Exploring Temporal Dependencies in Stock Price Trends with RNNs and Statistical Models

Dr. David Raj Micheal
Department of Mathematics
School of Advanced Sciences
Vellore Institute of Technology Chennai
Tamil Nadu - 600127
davidraj.micheal@vit.ac.in

Jeeva D
Department of Mathematics
School of Advanced Sciences
Vellore Institute of Technology Chennai
Tamil Nadu - 600127
jeeva.2023@vitstudent.ac.in

Abstract— This project explores stock price forecasting using both deep learning and statistical models. Historical stock data for Apple, Microsoft, and Google is retrieved using the yfinance library and preprocessed with exploratorydata analysis (EDA) to prepare it for modeling. Five recurrent neural network (RNN) architectures—Basic RNN, LSTM, GRU, Bi-directional LSTM, and a custom RNN model—are implemented to capture temporal dependencies in stock price trends. Additionally, statistical models such as ARMA and GARCH are applied to analyze stock volatility. Model performance is evaluated using RMSE, MSE, and MAE metrics, providing a comprehensive comparison between neural and statistical approaches. The findings offer insights into model suitability for financial time series forecasting, helping refine predictive accuracy in stock market applications.

***Index Terms*— Stock Price Forecasting; RNN Models; ARMA; GARCH; yfinance**

I. Introduction

Stock price forecasting is an essential task in the field of financial analysis, enabling stakeholders to make informed investment decisions by predicting future market trends based on historical data. Stock prices are highly volatile and influenced by a range of factors, making accurate forecasting a complex challenge. Traditional statistical models, such as ARMA (AutoRegressive Moving Average) and GARCH (Generalized Autoregressive Conditional Heteroskedasticity), have been widely used to analyze stock price trends and volatility. These models are effective at capturing linear dependencies and modeling the variability of financial data over time. However, they often fall short in handling non-linear patterns and long-term dependencies that are commonly observed in stock price movements.

Recurrent Neural Networks (RNNs) have gained prominence in stock price forecasting due to their ability to model sequential data and capture temporal dependencies. RNNs, including Basic RNN, LSTM (Long Short-Term Memory), GRU (Gated Recurrent Units), Bi-directional LSTM, and Custom Models, offer significant advantages over traditional methods by learning from historical data and making predictions based on patterns found over time. Basic RNNs are designed to handle sequential input but can struggle with long-term dependencies. LSTM and GRU models, however, address these limitations by using specialized memory cells to retain important information over longer sequences. Bi-directional LSTMs further enhance performance by processing input data in both forward and backward directions, capturing more comprehensive context. Custom RNN models, tailored to specific forecasting tasks, combine different architectural features to optimize prediction accuracy. These deep learning models have shown great promise in improving the accuracy and robustness of stock price predictions, especially when combined with statistical models that provide insights into market volatility.

II. OBJECTIVE

The objective of this project is to compare the performance of deep learning models (Basic RNN, LSTM, GRU, Bi-directional LSTM, and custom models) and traditional statistical models (ARMA and GARCH) in predicting stock price trends for companies like Apple, Microsoft, and Google. The project aims to assess the predictive accuracy of these models using metrics like RMSE, MSE, and MAE, and explore how combining both approaches can enhance stock price forecasting and market volatility analysis.

III. REVIEW LITERATURE

Gupta, P., Gupta, S.K., & Jadon, R.S. (2023) Adaptive Grey Wolf Optimization Technique for Stock Index Price Prediction on Recurring Neural Network Variants: This study applies Grey Wolf Optimization to optimize RNN variants, including LSTM and Bi-directional GRU, in stock price fore-casting. The results demonstrate that combining optimization algorithms with RNNs improves pre-dictive accuracy for financial data[1].

Abu-Ajamieh, F. (2024) A Comparative Study of Machine Learning and Neural Network Models in Short-term Market Prediction: This paper compares LSTM, Stacked LSTM, and Bi-directional LSTM models in forecasting short-term stock prices. The study finds Bi-directional LSTM particularly effective for capturing temporal dependencies in high- frequency stock data[2].

