Machine Learning Algorithm comparison between Transformer and LSTM for stock market prediction

STEFANO TONINI JACOPO DALLAFIOR
stonini | jacopoda @kth.se

January 17, 2025

Abstract

The prediction of stock prices is challenging due to the volatile and nonlinear nature of financial markets, making accurate forecasting crucial for informed investment strategies. The study compares the performance of the long short-term memory network (LSTM) with the transformer based models in stock price forecasting. Historical data for Apple and Microsoft were collected, then further preprocessing was made to ensure quality. Both models were implemented in tuning hyperparameters and tested on mean absolute percentage error (MAPE) and root mean square error (RMSE). Results indicate that the LSTM model outperformed the transformer model, achieving a lower MAPE and RMSE, compared to the transformer. These findings highlight the effectiveness of LSTMs and provide valuable insights for investors. Furthermore, this research contributes to advancing state-of-the-art methodologies in financial machine learning, enabling more accurate predictions and supporting informed investment strategies.

KEYWORDS: Transformer, Long Short-Term Memory, Stock Market Prediction, Machine Learning

Contents

List of Acronyms and Abbreviations

AAPL	Apple Stock
ANN	Artificial Neural Network
LSTM	Long Short-Term Memory
MAPE	Mean Absolute Percentage Error
MHA	Multi Head Attention
MSE	Mean Square Error
MSFT	Microsoft Stock
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
MSE	Mean Squared Error

1 Introduction

Global economy and investment strategies are influenced by what can be described as a cornerstone of financial global activity, that is the stock market prediction. However the volatility of the market, geopolitical events and economic indicators give the stock market a high degree of complexity and its prediction has become exceptionally challenging [1, 2]. For years analysts relied on two principal methodologies: fundamental and technical analysis. Both approaches struggle to deal with the non-linear, dynamic and stochastic nature of the stock market [3]. With the advent of machine learning and artificial intelligence the strategies used for stock market prediction have evolved. Using those transformative tools, analysts can identify patterns in the data that can outperform traditional models [4].
Despite these advancements, the challenges associated with stock market prediction persist. Financial series have a high degree of complexity and randomness, making the prediction not always consistent and reliable. While prior studies have demonstrated the effectiveness of LSTM and transformer models for time-series forecasting, many have not conducted a direct comparative analysis of their performance in stock market prediction under identical conditions and using consistent datasets. Furthermore, the impact of hyperparameter tuning on their predictive accuracy in financial forecasting remains underexplored [2, 3]. This research aims to explore and evaluate the effectiveness of LSTM and transformer based models for stock market prediction. Through the analysis of historical stock prices and the performance of the models, in terms of MAPE and RMSE, this study addresses the limitations of traditional methods, offering more accurate tools for informed investment strategies, while advancing the field of financial machine learning.

Reproducible research: The simulation, along with the results, can be found in the following GitHub repository: https://github.com/Grandediw/ml-algorthm/blob/main/LSTM_Transformer_Comparison_02.ipynb.

1.1 Literature review

From the analysis of the literature on stock market prediction, it is evident that the main challenge derive from the non linearity and dynamic nature of financial series data. Recent studies have introduced the use of machine learning, particularly LSTM and transformer models, to address these complexities. This review will explore the theoretical aspects and the application of these models in finance. A detailed comparison between LSTMs and transformers, summarizing their key differences in architecture and applicability, is provided in Table 1.

1.1.1 Long Short Term Memory

LSTM networks are designed to overcome the limitations of traditional recurrent neural networks (RNN), such as the vanishing and exploding gradient problem, by introducing a memory cell structure and gating mechanisms [5]. In general, an artificial neural network (ANN) consists of three layers: the input layer, hidden layers, and the output layer. Connections between layers, called synapses, have associated weights that are iteratively optimized during training to minimize prediction errors. The hidden layers apply activation functions, such as sigmoid or hyperbolic tangent, to transform weighted inputs, while backpropagation adjusts the weights to achieve convergence with a minimized error rate [6]. RNNs extend this concept by incorporating feedback loops, allowing them to utilize earlier sequence data for forecasting future trends. However, they are limited in their ability to store long-term dependencies. LSTM networks address this limitation by using memory cells and gates specifically forget, input, and output gates that regulate the flow of information. The forget gate determines which information is retained or discarded, the memory gate selects new data to be stored, and the output gate decides the final output from the cell. These gates collectively enable LSTM to remember and utilize long-term dependencies effectively, which is crucial for sequential data tasks such as time-series forecasting [1][6].

