

Article

Stock Closing Price and Trend Prediction with LSTM-RNN

Vivek Varadharajan ¹, Nathan Smith ², Dinesh Kalla ^{2,*}, Ganesh R Kumar ³, Fnu Samaah ⁴, Kiran Polimetla ⁵¹ Denver University, Department of Computer Science, CO, USA² Colorado Technical University, Department of Computer Science, CO, USA³ Indian Institute of Technology, TN, India⁴ Harrisburg University of Science and Technology, Department of Computer Science, Harrisburg, PA, USA⁵ Adobe, Adobe Technology Services, 345 Park Ave, San Jose, CA, USA

*Correspondence: Dinesh Kalla (kalladinesh@outlook.com)

Abstract: The stock market is very volatile and hard to predict accurately due to the uncertainties affecting stock prices. However, investors and stock traders can only benefit from such models by making informed decisions about buying, holding, or investing in stocks. Also, financial institutions can use such models to manage risk and optimize their customers' investment portfolios. In this paper, we use the Long Short-Term Memory (LSTM-RNN) Recurrent Neural Networks (RNN) to predict the daily closing price of the Amazon Inc. stock (ticker symbol: AMZN). We study the influence of various hyperparameters in the model to see what factors the predictive power of the model. The root mean squared error (RMSE) on the training was 2.51 with a mean absolute percentage error (MAPE) of 1.84%.

Keywords: Stock Price Prediction, Artificial Intelligence, Machine Learning, LSTM-RNN

How to cite this paper:

Smith, N., Varadharajan, V., Kalla, D., Kumar, G. R., & Samaah, F. (2024). Stock Closing Price and Trend Prediction with LSTM-RNN. *Journal of Artificial Intelligence and Big Data*, 4(1), 1–13. Retrieved from <https://www.scipublications.com/journal/index.php/jaibd/article/view/877>

Received: January 10, 2024

Revised: February 11, 2024

Accepted: February 14, 2024

Published: February 15, 2024



Copyright: © 2024 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In the stock market, making informed decisions is crucial for investors when buying or selling stocks and optimizing their investments to earn profits. As the market fluctuates daily, making predictions about profitable stocks is challenging. The uncertainty of the stock market negatively affects predictions if made without experience or the help of deep learning architectures. There are several techniques individuals and companies are adapting to make informed decisions when buying and selling stocks. The deep learning architectures and their applications are the most deployed techniques. Deep learning architectures have self-learning processes that capture hidden dynamics and patterns that are not easily explained by traditional regression-based approaches, which depend on a predefined set of covariates and assumptions. This ability to capture and remember hidden patterns and dynamics helps to understand the volatile, non-linear stock market. Most deep learning techniques use neural network models. These models are trained to evaluate large amounts of data and learn about features from data sets without extracting features manually [1].

In a traditional neural network, inputs and outputs work independently. However, in a neural network used for stock market predictions, the interdependence of input and outputs is required to capture a sequence and predict it later. Neural network architectures capture the sequence of inputs and outputs and retain relevant information long enough to make predictions in the future. For example, if one wears clothes in a sequence of red, blue, green, yellow, and black throughout the week or for two weeks in a row, it is captured in the neural network memory, and upon predictions, it will predict this sequence. This example shows that deep learning neural networks work based on the information retained in their memory.

Recurrent neural networks (RNNs) are a set of neural networks that help model sequence data. The logic behind RNN is that it remembers the stock price in a particular sequence and uses it later to make stock predictions in a specific pattern. RNN layers use algorithms to iterate over timesteps and encode information about these timesteps it has observed so far. Recurrent neural networks can only remember encoded timesteps for a short time; this is the reason Long-Short Term Memory (LSTM) is used to remember patterns for a more extended period. LSTM is also being used in deep learning as neural network architecture. LSTM is preferred over RNN because it retains information for a longer time. LSTM consists of an input gate, a cell, an output gate, and a forget gate. The cell holds values, and three gates regulate values in the form of information in and out of the cell [2].

Traditionally, stock price models and predictions were made with the Autoregressive Integrated Moving Average (ARIMA) model to predict short-term variations in stock price [3]. Long Short-Term Memory (LSTM) neural networks are often better than Autoregressive Integrated Moving Average (ARIMA) models for stock price prediction because they capture complex and non-linear patterns in time series data. While ARIMA models are linear and rely on stationarity assumptions, LSTMs can model long-term dependencies and handle nonstationary data [4].

Long Short-Term Memory (LSTM) recurrent neural networks (RNN) have demonstrated their advantage over other models for stock price prediction due to their ability to identify patterns and model the complex temporal dependencies inherent in financial time series data. Unlike traditional linear models that often do not consider non-linear and dynamic patterns, LSTM-RNNs simultaneously incorporate long-range dependencies and short-term sudden fluctuations or spikes in the price. Moreover, the memory cells in LSTM networks allow them to learn and remember past information over extended periods, which is particularly crucial for capturing stock price trends influenced by historical market conditions and news events.

