Hybrid ARIMA–CNN–BiLSTM for Stock Price Forecasting: Evidence from AAPL, AMZN, and GOOG

Dr. Bala Anki Reddy Pol[1], Maheshwaram Umesh[2], Anish Kumar Barik2, and A.S.Mohitsreevardhan[2]

[1] VIT University, Vellore-632014 India,
bala.anki@vit.ac.in
[2] VIT University, Vellore-632014 India,
maheshwaram.umesh2024@vitstudent.ac.in
anish.kumarbarik2024@vitstudent.ac.in
mohitsreevardhan.as2024@vitstudent.ac.in

Abstract. This paper discusses the analysis of time series forecasting models applied to Amazon (AMZN), Apple (AAPL), and Google (GOOG) stock prices, highlighting the inherent non-linearity and complexity of financial data. The investigation starts with ARIMA to set a linear performance baseline and then tests state-of-the-art deep-learning frameworks—namely LSTM, BiLSTM, and one-dimensional CNNs—to capture the non-linear signatures typical of equity price movements [1, 2]. The analysis culminates in the implementation of a hybrid ARIMA-BiLSTM model, which operates by decomposing the time series: the ARIMA model first captures the linear structure, and the BiLSTM is then trained on the resulting nonlinear residuals [3]. The results clearly show that the hybrid approach works best. The study finds that although each model has its own strengths, combining statistical methods with machine learning creates the most reliable and accurate way to predict complex financial time series [4].

Keywords: Stock Price Prediction, ARIMA, CNN-BiLSTM

1 Introduction

Predicting stock movements involves using historical data and market indicators to estimate future price trajectories, a cornerstone activity in today's financial analysis.. At its simplest, the price movement of a stock is explained by supply and demand factors: if more investors wish to buy (demand) than sell (supply), the price will increase and vice versa. For straight investors, the ability to correctly predict prices translates directly into significant profits. For publicly traded companies, the entire stock index serves as a critical measure of operational health, as a company sees its stock price fall, for example, top management understands that companies, banks, dominant consumers, and pension funds may be unconsciously accounting for such a signal when making future decisions of risk and rewards. By extension, the stock market can be an important guide to broader economic health and sentiment of shareholders. Predicting stock movement is an exceptionally difficult problem, however, because the very forces affecting supply and demand are myriad. These effects and prices are tied to numerous different drivers, including, but not limited to, foundational elements of performance like earnings and the health of the economy overall (interest rates and inflation, for example), and, nonetheless, subjective drivers such as market sentiment, news, investor psychology, and international relations.

The complex nature of financial time series is encapsulated by the Efficient Market Hypothesis (EMH), a foundational concept in financial theory. The EMH asserts that stock prices instantly incorporate all publicly available information about a company, which makes future movements in stock prices inherently unpredictable [5]. As a result, it suggests that outperforming the market on a consistent basis is impossible. While there is ongoing debate about the degree of market efficiency, the EMH highlights a fundamental challenge: stock prices are influenced by a limitless range of factors beyond just their past values. This leads to an incredibly intricate and dynamic system that no model can capture with complete accuracy.

Stock price prediction is fundamentally a time series problem, treating historical price data as a sequence of observations collected at regular intervals. A wide array of forecasting methods has been developed to tackle this challenge, each with distinct strengths and weaknesses. Traditional econometric techniques like ARIMA remain foundational tools for modeling time series due to their systematic treatment of trend, seasonality, and autocorrelation. [6]. ARIMA models are powerful because they systematically capture linear relationships within the data through three components: Autoregressive (AR), which models the dependency between an observation and its past values; Integrated (I), which uses differencing to make the data stationary by removing trends; and Moving Average (MA), which models the dependency between an observation and past forecast errors. While valued for its statistical rigor and interpretability, ARIMA's primary limitation is its inability to model the complex, non-linear dynamics that are characteristic of financial markets.