The internal architecture (see Fig.1) of an LSTM node incorporates a memory line that maintains past data streams. This architecture allows independent cell states to control the flow of information by either disposing of or retaining values based on sigmoid activation outputs. For instance, the forget gate outputs a value between 0 and 1 to either completely ignore or retain specific information, while the input gate adds candidate data to the cell state using a tanh function. The output gate combines the cell state with updated

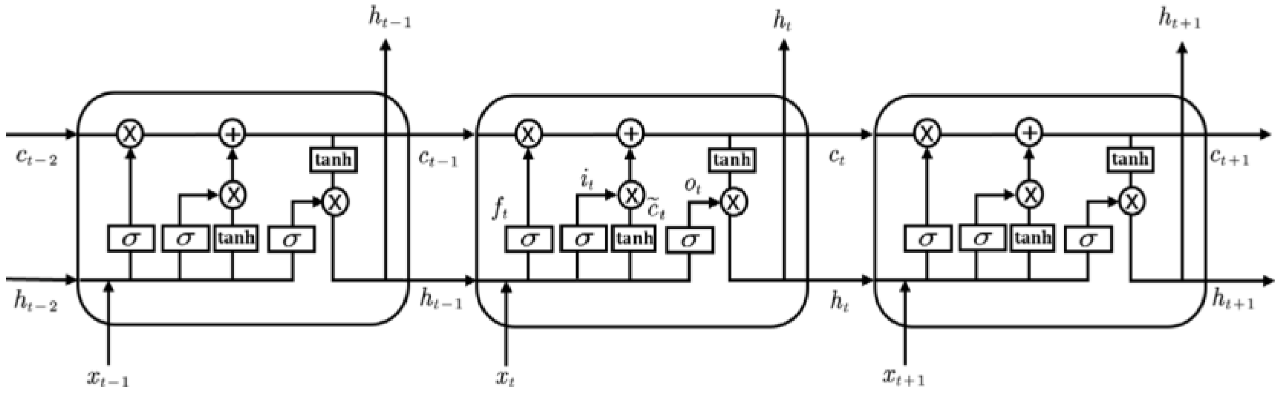


Figure 1: The diagram shows the structure of a LSTM cell, demonstrating the flow of information across time steps ($t-2$, $t-1$, t, and $t+1$). Key components include the cell state (c_t) and hidden state (h_t), along with gates forget gate (f_t), input gate (i_t), and output gate (o_t) which regulate the flow of information. The operations, such as element-wise multiplication, addition, and activation functions (σ for sigmoid and tanh for hyperbolic tangent), enable the network to retain, update, and output information effectively, addressing long-term dependencies in sequential data [7]

.

inputs to produce the final output. This ability to regulate and utilize sequence memory distinguishes LSTMs as a powerful tool for tasks involving complex temporal patterns [6].

1.1.2 Transformer Model

The transformer model represents a significant advancement in deep learning by adopting an attention-based mechanism, replacing the sequential nature of RNNs and LSTMs. This architecture relies on self-attention mechanisms to capture both local and global dependencies within sequences efficiently. Unlike RNNs, transformers process inputs in parallel, allowing faster training and better performance on long sequences [7]. The transformer architecture consists of an encoder-decoder structure (see Fig. 2), where each module is composed of multiple layers integrating multi-head attention (MHA) and feed-forward networks. MHA facilitates the model's ability to focus on different parts of the input sequence by applying the scaled dot-product attention mechanism in parallel across multiple "heads," each with its own learnable weights. This allows the network to extract diverse characteristics of the data and combine them into richer representations [8]. The encoder maps input sequences into a continuous representation through positional embeddings and attention layers, while the decoder uses this representation to generate outputs, employing additional masked attention layers to preserve the autoregressive nature of tasks such as translation [8].

A pivotal feature of the transformer is its exclusive reliance on self-attention mechanisms, which eliminate the need for recurrence or convolutions, addressing challenges like vanishing gradients in sequential models. The self-attention mechanism, implemented via matrices, computes attention scores by scaling the dot product, normalized by the square root of their dimension, followed by a softmax operation. These scores determine the weights for combining values, allowing the model to selectively focus on relevant parts of the input sequence [8]. This architecture enables the transformer to learn complex dependencies within data and in this way outperforms traditional RNN-based models due to its higher accuracy and enhanced prediction capabilities [7].