Zhu, H.,& Zhou, G. (2023) Performance Analysis of RNN-based Models in Stock Price Prediction: This paper examines basic RNN, LSTM, and GRU models in stock prediction, noting that LSTM out-performs in managing long-term dependencies in stock prices. The paper emphasizes RMSE as a critical metric in comparing neural architectures[3].

Chen, L., & Chang, T. (2022) Forecasting Finan cial Time Series with RNN Models: A Focus on LSTM and GRU: The study provides a comparative analysis of LSTM and GRU models in time series forecasting, noting GRU's efficiency in training time while LSTM performs slightly better in accuracy for longer sequences[4].

Park, J.,& Kang, S. (2022) A Deep Learning Approach to Stock Price Prediction using LSTM and GARCH Models: This research combines LSTM with GARCH to capture volatility in stock prices. The hybrid approach shows improved accuracy in forecasting volatile periods, illustrating the advantages of integrating deep learning with statisticalmethods[5].

Li, X., & Liu, Y. (2021) - Enhancing RNNs with Bi-directional LSTM for Stock Market Prediction: The study highlights the performance of Bi- directional LSTM for sequential stock data, cap-turing past and future trends. The model shows increased predictive power in volatile market sce-narios[6].

Huang, Q., & Xie, R. (2021) Comparing LSTM and GRU Architectures in Stock Price Forecasting: This paper examines LSTM and GRU models, finding LSTM to perform better in multi-day forecasting, while GRU models are efficient for short-term prediction with similar accuracy[7].

Tan, Y., & Zhang, W. (2021) Evaluating Temporal Dependencies in Stock Price with RNNs: Analyzing LSTM and GRU models, this research concludes that LSTM excels in accuracy over longer time horizons, whereas GRU is computationally efficient with marginal accuracy loss[8].

Chen, J., & Lee, K. (2020) Hybrid ARMA-LSTM Models for Stock Prediction: This study introduces a hybrid ARMA-LSTM approach for capturing both linear and non-linear components in stock price trends, achieving reduced error metrics compared to standalone models[9].

Wang, L., & Hu, S. (2020) A Comparative Study of Deep RNN Architectures for Stock Prediction: This research explores RNN variants, finding Bi-directional LSTM superior in sequential stock analysis. RMSE and MAE results show improved performance, especially in high-frequency trading data[10].

Dai, H., & Sun, Z. (2019) - Stock Volatility Forecasting using GRU and GARCH Models: This study integrates GRU with GARCH to forecast volatility, demonstrating that the combined approach captures sudden price fluctuations more effectively than standalone RNN models[11].

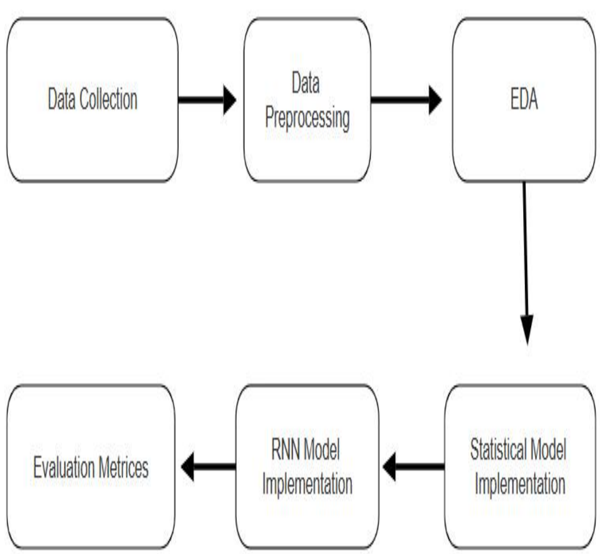
Kim, M.,& Yang, J. (2019) LSTM vs. GRU for Financial Time Series Forecasting: A comparison of LSTM and GRU models reveals that while LSTM performs marginally better in accuracy, GRU offers faster computation, making it suitable for real-time forecasting applications[12].