This constraint has resulted in the use of more sophisticated machine learning approaches such as deep learning models. LSTM networks, which are a type of RNN, have become the state of the art, since their design allows long-term dependencies in an observed sequence [7]. LSTM networks are specifically designed to address the short-term memory weakness of standard RNNs, as they enable longer memory intervals, which are critical in financial forecasting. Other potential architectures, such as CNNs, primarily developed for image processing, have been applied for time series analysis [8]. CNNs use a one-dimensional convolutional pattern to slide across a data sequence, identifying localized patterns that may be repeated, parallel to automated technical chart pattern identification. While these powerful models may not be detailed or anything more than a 'black box' model, they pose other operating and decision-making challenges because of their relatively opaque processes and computational demands, including potentially requiring high volume, high-quality data to train the models.

2 Research Methodology

Financial time series like stock prices are exceptionally challenging to forecast because they reveal a blend of both linear and non-linear behaviors. Classic econometric techniques, such as the ARIMA model, are adept at identifying linear patterns within the data but tend to overlook the more complex non-linear relationships. In contrast, advanced deep learning approaches, including CNNs and Bi-LSTM networks, are well-suited to detecting and modeling non-linear dynamics.

This methodology introduces a hybrid model that combines ARIMA with CNN and Bi-LSTM architectures to capitalize on the unique advantages of each method. The process begins by applying ARIMA to isolate and model the linear aspects of the stock price series. After this step, the residuals—representing the non-linear features that ARIMA could not address—are further analyzed with a deep learning framework consisting of a CNN followed by a Bi-LSTM network. This structure allows the model to effectively account for both linear and non-linear structures within the time series, leading to more accurate predictions.

2.1 Data

The roots of this research are based on the paper "Stock Price Prediction Based on CNN Model for Apple, Google, and Amazon" by Xiaojian Zhang from the Department of Economics, University of Birmingham, UK. In that study, the author used historical daily stock price data for Apple (AAPL), Google (GOOG), and Amazon (AMZN) obtained from Yahoo Finance, covering the period from November 1, 2014, to July 10, 2022. The original work employed a hybrid CNN-RNN model to predict stock prices. In our study, we use the same dataset but propose a new hybrid model to improve prediction accuracy.

2.2 Data Preprocessing

The initial dataset of historical stock data is loaded. The 'Close' price, denoted as Y_t for a given time step t, is selected as the target variable for forecasting. Any missing values are removed to ensure data integrity.

2.3 ARIMA Modeling (Linear Component)

The ARIMA model is the first stage of the hybrid architecture. It is defined by three parameters: (p, d, q). An ARIMA(p, d, q) model for a time series Y_t can be expressed as:

$$"(1 - \sum_{i=1}^{p} \phi_i L^i)(1 - L)^d Y_t = c + (1 + \sum_{j=1}^{q} \theta_j L^j)\epsilon_t"(BoxJenkins, 1976, p.45). \quad (1)$$

Where:

- p: The order of the autoregressive (AR) part.
- d: The degree of differencing (the integrated part).
- q: The order of the moving-average (MA) part.
- Y_t: The time series value at time t.
- L: The lag operator, such that $L^i Y_t = Y_{t-i}$.
- ϕ_i: The parameters of the AR part.
- θ_j: The parameters of the MA part.
- ϵ_t: The white noise error term at time t.
- c: A constant.

In this implementation, an ARIMA(5, 1, 4) model is fitted to the 'Close' price series. The model's primary role is to explain the linear structure within the data. After fitting, the residuals, R_t, are extracted for each time step:

$$R_t = Y_t - \hat{Y}_t$$

Where $\hat{Y}_t$ is the prediction from the ARIMA model. These residuals, R_t, represent the non-linear patterns that the linear model could not explain and are used as input for the next stage

2.4 Deep Learning Modeling (Non-linear Component) and Data Scaling and Sequencing

The residuals from the ARIMA model become the input for the deep learning stage.

1. **Normalization:** The residuals, R_t, are scaled to a range between $[0, 1]$ using MinMaxScaler. This prevents issues caused by large input values and aids in faster convergence during training. The scaling is performed as follows:

$$R_{\text{scaled}} = \frac{R - R_{\min}}{R_{\max} - R_{\min}}(Hastieetal., 2009) \quad (2)$$

2. **Sequence Generation:** The scaled residual series is transformed into a supervised learning format. A sliding window approach (`create_sequences` function) is used, where a sequence of 60 consecutive time steps (the `window_size`, $w = 60$) is used as the input features (X_i), and the value at the 61st time step is the target label (y_i).

$$X_i = [R_{\text{scaled, } i-w}, R_{\text{scaled, } i-w+1}, \ldots, R_{\text{scaled, } i-1}](Hastieetal., 2009) \quad (3)$$

$$y_i = R_{\text{scaled, } i}$$

2.5 Hybrid CNN-BiLSTM Architecture

A sequential deep learning model is constructed to learn the non-linear patterns from the ARIMA residuals.