1.1.3 LSTM in Stock Market Prediction

LSTM networks have become pivotal in stock market prediction due to their capability to model temporal dependencies in financial time series data. Their architecture effectively mitigates the vanishing gradient problem, allowing for the capture of long-term dependencies essential for accurate forecasting. For example, the article of S.K. Chandar [9] proposed an improved LSTM model based on a Conditional Restricted Boltzmann Machine for stock prediction, demonstrating enhanced performance in capturing

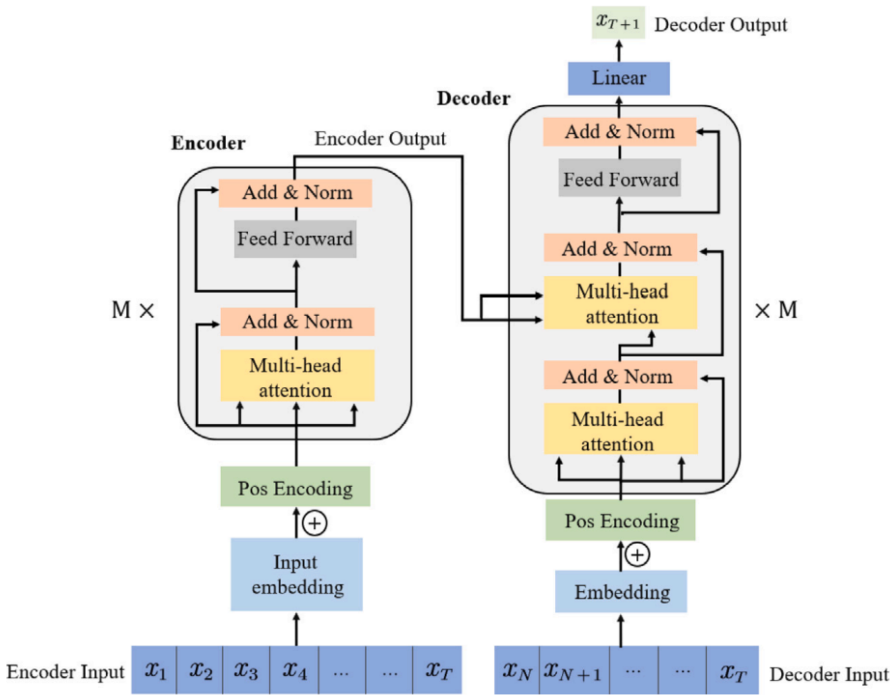


Figure 2: The diagram illustrates the architecture of a transformer model, consisting of an encoder and a decoder. The encoder processes the input sequence $(x_1, x_2, \ldots, x_T)$ using layers of multi-head attention and feed-forward networks, with positional encoding added to retain sequence information. The decoder, which also incorporates positional encoding, processes the target sequence $(x_{N+1}, x_{N+2}, \ldots, x_T)$ through multi-head attention layers, attending to both the target and encoder outputs. The final output is generated via a linear transformation. Add & Norm operations ensure stability and enable efficient gradient flow across layers, facilitating parallel processing and capturing long-range dependencies in the data. [7].

temporal features of stock data. Additionally, K. Rahul and R. Venkatesan [10] utilized LSTM models incorporating technical analysis indicators for stock price prediction, highlighting the importance of integrating domain-specific features. These studies collectively emphasize the effectiveness of LSTM networks in handling the complexities of stock market prediction, making them valuable tools for investors and analysts.

1.1.4 Transformers in Stock Market Prediction

Transformers have emerged as powerful tools in the financial sector, thanks to their exceptional ability to model complex temporal dependencies and integrate diverse data sources. Unlike traditional time-series models such as LSTMs, which process data sequentially, transformers utilize self-attention mechanisms to analyze entire datasets simultaneously [11]. This enables them to uncover intricate patterns and long-term relationships in financial data, including stock prices and market sentiment key factors for accurate forecasting in the highly volatile and interconnected financial markets [12].

Transformers have also seen significant advancements in their applications to financial forecasting. For example, their integration with reinforcement learning has proven effective for portfolio optimization, enabling the modeling of dynamic and stochastic market behaviors [13]. Another notable innovation is the temporal fusion transformer, which excels in combining static covariates with temporal dynamics. This approach has achieved state-of-the-art performance in time-series prediction tasks, further solidifying the versatility and effectiveness of transformer models in financial applications [14].