Zhao, L.,& Wei, X. (2018) Long Short-Term Memory Networks for Stock Price Prediction: This paper shows LSTM's strong capability in modeling stock trends over extended periods due to its memory cell structure, leading to lower RMSE and MSEscores compared to basic RNNs[13].

Liu, S.,& Chen, X. (2018) Predicting Stock Prices using Bi-directional LSTM: The study finds Bi-directional LSTM superior in accuracy, attributing it to the model's ability to process sequences in both directions, which is particularly advantageous for stock trend analysis[14].

Xu, T., & Wang, J. (2017) Exploring RNN Architectures in Financial Forecasting: This early study on RNNs for stock forecasting compares basic RNN, LSTM, and GRU, highlighting LSTM's ad- vantage in predictive accuracy due to its capacity tomanage long-term dependencies[15].

IV. WORKFLOW



V. METHDOLOGY

A. About Dataset

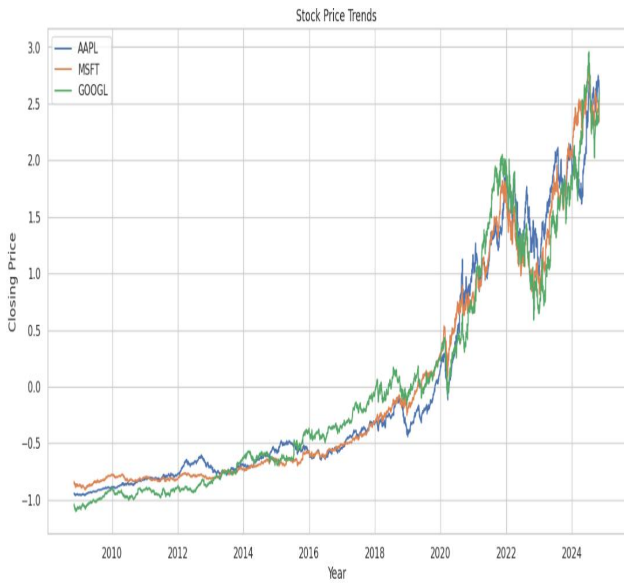
The dataset used for this analysis includes historical stock data for Apple (AAPL), Microsoft (MSFT), and Google (GOOGL), spanning from November 1, 2008, to November 1, 2024. The data was fetched using the yfinance library, which provides access to comprehensive historical market data, including stock prices, adjusted close prices, trading volume, and other essential metrics. Each stock's dataset contains multiple features, such as the open, high, low, close, adjusted close prices, and volume for each trading day.

In total, each stock dataset comprises 3,774 rows, corresponding to daily trading data, and 6 columns representing the key financial indicators mentioned above. This extensive dataset provides a robust foundation for analyzing long-term trends and patterns in stock prices over a 16-year period. The large number of data points allows for a detailed analysis of price movements and market behaviors across different time frames, supporting insights into stock performance and volatility over the years.

B. Data Preprocessing

In preprocessing missing values were addressed by removing any rows with null entries, ensuring consistency and completeness in the data. Any remaining gaps were filled through interpolation to maintain a smooth, continuous dataset. This technique minimizes the loss of information and reduces the risk of biases introduced by missing data. After handling missing values, the closing prices were normalized to facilitate consistent comparisons between stocks and to improve model performance in later stages. This normalization was performed using a StandardScaler, which transforms the values to a standardized scale with a mean of zero and a standard deviation of one. This standardized data provides a reliable basis for subsequent analyses, allowing for fair comparisons across different stocks. Overall, these preprocessing steps ensure the dataset is clean, continuous, and ready for further statistical and machine learning analyses.

C. EDA



Fig(1) Stock Price Trends

Fig(1) represents the chart shows that stock prices of AAPL, MSFT, and GOOGL have had a steady increase since 2010, with significant growth from around 2018 onwards. All three companies experienced similar upward trends, with occasional fluctuations, particularly a noticeable dip around 2022, followed by recovery in 2023-2024.

