An Efficient Deep Learning Model for Stock Market Prediction:

LSTM & Technical Indicators for Short-Term Forecasting

Ayobami Makinde & Arturo Franco

Department of Computer Science and Mathematics

St Joseph’s University & Molloy University

Advisor: Dr. Helen Dang

April 28, 2025

Abstract

Stock market prediction is a complex problem due to the nonlinear and volatile nature of financial data. It is inherently a time series forecasting problem, where historical price patterns, trends, and market indicators are analyzed to predict future movements. Traditional models tend to struggle to take into account long-term dependencies. For that, deep learning techniques, particularly Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks, are valuable for improving forecast accuracy. This study applies a Deep Learning model and fine-tunes its parameters to predict stock prices. The performance of this model is also evaluated using real-world datasets and other model evaluation parameters.

Keywords: Stock Market Prediction, Deep Learning, Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), Time Series Forecasting.

1. Introduction

The function of the stock market is particularly important not only for providing a source of liquidity but to allow individuals to access another layer of investments. Being able to estimate stock price changes correctly can provide significant benefits to individuals and financial institutions. Stock markets experience unpredictable factors, including political activities, economic indicators, and investor sentiment. Forecasting with an adequate degree of certainty can prove difficult with all the input into stock markets.

Deep Learning has made exciting strides recently and now presents new opportunities in examining time series data. The time-honored approaches of using ARIMAX or linear regression models typically do not stand for nonlinear activities found in complex financial datasets well (Kim, 2003). Deep learning models, and in particular recurrent neural networks (RNNs) and long short-term memory networks (LSTMs), can learn from sequential data and possible latent patterns and dependencies across time. As noted by Fischer and Krauss (2018), "Deep learning with LSTMs provides a powerful method for extracting patterns from financial time series, leading to improved predictive performance."

This study concentrates on implementing an LSTM-based model for predicting stock prices and describing its effectiveness based on several rigorous experiments with real stock market data.

2. Deep Learning

Deep learning is a type of Machine Learning that uses neural network structures with several layers to learn hierarchical patterns in data automatically. For time series forecasting tasks, such as predicting stock prices, deep learning models benefit from automatically learning complex, nonlinear relationships without needing to do any feature engineering (Goodfellow, Bengio, & Courville, 2016).

2.1 RNN

Recurrent Neural Networks (RNNs) are designed specifically for sequential data and tasks in which the order of inputs is critical and where the dependency of inputs is time dependent. While large learning-based neural networks typically do not share parameters, RNNs share their weights across time steps, a way of formally representing an effect of time that occurs in nature. In sequential data time dependent processes, such as stock prices, how a previous value drives the subsequent more recent value is manifested in the state, or hidden state, of the neural network. The network can only operate with one input to generate a state and return a new hidden state. If RNNs made separate connections for each input, the number of parameters would grow significantly, making the model inefficient to tune. RNNs solve by sharing a set of weights across all frames in a sequence; it provides a more efficient structural category to learn from a more flexible and unconstrained serial data stream that can be arbitrarily long or complex.

RNNs have significant drawbacks when learning from long sequences, despite their strengths. One of the greatest problems is the vanishing gradient problem. In back-propagation through time, the gradients will shrink exponentially and make it difficult for the network to learn long, temporal dependencies. The opposite of the vanishing gradient problem is a similar problem called exploding gradient problem, in which the gradients will explode and grow too large, and it becomes hard to stabilize the training of the model. These issues make standard RNNs not particularly useful for problems like stock market prediction that require long-term patterns to be established and acknowledged (Goodfellow, Bengio, & Courville, 2016).

2.2 LSTM

Long Short-Term Memory (LSTM) networks are a more advanced version of RNNs that utilize memory cells and gates to learn long-term dependencies, something traditional RNNs struggled to do. LSTMs provide a more complex architecture to RNNs, such as memory cells and gating functions, which allow for selective retention and forgetting of information over longer periods of time. Each LSTM cell has three main gates. The first one deals with forgetting information (forget gate). The second one deals with how new information is added (input gate). The last one handles what gets passed on as information to the next hidden state (output gate). Through these gates, LSTMs can keep track of important information and reject unimportant information across many time steps, allowing trends and patterns to appear in time series data, as with stock prices.

