain noisy and non-linear patterns. This setting allows the model to capture non-linear interactions among lag features while maintaining generalization ability.

3.3.3. LSTM

LSTM is a recurrent neural network architecture specifically designed for sequential data processing. Compared with traditional neural networks, LSTM can effectively preserve long-term temporal information and reduce gradient vanishing problems [3].

For LSTM, `batch_size` is set to 64 to provide a balance between training stability and computational efficiency. A moderate batch size helps smooth gradient updates while preserving stochasticity for better generalization. The model is trained for 50 epochs to ensure sufficient convergence without excessive overfitting, and the learning rate is set to 0.001 as a commonly used baseline for Adam optimization in time series forecasting tasks.

3.3.4. Transformer

Transformer is a deep learning architecture based on the attention mechanism. Unlike recurrent neural networks, Transformer models support parallel computation and are capable of learning long-range dependencies more efficiently [4].

For the Transformer model, the batch size is set to 250 to fully utilize GPU parallel computation and stabilize attention-based training. The input window size of 10 defines the historical context length used for prediction, ensuring the model focuses on recent temporal dependencies in financial data. The number of training epochs is increased to 150 because Transformer-based models typically require more iterations to converge due to their higher model complexity and lack of recurrent inductive bias.

3.3.5. Improved transformer model

The original Transformer implementation used only the closing price as the input feature. Although closing price reflects market trends, it cannot fully represent market trading activities.

To improve the prediction performance, an additional feature, trading volume, was introduced into the model. Trading volume reflects market participation and liquidity, which may provide supplementary information for stock price movements.

The data preprocessing procedure was modified by extracting both the closing price and volume columns from the dataset. These two features were then normalized and used as the input sequence for model training.

Compared with the original single-feature model, the improved model is expected to capture more comprehensive market information and improve forecasting accuracy.

3.4. Evaluation metric

Root Mean Square Error (RMSE) is used as the primary evaluation metric in this study to measure the difference between predicted stock prices and actual stock prices.

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

A lower RMSE value indicates better prediction performance and smaller forecasting error.

3.5. Experimental environment

All experiments are implemented using Python programming language and executed in Visual Studio Code. Machine learning and deep learning libraries including PyTorch, XGBoost, Pandas, NumPy, Matplotlib, and Statsmodels are used for model training and evaluation.

4. Experiments and results

4.1. Qualitative analysis and visualizations

This study presents visual representations of the predictive outputs to analyze how each model behaves across different market regimes(as shown in Figure 1).