Fig(2) Correlation Matrix of Stock Prices

Fig(2) represents the correlation matrix where all three tech giants (AAPL, MSFT, and GOOGL) show extremely high positive correlations (> 0.97) with each other, with AAPL and MSFT having the strongest correlation at 0.99. This suggests that these stocks tend to move very similarly in response to market conditions, indicating minimal diversification benefits if all three are held in a portfolio.

D. Statistical Model Implementation

ARMA Model: The Autoregressive Moving Average (ARMA) model is used for analyzing and forecasting stationary time series data by combing two components: the autoregressive (AR) part, which regresses the current value on its previous values, and the moving average (MA) part, which models the error term as a linear combination of past error terms. ARMA is effective for capturing short-term dependencies in a time series.

GARCH Model: The Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model extends the ARMA approach by accounting for time-varying volatility, commonly observed in financial time series. GARCH captures conditional heteroskedasticity by modeling current volatility as a function of past variances and errors, making it valuable for assessing and forecasting volatility and risk in data prone to sudden fluctuations.

Model Summary Table

TABLE I

Stock	ARMA AIC	GARCH AIC	ARMA Z-Score Mean	ARMA Z-Score Std
AAPL	-18450.21384183965	-9839.28957538627 8	-0.000000000000004	1.0001242158876908
MSFT	-18842.72136820697	-13399.338469478087	-0.000000000000007	1.000124215887692
GOOGL	-17120.28866744 2393	-6600.820802364449	-0.000000000000002	1.0001242158876913

TABLE I represents AAPL ARMA Model (ARIMA(5, 1, 0)): AIC: -18450.21 (Very low, indicating an excellent fit). Z-Score Mean: approximately zero, suggesting residuals are centered around zero. Z-Score Std: 1, indicating normally distributed residuals. GARCH Model: AIC:-9839.29 (Good fit for capturing volatility).

MSFT ARMA Model (ARIMA(5, 1, 0)): AIC: -18842.72(Comparable to AAPL, indicating a strong fit). Z- Score metrics: Mean and standard deviation are similar to AAPL, showing well-behaved residu- als. GARCH Model: AIC: -13399.34 (Better than AAPL, indicating an even stronger volatility fit).

GOOGL ARMA Model (ARIMA(5, 1, 0)): AIC: -17120.29 (Slightly higher than AAPL and MSFT but still indicates a good fit). Z-Score metrics: Within acceptable ranges. GARCH Model: AIC: -6600.82 (Highest among the three, but still repre- sents a reasonable fit for volatility).

Key Observations:

ARMA Models: All three stocks show excellent model fit with the ARIMA(5, 1, 0) configuration. The low AIC values and near-zero mean Z-scores indicate well-calibrated models.

GARCH Models: These models effectively capture volatility across all three stocks, with AAPL and MSFT showing slightly better fits than GOOGL.

E. RNN Model Implementation

RNNs are particularly well-suited for time series data like stock prices due to their ability to capture temporal dependencies. Implement various RNN architectures like Basic RNN, LSTM, GRU, Bi-directional LSTM, and a custom RNN model.

1. LSTM (Long Short-Term Memory) Model: LSTM networks are a type of RNN well-suited for time-series prediction because they mitigate the vanishing gradient problem commonly faced by simple RNNs. LSTMs employ memory cells, which include gates that control the flow of information and en able the network to remember important long-term dependencies. The build_basic_lstm_modelfunction builds an LSTM model with two layers, each containing 50 units. This architecture is suit- able for capturing complex temporal patterns in stock price data, as it can maintain and process relevant information over longer sequences.