The greatest strength of LSTMs is they can learn long-term patterns of data without the vanishing, exploding gradient problems that plagued previous models. They can map complex, nonlinear relationships in time series data, especially valuable in predicting financial trends such as pricing in stock market data. Moreover, they can easily accommodate multivariate input so that models can include any of the following features: opening prices, trading volume, or technical indicators. This adaptability and robustness make them a very wonderful way to capture the unpredictable and intricate movements of stock markets (Nelson, Pereira, & de Oliveira, 2017).

2.3 Self-Attention

The self-attention mechanism is a new innovation that solves the challenges posed by sequential models like RNNs and LSTMs to capture long-term dependencies. In classical recurrent models, all steps are computed sequentially, which can lead to inefficiencies when only trying to model very long sequences. Self-attention permits models to attend directly to all positions in a sequence jointly, measuring the relative weighted importance of each element in relation to the others. This ability to define dependencies without regard to distance enhances the modeling of financially relevant complex temporal structures. This includes financial time series behaviors such as sudden market shocks, delayed market reactions to news, and seasonal patterns (Goodfellow, Bengio, & Courville, 2016).

At the heart of Transformer models is self-attention, allowing them to outperform recurrent models in a variety of sequential prediction tasks. Regarding stock market prediction, the self-attention models applied in this research permitted the weight of certain historical price movements, macroeconomic events, and shifts in investor sentiment to focus the recently collected data based on their importance and how far back they were in time. This is a key advantage in volatile markets, where recent and older events influence the current price as well. Moreover, self-attention enables better parallelization during training to converge faster than RNN-based approaches (Vaswani et al. 2017). As stock market datasets tend to become increasingly complex and larger, attention-based models may present a means for future research in financial forecasting.

3. Dataset and methodology

3.1 Data Collection

The stock market data used in this project was retrieved using the yfinance Python library. The library gave access to historical financial data, which was needed to analyze stocks. What was focused on was daily closing price data from January 2020 to April 2025 for three stocks being Microsoft (MSFT), Tesla (TSLA), and American Airlines (AAL). These tickers were selected due to their varying volatility profiles and market behaviors as of recently. This unpredictability was also supported by the findings from Patel et al. (2015), who explained how market volatility affects predictive modeling. This was seen firsthand in this model, with accuracy being drastically improved when running the model with them. The parameters consisted of “indicators like RSI, MACD, and Bollinger Bands.” They “were found to be effective in capturing momentum and mean-reverting trends in stock data." These indicators, being able to help the model capture this information, were key to having a more accurate model.

3.2 Feature Engineering

To make sure a model can perform optimally, though, all numerical features were scaled to a range that was normalized between 0 and 1. This was because each feature has a different scale effect, so “To eliminate the scale effect among input variables, [you conduct] normalization … when applying machine learning techniques to financial time series," like this instance (Kim, 2003, p. 309). This was done using the MinMaxScaler from the imported package Scikit-learn. The process is also known as feature scaling. However, the reason the process of “MinMax normalization [is key, is because it] …helps prevent features with larger ranges from dominating model training and accelerates convergence" (Zhang & Zhou, 2020, p. 3). This means that the normalization process makes sure that there is a faster convergence to prevent one feature from dominating others due to the scale differences they have. That leaves you with the final dataset, which consists of five features. The list was made up of Close, RSI, MACD, Bollinger High, and Bollinger Low. Any missing values from technical indicators were removed to maintain data consistency for further steps along the process.

3.3 Preprocessing

After normalizing the data, it was split into an 80 to 20 ratio, with 80 percent used for training and 20 percent reserved for testing. Then it was decided to use a "sliding window approach to generate input sequences, allowing the LSTM to learn complex temporal patterns effectively" (Nelson et al., 2017, p. 1421). This meant the sliding window approach would generate training samples, and "Each sample comprises the returns of the past 100 trading days as inputs, and the directional return on the next day” (Fischer & Krauss, 2018, p. 661). The target was the closing price on the following day. This rolling structure allowed the model to learn from hundreds of overlapping windows of market behavior.

4. Results and Discussions

4.1 Model Performance

The model's performance was evaluated based on three key metrics being Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Test Loss. For each metric, it was evaluated across the three different stocks, being American Airlines (AAL), Tesla (TSLA), and Microsoft (MSFT). Overall, the results demonstrate strong predictive capability, especially for AAL and MSFT. The model achieved varying numbers for TSLA. The results obtained are recorded in table 1.