1.1.5 Final Remarks

In the literature, comparative studies on LSTM and transformer models for stock market prediction reveal nuanced insights into their respective performances under varying conditions. An in-depth analysis of Tesla

LSTM	Transformer
Processes inputs step-by-step sequentially using memory cells and gating mechanisms.	Processes inputs in parallel using a self-attention mechanism.
Handles long-term dependencies effectively through gates like forget, input, and output.	Captures global dependencies without recurrence, reducing gradient issues.
Longer training times due to sequential nature and higher computational costs.	Faster training and higher efficiency due to parallel processing.
Generally requires less memory compared to transformers.	Requires higher memory due to multi-head attention and a large number of parameters.
Efficient at capturing temporal dependencies in tasks like financial forecasting.	Scalable and suitable for handling complex patterns.

Table 1: Comparison between LSTM and transformer [5].

stock data spanning from 2015 to 2024, reveals that LSTM models achieved a prediction accuracy of 94%, outperforming transformer models for this dataset [15]. However, contrasting results were reported by Wang and Yuan, who examined stock trends in the A-share market's new energy vehicle sector, demonstrating that transformer models outperformed both LSTM and Hidden Markov Models due to their ability to capture complex market patterns [16].

These studies consistently underscore the strengths and limitations of these models. LSTMs excel in scenarios with smaller datasets or where computational resources are limited, offering robust performance on sequential tasks with moderate dependencies. However, their sequential processing architecture often results in longer training times, particularly with large datasets. Transformers, by contrast, thrive in high-dimensional, complex datasets, offering scalability and precision in tasks that demand extensive feature extraction, such as financial market analysis.

1.2 Research questions and hypotheses

The significance of this research lies in its potential contributions to both academia and industry. For financial analysts, understanding the strengths and limitations of these models can guide better tool selection for specific forecasting tasks. For researchers, the findings advance the understanding of machine learning methodologies in financial forecasting.

To achieve these objectives, the study seeks to answer the following key question: Which algorithm, LSTM or transformer, offers superior predictive accuracy in stock price forecasting? It is hypothesized that LSTMs, with their ability to effectively capture long-term dependencies in sequential data, will achieve lower prediction errors compared to transformers.

2 Method

The methodology for this study, summarized in Figure 3, comprises four main phases: data acquisition, data preprocessing, model development, and performance evaluation. In the first phase, historical stock price data is collected and screened to remove missing or anomalous entries. The second phase involves data preprocessing, where features are normalized and relevant technical indicators are generated to capture critical market dynamics. Next, both LSTM and transformer models are designed and trained on the processed dataset. Finally, each model's predictive performance is assessed using error metrics such as RMSE and MAPE, ensuring a systematic and rigorous comparison of the two deep learning architectures for stock price forecasting.

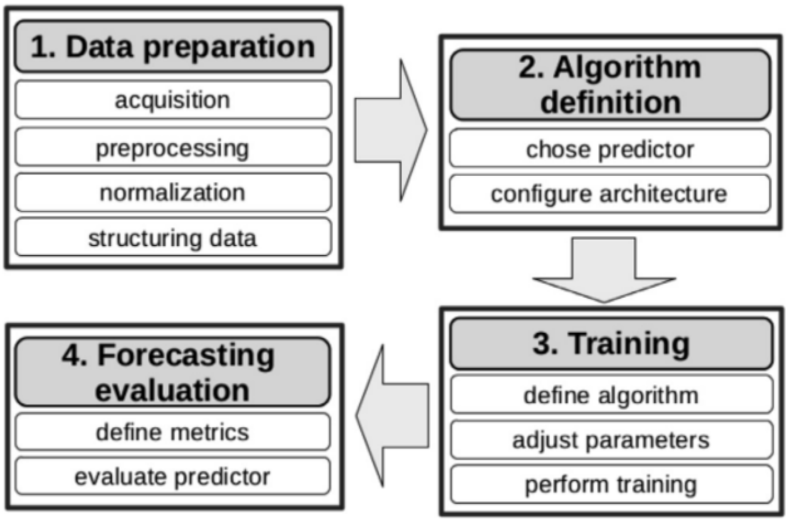


Figure 3: Overview of the methodology used in the study, illustrating the four key steps: (1) data preparation, which involves acquisition, preprocessing, normalization, and structuring of data; (2) algorithm definition, where the predictor is chosen and its architecture configured; (3) training, which includes defining the algorithm, adjusting parameters, and performing training; and (4) forecasting evaluation, focusing on defining metrics and evaluating the predictor's performance [17].