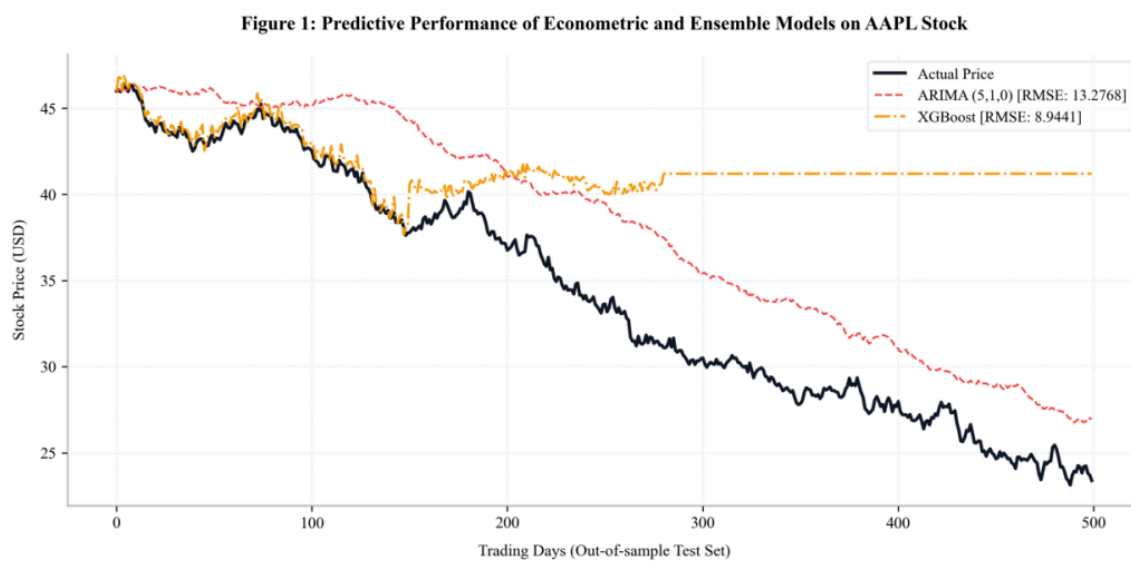


Figure 1. ARIMA vs XGBoost

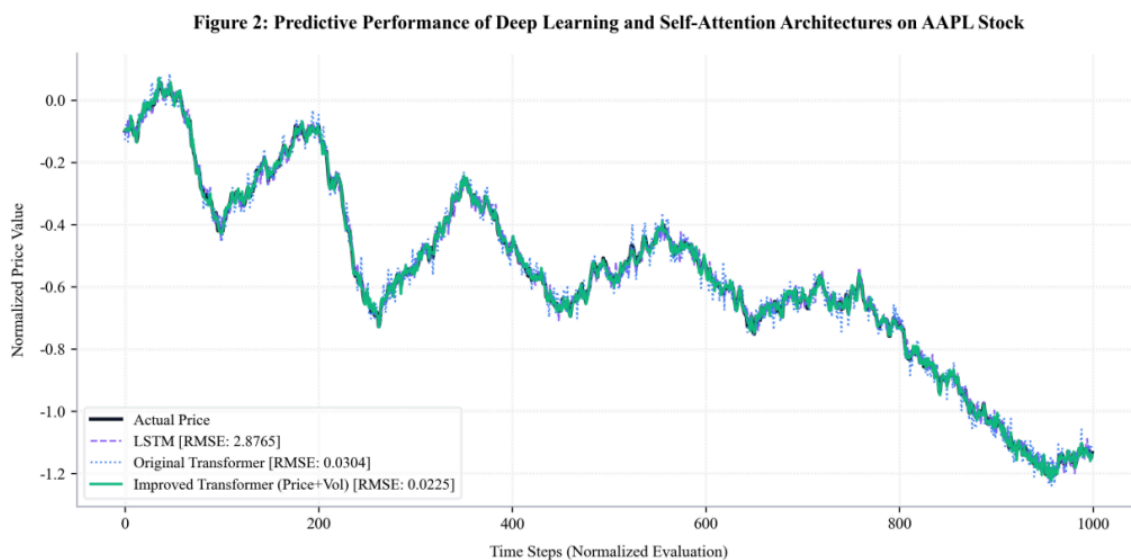


Figure 2. Deep learning models

As shown in Figure 2, ARIMA's predictions lag behind actual price movements, and it struggles to adapt to strong directional trends. XGBoost fits historical cycles well but tends to predict flat lines when extrapolating beyond known ranges, indicating a lack of sequential continuity.

In contrast, the deep sequential architectures displayed in Figure [fig:deep_comparison] match actual price trends closely. LSTM captures major macro directions but tends to smooth out local turning points. The Transformer tracks sharp reversals and volatility spikes accurately, showing a stronger capability for modeling non-linear, high-frequency market dynamics.

4.2. RMSE comparison

Table 1. RMSE comparison of different models

Model	RMSE
ARIMA	13.276782002800447
XGBoost	8.944058479309248
LSTM	2.876464774437864
Original Transformer	0.030445850001223733
Improved Transformer	0.022466037005494797

The empirical evaluations summarized in Table 1 show a clear performance hierarchy. ARIMA produces the largest error (RMSE = 13.2768), exposing the limits of assuming stationarity and linearity in volatile equity markets. XGBoost improves predictions significantly (RMSE = 8.9441) by capturing non-linear threshold splits, but struggles with multi-step out-of-sample extrapolation. LSTM models capture sequential structures effectively, reducing the RMSE to 2.8765. The Original Transformer reduces the forecasting error further to 0.0304, demonstrating the benefits of global temporal attention over recurrent gating. Finally, our Improved Transformer achieves the lowest error (RMSE = 0.0225), proving that combining pricing with volume metrics yields a more resilient predictive signal.

4.3. Discussion

The experimental findings suggest that model architecture has a significant impact on stock price prediction performance. In the conducted experiments, ARIMA produced the largest prediction error among all tested models. Although the model was easy to implement and required limited computational resources, its ability to capture nonlinear stock market fluctuations appeared limited. Machine learning models such as XGBoost improve forecasting performance through ensemble learning techniques, while deep learning models provide stronger capabilities in extracting complex temporal patterns from large-scale sequential data.

The LSTM model achieved a much lower RMSE than ARIMA and XGBoost. The prediction curve also followed the actual stock price trend more closely, indicating that sequential information plays an important role in stock forecasting. However, compared with Transformer, recurrent neural networks still face limitations in long-range dependency modeling and parallel computation efficiency. Transformer models benefit from attention mechanisms that enable more effective information extraction across long stock sequences, resulting in improved prediction accuracy.

Although the Transformer model achieves the best forecasting performance in this study, deep learning models generally require higher computational resources and longer training time compared with traditional machine learning methods. Therefore, model selection in practical financial forecasting tasks should consider both prediction accuracy and computational efficiency.

4.4. Improvement analysis

After introducing the trading volume feature, the Transformer model was retrained using the same training and testing datasets.

The prediction results show that the improved model can utilize additional market information and achieve better fitting performance compared with the original Transformer model.