Table 1. MAE, RMSE and Test Loss collected during the prediction of AAL, TSLA, MSFT values using DL network for the duration of 100 days.

Company	MAE	RMSE	Test Loss
AAL	0.053	0.065	0.004
TSLA	0.163	0.168	0.029
MSFT	0.043	0.056	0.003

These results suggest that the model is making relatively low errors in certain stock price predictions. For AAL, the model achieved an MAE of 0.053, RMSE of 0.065, and a Test Loss of 0.004. Then for MSFT results, they were strong, with an MAE of 0.043, RMSE of 0.056, and a Test Loss of 0.003. The "lower MAE and RMSE values indicate that a forecasting model captures more accurate and consistent patterns in financial time series" (Kim, 2003, p. 311). Also, the very low Test Loss values for AAL and MSFT further confirm that the model successfully learned meaningful patterns during training without overfitting. On the other hand, Tesla (TSLA) had a higher MAE of 0.163 and RMSE of 0.168, with a Test Loss of 0.029, reflecting the higher volatility which is often associated with TSLA's stock behavior as of the recent months. "Tesla's stock exhibited high volatility driven by a combination of earnings surprises, regulatory news, and CEO-related public statements" (Zhang & Weng, 2022, p. 10). This explains the higher error metrics observed for TSLA in Table 1. In general, “good forecasting models are characterized by minimal test loss, indicating robust learning rather than overfitting” (Zhang & Zhou, 2020, p. 5). So, the low-test loss implies that the model generalizes well and does not just memorize the training data.

4.2 Model Stability

Model stability was a crucial factor in this study to prevent overfitting as well. The 100-day period set for each stock would make for a lot of noise to run through the model. Therefore, EarlyStopping was implemented to halt training once the validation loss stopped improving. This feature helped maintain a balance between underfitting and overfitting while running the model. Additionally, Dropout (rate 0.2) and Batch Normalization layers were also incorporated into the architecture. Dropout randomly turns off neurons during training, which forces the model to not fully rely on specific paths in the network. “Dropout [also] improves the generalization of neural networks by preventing co-adaptation of feature detectors” (Srivastava et al., 2014, p. 2). Batch Normalization further aids in stabilizing and accelerating the training by standardizing the inputs to each layer. It also “reduces internal covariate shift, thus enhancing model stability and allowing higher learning rates” (Ioffe and Szegedy, 2015, p. 3). These implementations helped the model from overshooting or undershooting predictions.

4.3 Impact of Technical Indicators

Having technical indicators like RSI, MACD, and Bollinger Bands proved to be an extremely helpful contributor to improved model accuracy. In the first iterations of the model, these features were not within the scope of the model, but after seeing the inaccuracies of the results, changes were necessary. These indicators allowed the model to capture more aspects of price behavior beyond simple closing values. Aspects such as market momentum, trend reversals, and volatility boundaries were now being viewed when running. Such features were previously not recognized. However, after research, it was learned that “the integration of technical indicators significantly enhances the ability of machine learning models to predict stock market behavior” (Patel et al., 2015, p. 262). To be specific, the Bollinger Bands helped in modeling mean-reverting tendencies, while RSI and MACD helped analyze momentum shifts. This demonstrated that for effective feature engineering, that meant creating more meaningful inputs. Therefore, that made a boost to the model’s capabilities for the financial forecasting tasks.

4.4 Challenges and Limitations

While the model achieved strong performance metrics, several limitations were encountered during the project. First, the model would have improved with sentiment for each of the respective stocks. However, due to financial reasons, the free versions could not be used for such a long period, like the one we were using, being 100 days. This led to struggles for the model to predict prices accurately when sudden news-driven events occurred, highlighting a weakness in capturing non-technical influences. For any model trying to make such predictions, "stock prices are not solely determined by historical data and technical patterns but can be heavily influenced by external, unpredictable factors" (Nelson et al., 2017, p. 1422). Additionally, the model only used technical indicators and did not use fundamental data such as earnings reports or economic news. This also limited its ability to take into account sharp market shifts. Overall, the model performs best under standard and simple market conditions. For it to be used in the real world accurately, it would need news sentiment analysis at the very least to handle real-world volatility. The year 2025 is a prime example of needing this analysis, due to the drastic economic news weekly.