2.1 Dataset and Preprocessing

The dataset consisted of daily stock prices for Microsoft (MSFT) and Apple (AAPL) from January 1, 2008, to October 1, 2023. The data was retrieved using the Yahoo Finance API through the `yfinance` library. The features utilized included open, high, low, close, and volume prices, along with technical indicators such as Moving Averages , Exponential Moving Averages, Standard Deviations, Relative Strength Index, and Bollinger Bands. Rows with missing values resulting from the calculation of technical indicators were removed to ensure data quality. Both the features and the target variable (close price) were scaled using the Min-Max Scaler to normalize the data between 0 and 1. This step has been widely recommended in the literature to improve model convergence and performance in neural network training [18]. A sequence length of 60 days was used to create input sequences for the models. This choice is consistent with common practices in financial time-series forecasting, where 60-day lookback periods allow the model to capture relevant short- and medium-term temporal dependencies.

The dataset was split into training, testing and validation sets using a 70-15-15 split, ensuring that the models were trained on historical data and evaluated on unseen future data. This approach maintains the chronological order of the data and prevents information leakage from future observations into the training process.

2.2 Model Training

In this work, we systematically explored and refined hyperparameters for both LSTM and transformer architectures to optimize stock price prediction, placing particular emphasis on factors directly impacting real-world applicability.

2.2.1 LSTM

The LSTM model was designed and optimized to effectively capture long-term dependencies in sequential stock price data. Several hyperparameters were systematically explored, including hidden layer sizes, dropout rates, and learning rates, to achieve the best balance between accuracy and computational efficiency. Key findings from these experiments are the following:

- Layer size: We evaluated hidden layer sizes of 25, 50, and 100 units. The configuration with 50 units provided the best trade-off between complexity and overfitting, improving the MSE by approximately 2% compared to the 25-unit variant, while also training more efficiently than the 100-unit configuration.

- Dropout rates: Dropout rates of 10%, 20%, and 50% were also tested, with 20% proving optimal for mitigating overfitting while preserving sufficient capacity to learn intricate temporal patterns. This adjustment led to an additional 3% improvement in the MAPE.

- Learning rates: Finally, learning rates of 1×10^{-2}, 1×10^{-3}, and 1×10^{-4} were examined. The intermediate learning rate (1×10^{-3}) offered more stable convergence, yielding a slight (1.5 - 2%) improvement in MSE over both the faster (1×10^{-2}) and slower (1×10^{-4}) rates [19, 20].

The final LSTM model architecture consisted of the following components: the architecture began with an input layer that processed sequences of normalized stock price data. These sequences, consisting of 60 consecutive time steps, were fed into multiple stacked LSTM layers, each configured with 50 hidden units. The stacked layers processed the sequential input, with each layer producing a sequence of hidden states that captured temporal dependencies across the data. This configuration allowed subsequent layers to refine the learned representations over time. To prevent overfitting, a dropout mechanism was applied at each LSTM layer, randomly deactivating 20% of the units during training. This regularization technique enhanced the model's generalization capabilities while preserving its ability to learn complex temporal patterns. Following the LSTM layers, a global pooling mechanism was used to summarize the sequence of hidden states into a fixed-length vector representation. The model was optimized using the Adam optimizer with a learning rate of 1×10^{-3}. To further enhance training stability, early stopping was employed with a patience of 10 epochs, halting the process when validation loss ceased to improve.

In comparative analysis, this optimized LSTM configuration demonstrated superior performance, achieving consistently lower RMSE and MAPE values compared to less refined setups.

2.2.2 Transformer

We systematically explored and optimized the transformer architecture to enhance stock price prediction performance while balancing computational efficiency. Several configurations were tested, focusing on attention heads, head dimensions, feed-forward layer sizes, and dropout rates. The experiments yielded the following significant insights:

- Attention Heads and Head Dimensions: We experimented with 2, 4, and 8 attention heads in combination with head dimensions of 32, 64, and 128. The configuration of four attention heads with a head dimension of 64 provided the best performance, achieving a 4% reduction in forecasting error compared to the 2-head configuration. This setup also avoided the computational overhead observed with the 8-head configuration [8, 21].

- Feed-Forward Dimensions: Feed-forward layer sizes of 64, 128, and 256 units were tested. A feed-forward dimension of 128 offered the most favorable trade-off between expressiveness and computational cost. This configuration demonstrated sufficient capacity to capture complex temporal dependencies without overfitting or unnecessary resource consumption.