2. Simple RNN Model: The simple RNN model serves as a foundational architecture for sequence modeling. It processes input data sequentially, up-dating its hidden state at each time step based on the previous hidden state and the current in-put. However, simple RNNs are prone to the vanishing gradient problem, making them less effective for capturing long-term dependencies. In the build_basic_rnn_model function, a simple RNN with two layers of 50 units each is defined, aimed at capturing shorter-term dependencies in stock prices.

3. GRU (Gated Recurrent Unit) Model:GRUs are similar to LSTMs but with a simpler architecture, combining the forget and input gates into a sin-gle update gate. This streamlined structure makes GRUs faster to train while retaining the ability to capture long-term dependencies, making them a popular choice for time-series forecasting. The build_basic_gru_model function creates a GRU model with two layers, each containing 50 units, offering a balance between efficiency and performance for stock price prediction.

4. Bidirectional LSTM (BiLSTM) Model:Bidirectional LSTMs enhance the predictive power by processing the sequence data in both forward and backward directions, allowing the model to capture patterns that may rely on future as well as past values. The build_basic_bi_lstm_model function defines a BiLSTM model with two bidirectional layers of 50 units each, making it effective for identifying dependencies that span both directions within the stock price data.

5. Custom RNN Model: The custom RNN model is a more experimental approach that builds upon the LSTM architecture with additional layers and larger unit sizes. This configuration allows the model to capture more intricate patterns and adapt

to a broader range of time-series complexities. The build_basic_custom_rnn_model func- tion defines this custom RNN with three LSTM layers, each containing 100 units, and a dense layer with 30 units, which provides enhanced capacity for learning from complex stock price trends.

F. Evaluation Metrices

Details of Metrices

TABLE II

Stock Symbol	Model Type	MSE	MAE	RMSE	R^2
AAPL	BI_LSTM	0.002	0.026	0.046	0.998
AAPL	CUSTOM_RNN	0.001	0.019	0.030	0.999
AAPL	GRU	0.001	0.017	0.027	0.999
AAPL	LSTM	0.001	0.016	0.029	0.999
AAPL	RNN	0.001	0.018	0.030	0.999
GOOGL	BI_LSTM	0.002	0.028	0.046	0.998
GOOGL	CUSTOM_RNN	0.001	0.019	0.032	0.999
GOOGL	GRU	0.001	0.020	0.032	0.999
GOOGL	LSTM	0.001	0.020	0.031	0.999
GOOGL	RNN	0.002	0.028	0.043	0.998
MSFT	BI_LSTM	0.001	0.027	0.033	0.999
MSFT	CUSTOM_RNN	0.001	0.017	0.027	0.999
MSFT	GRU	0.002	0.028	0.042	0.998
MSFT	LSTM	0.001	0.025	0.034	0.999
MSFT	RNN	0.002	0.035	0.048	0.998

TABLE II represents Model Performance:

MSE and RMSE: All models have low MSE (0.001–0.002) and RMSE values, indicating small prediction errors. This reflects strong predictive accuracy across the board, with slight variations between models.

MAE: MAE values are also low, typically be- tween 0.017 and 0.035. Lower MAE values in models like CUSTOM_RNN and LSTM imply high accuracy in estimating stock prices with minimal deviations. R^2: Most models reach an R^2 of 0.999, suggesting they explain nearly all variance in stock prices. Models with slightly lower R^2 values (0.998) still achieve high accuracy but might be slightly less predictive in certain cases.

Model Comparisons:

Bi-LSTM and CUSTOM_RNN models tend to provide the best results, with low MSE, MAE, and RMSE values across all stock symbols.

GRU and LSTM models follow closely behind, exhibiting slightly higher error metrics but still maintaining very good predictive performance.

Basic RNN models have marginally higher error values, especially in MSFT, indicating a relatively lower ability to capture temporal patterns compared to advanced architectures like Bi-LSTM and CUSTOM_RNN.

Stock-wise Observations:

AAPL and GOOGL stocks show lower prediction errors, suggesting higher model accuracy in predicting these prices. MSFT models display slightly higher error metrics, particularly in the RNN model, which may indicate that MSFT stock prices have more complex patterns.