5. Visualizations

5.1 Stock Price Movements (Candlestick Charts)

To better understand the behavior of the stocks that are being analyzed, candlestick charts were used (based on historical daily data) from November 2024 to April 2025. Candlestick visualizations are good for the observation of stock prices' volatility, trends, and trading volume from that period. As shown in Figure 1, all three stocks' candlestick charts reveal periods of strong volatility during early 2025.

Figure 1. AAL, MSFT, and TSLA Candlestick Charts all between Nov 2024 to Apr 2025.

American Airlines (AAL) exhibited a generally declining trend throughout the observed period, with occasional sharp rebounds. The candlestick chart shows several periods of increased volatility, shown by larger candle bodies. This volatility adds more complexity to the forecasting task, though the model still achieved strong performance metrics on AAL compared to other stocks.

Microsoft (MSFT) showed a more stable and predictable pattern. The candlestick chart displays smaller daily fluctuations and fewer sudden shifts compared to AAL and TSLA. This greater stability likely contributed to the model achieving the lowest MAE (0.043) and RMSE (0.056) among the three stocks.

Tesla (TSLA) showed plenty of volatility during this period. The candlestick chart clearly shows wide price changes, gaps, and higher trading volumes. As stated before, Tesla is known to be very volatile as of recent. This behavior aligns with our model's higher prediction error on TSLA, reflecting the difficulty in forecasting such erratic stock movements.

5.2 Four-Day Price Forecasts (Prediction Graphs)

In addition to analyzing historical performance, the model was tasked with predicting the next four trading days for each stock. As shown in Figure 2, the four-day forecasted closing prices for AAL, MSFT, and TSLA are presented.

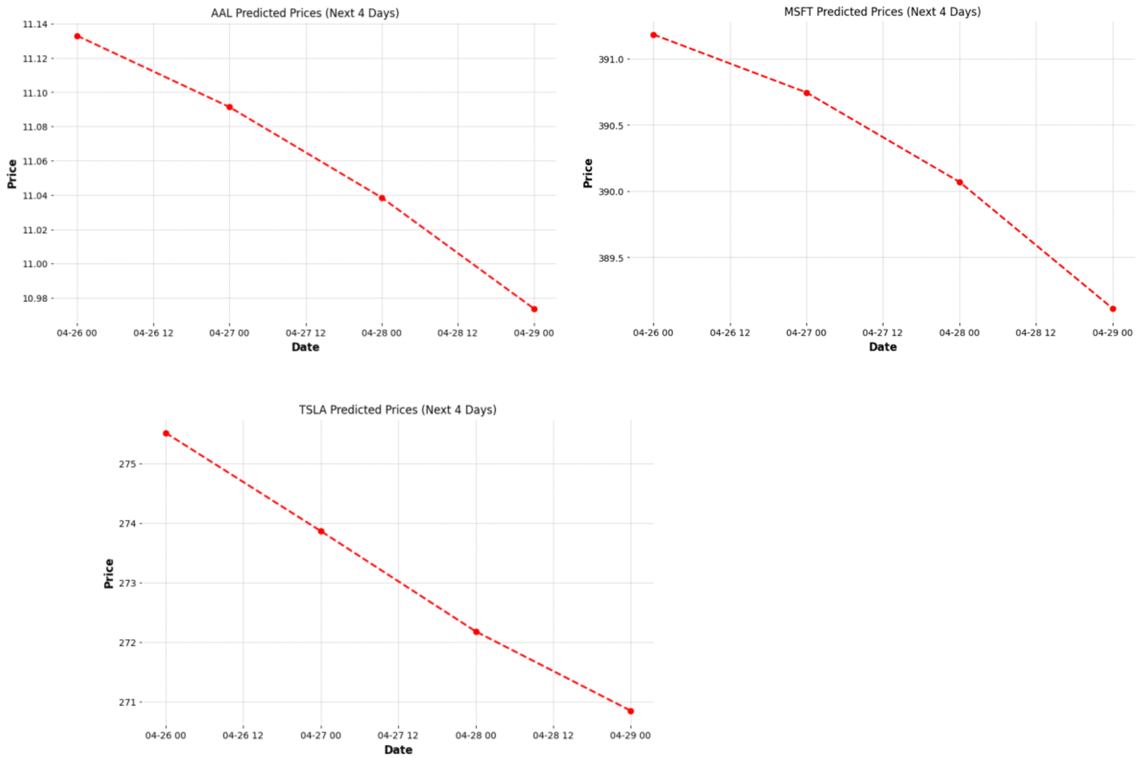

Figure 2. AAL, MSFT, and TSLA predicted four-day forecast numbers from Apr 2025.