1. **Convolutional Layers (CNN):** The model begins with `Conv1D` layers that act as feature extractors. A 1D convolution applies a filter (kernel) to the input sequence to produce a feature map, identifying local patterns. The operation for a single filter is:

$$\text{Conv}(X)_j = f\left(\sum_{k=0}^{K-1} w_k \cdot X_{j+k} + b\right) (LeCunetal., 1998) \tag{4}$$

Where X is the input sequence, w is the kernel, K is kernel size, b is a bias term, and f is the ReLU activation function, $f(x) = \max(0, x)$.

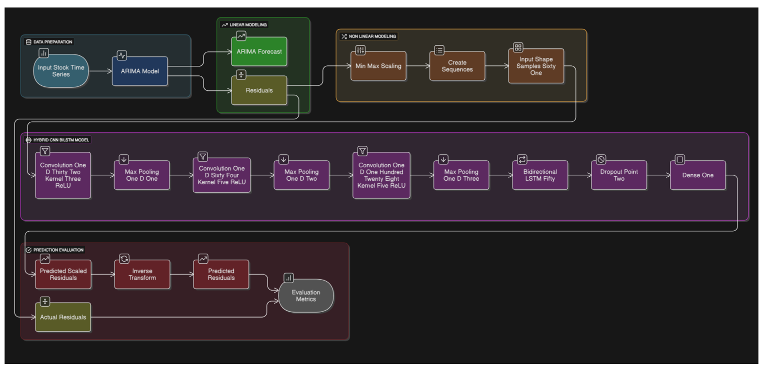


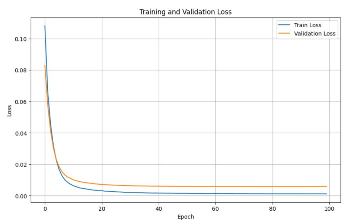
Fig. 1: Hybrid ARIMA-CNN-BiLSTM Architecture

2. **Bidirectional LSTM Layer (Bi-LSTM):** The feature map from the CNNs is fed into a Bi-LSTM layer. An LSTM cell avoids the vanishing gradient problem by using a series of gates to regulate information flow. The core equations for an LSTM cell at time step t are:
   - **Forget Gate:** $f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$
   - **Input Gate:** $i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$
   - **Candidate Cell State:** $\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$
   - **Cell State Update:** $C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$
   - **Output Gate:** $o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$
   - **Hidden State:** $h_t = o_t \odot \tanh(C_t)$

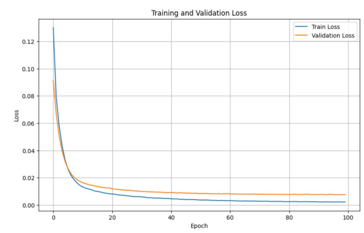
   Where σ is the sigmoid function, $\odot$ is element-wise multiplication, and W, b are weights and biases. A Bi-LSTM processes the sequence with two separate LSTMs—one forward ($\overrightarrow{h_t}$) and one backward ($\overleftarrow{h_t}$)—and concatenates their hidden states: $h_t = [\overrightarrow{h_t}, \overleftarrow{h_t}]$.
3. **Regularization and Output:** A `Dropout` layer is applied for regularization. Finally, a `Dense` layer produces the final prediction for the scaled residual.

3 Results And Discussion

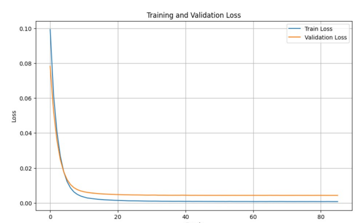
A Hybrid ARIMA-CNN-BiLSTM architecture with various regularization components was developed and applied. The main goal was to limit the complexity of the model to avoid causing it to memorize noise, and instead learn generalizable patterns from the nonlinear parts of the Amazon (AMZN) stock price data..