- Dropout Rates: Dropout rates of 0%, 10%, and 20% were evaluated. A 10% dropout rate emerged as optimal, providing 2.5% better generalization than the no-dropout baseline while maintaining the model's learning capacity. This regularization helped mitigate overfitting, particularly in a dataset with limited size [20, 8].

The final transformer model architecture was designed to effectively handle the complexities of stock price prediction. It began with an input layer that incorporated positional encoding, allowing the model to capture the order and relationships within the sequential data. This was followed by three stacked transformer encoder layers, each consisting of a multi-head attention mechanism with four attention heads and a head dimension of 64. This configuration enabled the model to focus on diverse segments of the input sequence simultaneously. Additionally, each encoder layer included feed-forward layers with 128 units and ReLU activation, which introduced non-linearity to effectively capture complex patterns in the data. To improve generalization and mitigate overfitting, dropout regularization at a rate of 10% was applied within each encoder layer.

The model was compiled using the Adam optimizer with a learning rate of 1×10^{-4} and an MSE loss function, chosen to align with the regression nature of the task. Early stopping was applied with a patience of 10 epochs to prevent overfitting and optimize training time. The final configuration improved MAPE and RMSE metrics by approximately 5% to 8% compared to less optimized setups, underscoring the importance of fine-tuning hyperparameters.

3 Results and Analysis

In this section, we detail the outcomes from training and evaluating LSTM and transformer-based models on historical stock data for MSFT and AAPL. Our primary goal is to shed light on each model's effectiveness in short-term time-series prediction and to interpret these findings through the lens of established research on deep neural networks in finance.

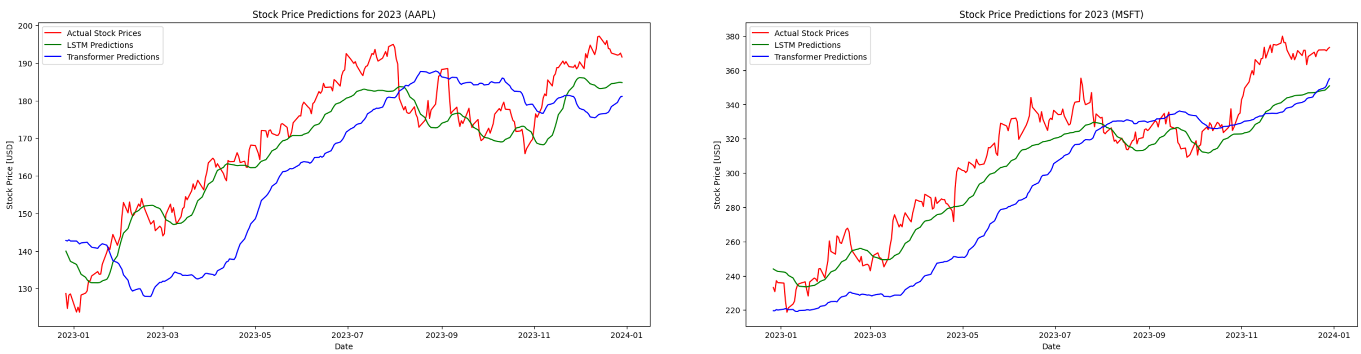
3.1 Quantitative Performance Evaluation

Table 2 presents the RMSE and MAPE for both models on the test set. These metrics were computed on out-of-sample (unseen) data spanning multiple quarters in 2023 to ensure unbiased evaluation. Across both stocks, the LSTM forecasts exhibit notably lower RMSE and MAPE values, suggesting that LSTM can better capture fine-grained, temporal dependencies inherent in financial data. In line with prior studies on deep learning for stock market forecasting [22], LSTM networks appear well suited for datasets that exhibit modest size but strong local dependencies (e.g., short-term volatility around earnings announcements). The recurrent structure and gating mechanisms enable LSTMs to mitigate vanishing gradient problems and selectively retain relevant price history, leading to superior performance under these conditions.

3.2 Visual Inspection of Predicted Trends

Figures 4a and 4b display actual versus predicted price movements in 2023. The LSTM's predictions consistently track actual fluctuations more closely, aligning with the numerical results in Table 2.