The predicted closing prices for AAL show a slowing downward trend over the four-day forecast horizon. This suggests that the model detected a slight stabilization in AAL's recent trading behavior, consistent with the candlestick patterns observed earlier.

Microsoft's 4-day prediction maintains a relatively flat trend, with little deviation between forecasted points. This shows MSFT's historical stability and explains the strong predictive accuracy achieved by the model.

Tesla's 4-day forecast predicts a small but consistent downward trend. However, due to TSLA's historical high volatility as seen on Figure 1, prediction confidence is lower. This matches the larger MAE and RMSE values recorded for TSLA from Table 1, further showing the difficulty in predicting with high volatility.

6. Conclusion

This study illustrates that deep learning models such as LSTMs, enhanced with technical indicators and preferably stabilization techniques (Dropout and Batch Normalization), offer promising stock price prediction in agitated financial markets. The predictive ability of the model was good especially for stocks with better-behaved performance like Microsoft (MSFT), and American Airlines (AAL). Tesla (TSLA) provided more of a challenge because of its comparative volatility, nonetheless, the model adequately captured general trends. This demonstrated the effectiveness of time series forecasting using LSTM-based approaches. The model's accuracy improves significantly by adding technical indicators such as RSI, MACD, and Bollinger Bands that provide much richer feature set than the closing price alone.

While we made important contributions in conducting this project, our work revealed some important limitations. For example, we struggled to adapt to real time news events, as well as integrate essential data substructures into the financial transaction data. This discussion leads us to a few important directions moving forward, including sentiment analysis, news impacts, and attention networks like Transformers to enhance predictive accuracy given the complexities of the real world situations we are operating in. Therefore, in summary, we have reinforced the emergent potential for deep learning in financial market prediction and provided a basic foundation for a more complete stock market forecast system in the future.

References

(1) Fischer, T., & Krauss, C. (2018). *Deep learning with long short-term memory networks for financial market predictions*. European Journal of Operational Research, 270(2), 654–669. https://doi.org/10.1016/j.ejor.2017.11.054

(2) Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press. http://www.deeplearningbook.org

(3) Ioffe, S., & Szegedy, C. (2015). *Batch normalization: Accelerating deep network training by reducing internal covariate shift.* Proceedings of the 32nd International Conference on Machine Learning (ICML-15), 448-456. http://proceedings.mlr.press/v37/ioffe15.html

(4) Kim, H. Y. (2003). *Financial time series forecasting using support vector machines*. Neurocomputing, 55(1-2), 307–319. https://doi.org/10.1016/S0925-2312(03)00372-2

(5) Nelson, D. M., Pereira, A. C. M., & de Oliveira, R. A. (2017). *Stock market's price movement prediction with LSTM neural networks*. International Joint Conference on Neural Networks (IJCNN), 1419–1426. https://doi.org/10.1109/IJCNN.2017.7966019

(6) Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). *Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques*. Expert Systems with Applications, 42(1), 259–268. https://doi.org/10.1016/j.eswa.2014.07.040

(7) Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). *Dropout: A simple way to prevent neural networks from overfitting*. Journal of Machine Learning Research, 15(1), 1929-1958. http://jmlr.org/papers/v15/srivastava14a.html

(8) Zhang, J., & Zhou, Y. (2020). *Stock market prediction via multi-source multiple instance learning*. Expert Systems with Applications, 141, 112963. https://doi.org/10.1016/j.eswa.2019.112963

(9) Zhang, X., & Weng, W. (2022). *Stock Price Volatility Analysis of Tesla: A Comprehensive Study Using Event-Driven Data*. Journal of Risk and Financial Management, 15(6), 238. https://doi.org/10.3390/jrfm15060238