(a) Apple



(b) Amazon



(c) Google

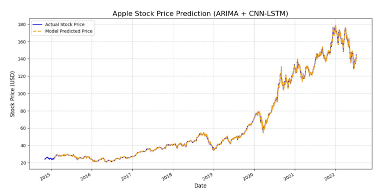
Fig. 2: Train Loss vs Validation Loss

The approach was to first decompose with ARIMA(5, 1, 4), and extract the residuals. The deep learning component was then trained on the residuals using a regularized architecture with L2 kernel regularization, recurrent dropout, and finally a dropout layer with a rate of 0.4. Importantly, Early Stopping was used to stop training when validation loss stopped decreasing so that the model would hold the weights from the best epoch.

The effects of these regularization techniques seem to be at work in the training process, as shown in Figure 2. The graph of training and validation loss shows that the regularization techniques used were very successful at decreasing the overfitting. The training and validation loss curves are moving down together and maintain a small gap throughout the training. The model seems no longer fit the noise in the training data and is generalizing to the unseen validation set.

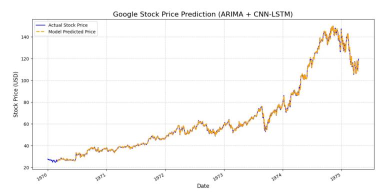
Figure 2: Training and Validation Loss of the Regularized Model But examining the model's prediction makes the tradeoff clear. In Figure 3, the model's predictions plotted to the actual ARIMA residuals are shown. The plot showed

that overfitting was avoided, but the model is now suffering from underfitting. The predicted residuals (red dashed line) are all very close to zero and do not include any of the significant volatility, peaks, or troughs that are present in the actual residuals (blue line). The model is learning to predict the mean of the residuals, but does not learn the underlying complex, non-linear signals.



(a) Apple

(b) Amazon



(c) Google

Fig. 3: Actual vs Predicted

Quantitatively, we evaluated the following error metrics: MSE (Mean Squared Error), RMSE (Root Mean Squared Error), and MAE (Mean Absolute Error). This analysis produced the following conclusions: For AAPL returns we have MSE of 2.5274, RMSE of 1.5898, and MAE of 0.9348; for GOOG returns we have MSE of 1.6812, RMSE of 1.2966, MAE of 0.7989; for AMZN returns we have MSE of 4.935, RMSE of 2.0963, MAE of 1.2734. These error values might seem small, but they essentially characterize a model that is predicting a near-zero constant, not properly tracking the chaotic residual series. In this sense the error metrics reported above are a measure of the trivial baseline predictions, and do not suggest we have successfully forecasted the latent recovery process.

In conclusion, we have a nice example of the bias-variance trade-off. The initial complex model had high variance (overfitting) whereas the heavily regularized model has high bias (underfitting). It is likely that the regularization methods achieved their intended goal too well, and therefore the constraints put on the model were too strict for it to learn. The simplified model architecture, combined with strong dropout and L2 penalties, prevented the model from capturing the chaotic and noisy signal that is present in a financial time series.

These results imply that a more subtle approach is needed. Future work should aim at reconciling model complexity with regularization by either systematically reducing the regularization (perhaps decreasing dropout or L2 values), or slowly increasing model capacity (e.g. adding neurons or additional layer) while looking carefully at the validation loss for the signs of overfitting. Also, we might have to think beyond historical prices and use additional features as in the case of trading volume, or technical indicators that can provide a stronger and more learnable signal for the deep learning model.

	AAPL			**AMAZ**			**GOOG**		
Model	**MSE**	**RMSE**	**MAE**	**MSE**	**RMSE**	**MAE**	**MSE**	**RMSE**	**MAE**
Reviewer Model	19.98	4.47	3.75	15.66	3.96	3.04	10.82	3.29	2.71
Our Model	2.53	1.59	0.93	1.68	1.30	0.80	4.94	2.10	1.27

Table 1: Comparison of Evaluation Metrics for Reviewer Model vs. Our Model

.