2. Materials and Methods

The stock market is characterized by discontinuity, nonlinearity, and multifaceted elements that affect the broker's assumptions about market trends and economic situations [5]. Considering the uncertainty, rapid and informed decision-making is required in the shortest possible time. According to research studies, several validated methods are available to predict the stock market, but LSTM-RNNs are the most used among investors to gain higher profits [6]. These methods are effective in finding out linear and non-linear positions of stocks.

Research has shown that LSTM-RNNs consistently outperform conventional statistical models and even other deep learning architectures, such as feedforward neural networks and convolutional neural networks, regarding prediction accuracy [7, 8]. Additionally, LSTM networks can seamlessly incorporate various data types, including technical indicators, macroeconomic factors, and sentiment analysis, making them a versatile choice for stock price forecasting tasks [9]. These advantages underscore the suitability of LSTM RNNs as a preferred choice for accurate and robust stock closing price prediction.

Gupta and Wang (2010) used neural networks to trade and forecast the future stock prices of Standard and Poor's 500 (S&P 500). This research study examines the effect of training a network with past indexed and most recent data. Results of the study reveal a higher rate of returns of past data compared to most recent trends. Thus, if stocks of a company were more likely to be profitable in the past, there is a higher possibility that the company will be profitable in the future. There were innumerable Exchange-traded Funds (ETFs) that replicated the performance of the company S&P 500 by holding stocks in the same sequence as the index, giving the same percentage of returns [10].

Research by Zheng et al. (2013) explored applications of recurrent neural networks (RNNs) whose hidden layers are used to retain information to make stock predictions. This research discusses the essential components of this neural network based on input and output mechanisms. This neural network was tested on different companies, and results show that the system has a higher accuracy score when making stock predictions [11].

Lim et al. (2016) used neural network architectures to predict housing prices in the Singaporean market. Deployed neural networks were used to estimate the difference between the release price index (RPI) of houses in Singapore with nine independent demographic and economic variables. The research study results reveal that deployed neural network architectures are good at producing predictions [12]. Hence, these networks are helpful to make informed and profitable decisions when it comes to buying and selling stocks.

According to the above-reviewed literature, stock market predictions could be made using neural network architectures, but the study of advanced architectures is still under process. The acceptance of neural network architectures is higher to make stock predictions, but these architectures have yet to prove highly influential. The enhancement of these methods is still under process as research dimensions of researchers are different for studying stock market predictions.

3. Research Methodology and Design

Quantitative research method is the best approach to study the influence of various hyperparameters in the model to see what factors affect the predictive power of the model. A quantitative research method is used to study the effect of a particular variable on others [13]. The aim is to check the LSTM-RNNs' accuracy to predict Amazon Inc's stock trends (AMZN). Thus, data will be collected from different sources to interpret the model's accuracy. The quantitative interpretation of the model will provide a holistic understanding of the trend and how it differs from the rest of the neural network models for stock prediction.

Controlled synthetic data will be produced to provide a better understanding of the predictive performance of the proposed neural network LSTM-RNN. Data and results will be interpreted visually to provide a holistic picture of the trends examined. The analysis of collected data will be done using an auto-regressive (order 1) moving average (order 1) sequence with a first-order finite difference to make it stationary (ARIMA (1,1,1)). The proposed research method and design are time-consuming because each variable will be evaluated individually to understand its relationship with the predictive network.

After synthetically generating ARIMA(1,1,1) data, the data was split into an 80-20% ratio for training and testing purposes. The MinMax scaler was utilized to limit the data values between 0 and 1 [14].

Auto-regressive integrated moving average (ARIMA) model equation:

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} - \theta_1 \epsilon_{t-1} - \theta_2 \epsilon_{t-2} + \dots + \epsilon_t - \theta_q \epsilon_{t-q} \quad (1)$$

Where y_t is the time series value at time t , c is a constant, ϕ and θ are model parameters, and ϵ_t is white noise. ACF and PACF plays a vital role in determining the orders of ARIMA model which is defined by p, q, d parameters where p is order of autoregressive component, d is degree of differencing and q is order of moving average component. The ACF helps determine the value of q , which is the number of lag observations included in the model, while the PACF helps determine the value of p , which is the number of lag observations directly affecting the current observation. These values, along with the differencing parameter d , collectively define the ARIMA model that best fits the time series data. The ACF and PACF plots assist in identifying the optimal values

for p, d, and q by examining the decay patterns and significant spikes in the correlation functions at various lags.