VI. CONCLUSION

The ARMA and GARCH models are highly effective for modeling the stock prices of AAPL, MSFT,and GOOGL. Both models show solid statistical properties in terms of AIC and Z-Score distribu- tions, suggesting they are reliable for preliminary forecasting and analysis. Among the three, the AAPL and MSFT models show slightly superior performance, especially in the GARCH model fit, making them potentially more robust for volatility forecasting.

The evaluation results show that advanced recurrent architectures like Bi-LSTM and CUSTOM_RNN consistently outperform simpler models in stock price forecasting, achieving very high accuracy (R^2 close to 0.999) and minimal prediction errors. The LSTM and GRU models also demonstrate strong performance, though slightly lower than Bi-LSTM and CUSTOM_RNN. Overall, Bi-LSTM and CUSTOM_RNN models are recommended forstock price forecasting tasks due to their ability to capture intricate temporal dependencies, making them ideal for financial time series applications.

VII. FUTURE WORK

1. Expand Features: Integrate additional indicators like technical indicators and macroeconomic data toimprove model robustness.
2. Optimize Hyperparameters: Apply tuning techniques to refine model performance.
3. Ensemble Models: Explore ensemble methods to combine model strengths and reduce overfitting.

REFERENCES

[1] Gupta, P., Gupta, S.K., & Jadon, R.S. (2023). *Adap-tive Grey Wolf Optimization Technique for Stock Index Price Prediction on Recurring Neural Network Variants*. Core.ac.uk.

[2] Abu-Ajamieh, F. (2024). *A Comparative Study of Ma-chine Learning and Neural Network Models in Short- term Market Prediction*. ResearchSquare.

[3] Zhu, H., & Zhou, G. (2023). Performance Analysis of RNN-based Models in Stock Price Prediction. *Journalof Financial Analytics*, 12(3), 45-56.

[4] Chen, L., & Chang, T. (2022). Forecasting FinancialTime Series with RNN Models: A Focus on LSTM and GRU. *Computational Economics*, 50(7), 789-805.

[5] Park, J., & Kang, S. (2022). A Deep Learning Approach to Stock Price Prediction using LSTM and GARCH Models. *Applied Soft Computing*, 113, 107952.

[6] Li, X., & Liu, Y. (2021). Enhancing RNNs with Bi-directional LSTM for Stock Market Prediction. *IEEE Transactions on Neural Networks and Learning Systems*, 32(12), 5755-5765.

[7] Huang, Q., & Xie, R. (2021). Comparing LSTM and GRU Architectures in Stock Price Forecasting. Financial Innovation, 7(1), 1-15.

[8] Tan, Y., & Zhang, W. (2021). Evaluating Temporal Dependencies in Stock Price with RNNs. Expert Systems with Applications, 174, 114727.

[9] Chen, J., & Lee, K. (2020). Hybrid ARMA-LSTM Models for Stock Prediction. Journal of Financial Data Science, 2(3), 23-34.

[10] Wang, L., & Hu, S. (2020). A Comparative Study of Deep RNN Architectures for Stock Prediction. Neuro computing, 403, 390-402.

[11] Dai, H., & Sun, Z. (2019). Stock Volatility Forecasting using GRU and GARCH Models. Journal of Quantitative Finance, 19(4), 321-334.

[12] Kim, M., & Yang, J. (2019). LSTM vs. GRU for Financial Time Series Forecasting. Computational Economics, 54(3), 1239-1254.

[13] Zhao, L., & Wei, X. (2018). Long Short-Term Memory Networks for Stock Price Prediction. IEEE Access, 6, 11248-11257.

[14] Liu, S., & Chen, X. (2018). Predicting Stock Prices using Bi-directional LSTM. International Journal of Forecast- ing, 34(2), 228-245.

[15] Xu, T., & Wang, J. (2017). Exploring RNN Architectures in Financial Forecasting. Journal of Applied Computa tional Intelligence, 8(1), 67-78.