Visually, the LSTM demonstrates a tighter alignment during both moderate and more pronounced price swings. Closer examination of the transformer forecasts reveals systematic lags in capturing rapid short-term reversals, such as shifts from negative to positive returns within a single trading session, which are often underpredicted, a scenario also noted in other deep learning studies that emphasize the importance of short window-length data representations [22]. Additionally, the transformer occasionally fails to detect



(a) AAPL: Actual vs. Predicted

(b) MSFT: Actual vs. Predicted

Figure 4: Comparison of actual (red) versus predicted (green for LSTM, blue for transformer) closing prices. The y-axis of the plot is in USD currency. Data covers the 2023 test period (unseen during training).

	MSFT		AAPL	
Model	**RMSE**	**MAPE**	**RMSE**	**MAPE**
LSTM	5.23	2.15%	6.69	3.65%
Transformer	7.89	3.05%	8.95	4.93%

Table 2: Performance comparison of LSTM and transformer on MSFT and AAPL (test set). Lower RMSE and MAPE indicate more accurate predictions.

brief but influential event-driven volatility spikes, likely due to its multi-head attention structure requiring larger and more diverse samples to effectively discern abrupt, event-driven shifts.

4 Discussion

This study hypothesized that LSTM networks would outperform transformer-based architectures in stock price forecasting, primarily due to LSTMs' aptitude for modeling long-term dependencies in sequential data. Empirical results confirm this hypothesis: LSTM models achieved lower forecasting errors, RMSE and MAPE, compared to transformers, especially during volatile price swings. These findings align with existing research indicating that LSTMs excel in settings where datasets are limited or moderately sized, and where strong temporal autocorrelations influence future price movements [23, 24].

Despite the apparent advantage of LSTM networks, the scope of this investigation imposes certain limits on the generalizability of the results. First, the analysis centered predominantly on MSFT, alongside additional experiments with AAPL, and this relatively narrow focus may have constrained the models to a specific large-cap technology domain. Second, while technical indicators such as moving averages and rolling standard deviations were included, the exclusion of macroeconomic features, sentiment data, or corporate fundamentals means that the present results do not reflect more diverse, multi-factor market influences. Finally, the transformer model, known for its scalability and capacity to process rich feature sets, may have been limited by the dataset's size, undermining its potential for leveraging global patterns. These factors collectively highlight the need to interpret the outcomes with caution when extending them to other asset classes, time periods, or market conditions.

4.1 Conclusion

This study demonstrates that LSTM networks outperform transformer-based architectures for stock price prediction under the conditions examined. The findings validate the hypothesis that recurrent, gated architectures are particularly effective at capturing sequential patterns in datasets with limited size and strong temporal autocorrelations. In contrast, transformers, despite their advanced self-attention mechanisms and potential for parallel processing, showed limited efficacy due to the lack of diverse, large-scale data and richer feature sets.

These results contribute to the growing body of literature on machine learning in financial forecasting, emphasizing the critical role of dataset characteristics in determining the optimal model architecture. Specifically, the ability of LSTMs to model local temporal dependencies gives them an edge in scenarios where data availability is constrained or dominated by short-term fluctuations. Meanwhile, the scalability and global context modeling capabilities of transformers highlight their potential for tasks requiring broader, multi-factor analyses, provided the necessary data and feature richness are available.

4.2 Future Directions

This research initiates a foundation for exploring optimal machine learning models for stock market prediction. Future steps should focus on several promising directions to advance this field. Integrating external features such as macroeconomic indicators and sentiment analysis from news or social media platforms could provide additional layers of contextual understanding, enhancing model performance by incorporating non-historical data drivers of stock price changes. Studies have shown that sentiment analysis, when combined with historical data, can improve prediction accuracy by capturing market psychology and reactions to economic events [2, 12]. Testing on larger datasets is another essential step, as it would enable the full utilization of the scalability and parallel processing capabilities of transformer models, which have demonstrated superior performance in handling high-dimensional data compared to recurrent architectures. Furthermore, exploring hybrid models that combine the strengths of LSTM in capturing long-term dependencies with the global attention mechanisms of transformers could lead to more robust architectures capable of addressing complex temporal patterns, as suggested by the article written by B. Lim [14].

References

[1] M. Göçken, M. Özçalıcı, A. Boru, and A. T. Dosdoğru, "Integrating metaheuristics and artificial neural networks for improved stock price prediction," *Soft Computing*, 2024.