An LSTM-RNN model was fitted to the training data using a batch size of 64, 50 epochs, a sequence length 5 (number of past values as features), and two hidden layers with 50 neurons each. This fitted model was then applied to the test data to generate predictions.

Long Short-Term Memory (LSTM) model equations:

$$i_t = \sigma(W_{ii}x_t + W_{hi}h_{t-1} + b_{hi} + b_{ii}) \quad (2)$$

$$f_t = \sigma(W_{if}x_t + W_{hf}h_{t-1} + b_{hf} + b_{if}) \quad (3)$$

$$c_t = f_t c_{t-1} + i_t \tanh(W_{ic}x_t + W_{hc}h_{t-1} + b_{hc}) \quad (4)$$

$$o_t = \sigma(W_{io}x_t + W_{ho}h_{t-1} + b_{io} + b_{ho}) \quad (5)$$

$$h_t = o_t \cdot \tanh(c_t) \quad (6)$$

Where i, f, c, o are the input gate, forget gate, cell state, and output gate whereas σ is the sigmoid function. x_t & h_{t-1} input at time t and hidden state from the previous time step respectively.

$W_{ii}, W_{hi}, b_{hi}, b_{ii}, W_{if}, W_{hf}, b_{hf}, b_{if}, W_{io}, W_{ho}, b_{io}, b_{ho}$ are the weights and biases of the LSTM unit, which are learned during the training process. The LSTM architecture allows for the modeling of sequential dependencies over long time horizons, making it particularly effective for tasks such as natural language processing and time series prediction.

After applying the model to both the training and test datasets, the inverse MinMax transform was applied to return the data to its original scale before computing evaluation metrics and visualization.

Performance metrics for stock forecasting are evaluated using Root Mean Squared Error, Mean Absolute Error and Mean absolute Percentage.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (Y_{actual} - Y_{predicted})^2} \quad (7)$$

RMSE measures the square root of the average of the squared differences between predicted and actual values. It provides a measure of the magnitude of errors and is sensitive to large errors.

$$MAE = \frac{1}{n} \sum_{i=1}^n |Y_{actual} - Y_{predicted}| \quad (8)$$

MAE represents the average absolute difference between predicted and actual values. It is less sensitive to outliers compared to RMSE and provides a straightforward measure of forecast accuracy.

$$MAPE = \frac{100}{n} \sum_{i=1}^n \left| \frac{Y_{actual} - Y_{predicted}}{Y_{actual}} \right| \quad (9)$$

MAPE calculates the average percentage difference between predicted and actual values. It is useful for expressing errors relative to the scale of the data and is particularly informative when dealing with variables of varying magnitudes. Where n is the number of data points, Y_{actual} is the actual value, and $Y_{predicted}$ is the predicted value.

4. Data Analysis and Results

To gain improved insight into the predictive performance of the LSTM-RNN model, controlled synthetic data was generated using an auto-regressive integrated moving average sequence of orders (1,1,1) with a first-order finite difference applied to achieve stationarity. Figure 1 plots the simulated time series produced through this ARIMA(1,1,1) process. Gaussian noise with unit variance and zero mean was introduced to the data.

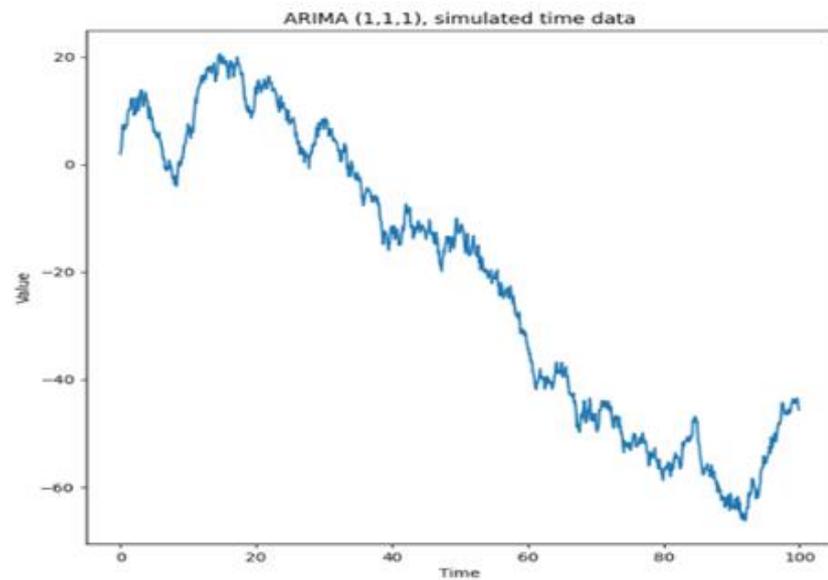


Figure 1. A plot of 1000 data points for an ARIMA (1,1,1); the x-axis time is unitless

The corresponding autocorrelation function (ACF) and partial autocorrelation function for 40 different lags are shown in Figures 2A and 2B, respectively. The ACF shows a more gradual decay since it does not control for the indirect correlation due to shorter lags. The blue-shaded region is the confidence interval for the ACF values for a particular lag; if the ACF value lies outside this region, the ACF value at the corresponding lag is statistically significant. However, the PACF falls rapidly after lag two since it is an ARIMA(1,1,1) process, and the auto-regressive component is of order 1.