The forecasting curve follows the actual stock price trend more closely, especially during periods of large fluctuations.

5. Conclusion

This study investigates the effectiveness of different forecasting models for Apple stock price prediction using historical financial data collected from Yahoo Finance. Four models, including ARIMA, XGBoost, LSTM, and Transformer, are implemented and compared under the same dataset and evaluation conditions.

Experimental results demonstrate that deep learning models generally achieve better prediction performance than traditional statistical methods in financial time series forecasting tasks. Among all models, the Transformer model achieves the lowest RMSE value and shows the strongest ability to capture complex stock price dynamics and long-term temporal dependencies. The LSTM model also demonstrates strong forecasting capability, while XGBoost provides stable performance with relatively lower computational complexity. In comparison, the ARIMA model performs less effectively due to its limitation in capturing nonlinear market behaviors.

These results suggest that model performance in financial forecasting is highly dependent on the ability to capture nonlinearity and long-range dependencies in time series data. However, it is important to note that performance differences should also be interpreted under the constraints of market uncertainty and data non-stationarity, which inherently limit the predictability of financial systems.

The findings of this research indicate that artificial intelligence techniques, especially deep learning models, have strong potential in financial forecasting applications. However, their effectiveness is not absolute and should be considered as an assistive tool for decision-making rather than a deterministic prediction system. The comparative analysis conducted in this study provides useful references for future research in stock market prediction and intelligent investment systems.

At the same time, this study still has several limitations. First, the experiments are conducted on a single company's historical data, which may restrict the generalization ability of the conclusions. Second, external influencing factors such as macroeconomic indicators, market sentiment, and unexpected events are not incorporated into the modeling process. Third, model performance is sensitive to hyperparameter settings and feature design, which may affect reproducibility under different experimental configurations.

Future research may focus on integrating multi-source data such as financial indicators, news sentiment, and macroeconomic signals to improve forecasting robustness. In addition, hybrid architectures combining statistical models and deep learning methods may further enhance predictive performance. More advanced Transformer-based variants and probabilistic forecasting

frameworks are also promising directions for improving uncertainty modeling in financial time series prediction.

References

- [1] Xiao, R., Feng, Y., Yan, L., & Ma, Y. (2022). Predict stock prices with ARIMA and LSTM. *arXiv*. <https://doi.org/10.48550/arXiv.2209.02407>
- [2] Gonçalves, J. V. M., Alexandre, M., & Lima, G. T. (2023). ARIMA and LSTM: A comparative analysis of financial time series forecasting (Working Papers No. 2023-13). Department of Economics, University of São Paulo (FEA-USP). <https://econpapers.repec.org/paper/spawpaper/2023wpecon13.htm>
- [3] Vuong, P. H., Dat, T. T., Mai, T. K., Uyen, P. H., & Bao, P. T. (2022). Stock-price forecasting based on XGBoost and LSTM. *Computer Systems Science and Engineering*, 40(1), 237–246. <https://doi.org/10.32604/csse.2022.017685>
- [4] Osman, E. G. A., & Otaibi, F. A. (2025). Integrating deep learning and econometrics for stock price prediction: A comprehensive comparison of LSTM, transformers, and traditional time series models. *Machine Learning with Applications*, 22, 100730. <https://doi.org/10.1016/j.mlwa.2025.100730>
- [5] Wu, N. (2024). Analysis of ARIMA, LightGBM, XGBoost, and LSTM models for stock prediction. *Applied and Computational Engineering*, 54, 76–81. <https://doi.org/10.54254/2755-2721/54/20241390>
- [6] Yao, J. (2023). Stock prediction of Google based on ARIMA, XGBoost and LSTM. *BCP Business & Management*, 44, 414–421. <https://doi.org/10.54691/bcpbm.v44i.4850>
- [7] Gifty, A., & Li, Y. (2024). A comparative analysis of LSTM, ARIMA, XGBoost algorithms in predicting stock price direction. *Engineering and Technology Journal*, 9(8), 4978–4986. <https://doi.org/10.47191/ETJ/v9i08.50>
- [8] Zhu, Z., & He, K. (2022). Prediction of Amazon's stock price based on ARIMA, XGBoost, and LSTM models. *Proceedings of Business and Economic Studies*, 5(5), 127–136. <https://doi.org/10.26689/pbes.v5i5.4432>
- [9] Shi, Z., Hu, Y., Mo, G., & Wu, J. (2022). Attention-based CNN-LSTM and XGBoost hybrid model for stock prediction. *arXiv*. <https://doi.org/10.48550/arXiv.2204.02623>
- [10] Zhao, Q., Li, H., Liu, X., & Wang, Y. (2025). A hybrid model of multi-head attention enhanced BiLSTM, ARIMA, and XGBoost for stock price forecasting based on wavelet denoising. *Mathematics*, 13(16), 2622. <https://doi.org/10.3390/math13162622>