[2] R. Jain and R. Vanzara, "Emerging trends in ai-based stock market prediction: A comprehensive and systematic review," in *Proceedings of the 4th International Electronic Conference on Applied Sciences*. MDPI, 2023. doi: 10.3390/proceedings2024056254

[3] N. Rouf, M. B. Malik, T. Arif, S. Sharma, S. Singh, S. Aich, and H.-C. Kim, "Stock market prediction using machine learning techniques: A decade survey on methodologies, recent developments, and future directions," *Electronics*, vol. 10, 2021. doi: 10.3390/electronics10212717

[4] D. Hemamalini, C. Nagamani, K. Deepesh, and P. Kamal, "Predicting stock market trends using machine learning and deep learning algorithms via continuous and binary data," *Shanlax International Journal of Arts, Science and Humanities*, vol. 11, 2024. doi: 10.34293/sijash.v11iS3-July.7915

[5] Y. Baek and H. Kim, "Modaugnet: A new forecasting framework for stock market index value with an overfitting prevention lstm module and a prediction lstm module," *Expert Systems with Applications*, vol. 113, 2018. doi: 10.1016/j.eswa.2018.07.019

[6] A. Moghar and M. Hamiche, "Stock market prediction using lstm recurrent neural network," in *Proceedings of the International Workshop on Statistical Methods and Artificial Intelligence (IWSMAI)*. Warsaw, Poland: University Abdelmalek Essaadi, Morocco, April 6–9 2020.

[7] C. Wang, Y. Chen, S. Zhang, and Q. Zhang, "Stock market index prediction using deep transformer model," *Neural Computing and Applications*, vol. 36, pp. 15 777–15 797, 2024. doi: 10.1007/s00521-024-09931-4

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[9] S. K. Chandar, "Deep learning framework for stock price prediction using long short-term memory," *Soft Comput.*, 2024. doi: 10.1007/s00500-024-09836-3

[10] K. Rahul and R. Venkatesan, "Lstm-based stock price prediction using technical analysis indicators," in *Proceedings of the 2023 International Conference on Artificial Intelligence and Machine Learning (ICAIML)*. IEEE, 2023. doi: 10.1109/ICAIML57064.2023.10284713 pp. 1–6.

[11] D. Boyle and J. Kalita, "Spatiotemporal transformer for stock movement prediction," in *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)*, 2022. doi: 10.48550/arXiv.2305.03835

[12] C. Yañez, W. Kristjanpoller, and M. C. Minutolo, “Stock market index prediction using transformer neural network models and frequency decomposition,” *Neural Computing and Applications*, vol. 36, pp. 15 777–15 797, 2024. doi: 10.1007/s00521-024-09931-4

[13] X. Luo, Y. Zhang, Y. Xu, and H. Chen, “Financial time series forecasting with transformer and reinforcement learning,” *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[14] B. Lim, S. Arık, N. Loeff, and T. Pfister, “Temporal fusion transformers for interpretable multi-horizon time series forecasting,” *International Journal of Forecasting*, vol. 37, 2021. doi: https://doi.org/10.1016/j.ijforecast.2021.03.012

[15] J. Xiao, T. Deng, and S. Bi, “Comparative analysis of lstm, gru, and transformer models for stock price prediction,” in *Proceedings of the International Conference on Digital Economy, Blockchain and Artificial Intelligence*, 2024. doi: 10.1145/3700058.3700075

[16] Q. Wang and Y. Yuan, “Stock price forecast: Comparison of lstm, hmm, and transformer,” in *Proceedings of the 2nd International Academic Conference on Blockchain, Information Technology and Smart Finance (ICBIS 2023)*, 2023.

[17] R. C. Cavalcante, R. C. Brasileiro, V. L. F. Souza, J. P. Nobrega, and A. L. I. Oliveira, “Computational intelligence and financial markets: A survey and future directions,” *Soft Computing*, 2024.

[18] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016, http://www.deeplearningbook.org/.

[19] D. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” in *International Conference on Learning Representations (ICLR)*, 2015, [Online]. Available: https://arxiv.org/abs/1412.6980.

[20] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” *Journal of Machine Learning Research*, 2014, [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html.

[21] H. Zhou, S. Zhang, J. Peng, S. Zhang, G. Li, R. Xiong, and W. Zhang, “Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

[22] E. Chong, C. Han, and F. C. Park, “Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies,” *Expert Systems with Applications*, 2017.

[23] D. M. Nelson, A. C. Pereira, and R. A. de Oliveira, “Stock market’s price movement prediction with lstm neural networks,” in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2017.

[24] K. J. Kim and I. K. Kim, “Forecasting stock prices with a feature fusion lstm-cnn model using different representations of the same data,” *PLOS ONE*, 2019.