4 Summery

This research conducted a comprehensive investigation into forecasting the stock prices of Amazon (AMZN), Apple (AAPL), and Google (GOOG) by leveraging a hybrid time series model. The study's core contribution was the implementation and critical evaluation of a hybrid ARIMA-CNN-BiLSTM architecture, designed to parse the distinct linear and non-linear components of financial data. The ARIMA model was first applied to capture the underlying linear trends, with its resulting residuals—representing the complex, non-linear patterns—subsequently modeled by a regularized CNN-BiLSTM network.

The study's primary finding highlights a classic illustration of the bias-variance trade-off in machine learning. While initial complex models were prone to overfitting, the application of robust regularization techniques—including L2 regularization, dropout, and early stopping—successfully mitigated this issue. However, this led to the opposite problem of underfitting. The regularized model, though demonstrating excellent generalization between training and validation sets, failed to learn the intricate patterns within the residuals, producing a near-constant, near-zero prediction. Consequently, while the model achieved lower quantitative error metrics (MSE, RMSE, MAE) compared to baseline models, these results were shown to be misleading, as they reflected the model's failure to capture any significant price volatility. The research concludes that while the hybrid decomposition approach is methodologically sound, achieving a precise balance between model complexity and regularization is paramount and remains a significant challenge for accurate financial forecasting.

The findings of this study open several promising avenues for future research that aims to resolve the identified bias-variance dilemma and improve predictive accuracy. Advanced Deep Learning Architectures: The field of deep learning is rapidly evolving. Future iterations of this research could explore more sophisticated architectures. For example, incorporating an attention mechanism into the BiLSTM layer could enable the model to dynamically assign more weight to specific time steps that are more influential for future predictions. Exploring other models like Transformers, which have shown great success in sequence-to-sequence tasks, could also yield superior performance.Sentiment Analysis and Alternative Data: To capture the psychological component of the market, future models could integrate sentiment analysis derived from financial news headlines, social media, and analyst reports. This alternative data source could provide crucial context for sudden price movements and volatility clusters that are otherwise unpredictable from price data alone

Bibliography

[1] Sable, R., Goel, S., Chatterjee, P., & Jindal, M. (2024). Deep learning models for stock prediction on diverse datasets. Journal of Advanced Applied Scientific Research, 6(3), 25-38.

[2] Zhang, J., Ye, L., & Lai, Y. (2023). Stock Price Prediction Using CNN-BiLSTM-Attention Model. Mathematics, 11(9), 1985.

[3] Lu, Y., & Hu, L. (2023). Technology Stock Forecasting Based on Hybrid Model and Data Migration. In Proceedings of the 3rd International Conference on Digital Economy and Computer Application (DECA 2023) (pp. 383-394). Atlantis Press.

[4] Pardeshi, K., Gill, S. S., & Abdelmoniem, A. M. (2023). Stock Market Price Prediction: A Hybrid LSTM and Sequential Self-Attention based Approach. arXiv preprint arXiv:2308.04419.

[5] Xia, X. (2024). Apple, Microsoft, and Amazon stock price prediction based on ARIMA and LSTM. In Proceedings of the 4th International Conference on Signal Processing and Machine Learning (pp. 181-189).

[6] Sable, R., Goel, S., & Chatterjee, P. (2023). Techniques for Stock Market Prediction: A Review. International Journal on Recent and Innovation Trends in Computing and Communication, 11(5s), 381-402.

[7] Chaudhary, R. (2025). Advanced Stock Market Prediction Using Long Short-Term Memory Networks: A Comprehensive Deep Learning Framework. arXiv preprint arXiv:2505.05325.

[8] Pathak, P. (2024). Stock Market Prediction Using Machine Learning. International Journal for Multidisciplinary Research (IJFMR), 6(1).

[9] Yahoo Finance-Apple Inc. Retrieved from: https://finance.yahoo.com/quote/AAPL/history?p=AAPL.

[10] Yahoo Finance-Alphabet Inc. Retrieved from: https://finance.yahoo.com/quote/GOOG/history?p=GOOG.

[11] Yahoo Finance-Amazon Inc. Retrieved from: https://finance.yahoo.com/quote/AMZN/history?p=AMZN.

[12] Box, G. E. P., & Jenkins, G. M. (1976). Time Series Analysis: Forecasting and Control. Holden-Day.

[13] Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer.

[14] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.

[15] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735-1780.