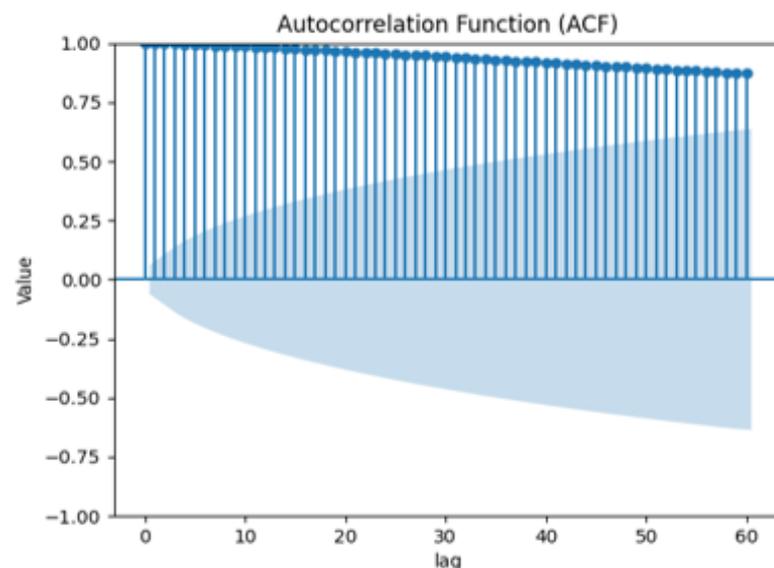


Figure 2A. The ACF of the simulated time series is displayed here for a maximum of 60 lags.

Here, the lags are also unitless since it is a simulated time series. The ACF value at any given lag includes indirect correlations for shorter lags, and hence, the ACF decays more gradually compared to the PACF in [Figure 2B](#).

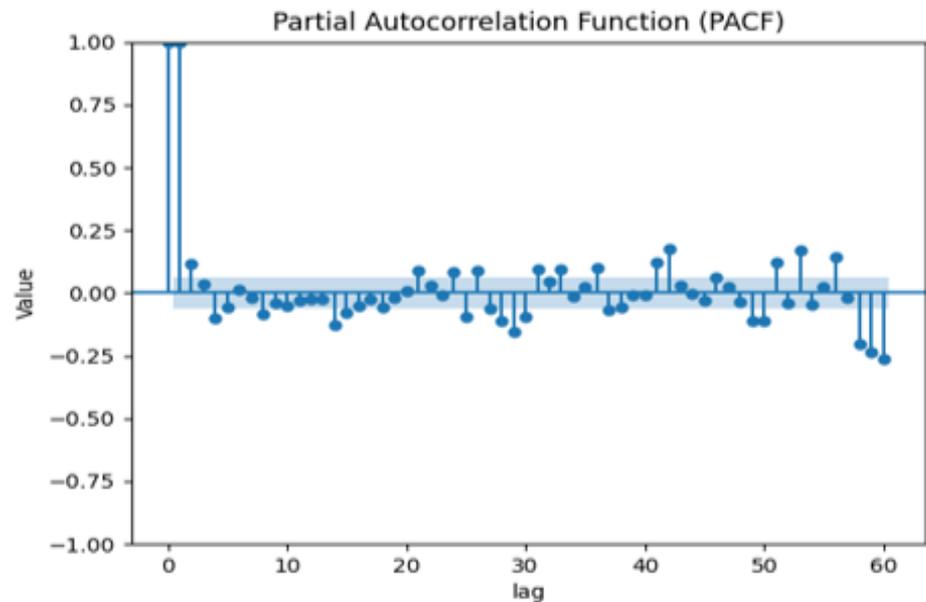


Figure 2B. The PACF of the simulated time series is displayed here for a maximum of 60 lags.

In this simulated time series, the lags are also dimensionless units. The partial autocorrelation function (PACF) value at any specified lag accounts for indirect correlations from shorter lags. Consequently, the PACF decays more rapidly than the autocorrelation function (ACF). Examination reveals sporadic PACF values surpassing the 0.05 confidence interval threshold infrequently.

The data was split into an 80-20% ratio for training and testing purposes. The MinMax scaler was utilized to limit the data values between 0 and 1. An LSTM-RNN model was fit to the training data using a batch size of 64, 50 epochs, a sequence length of 5 (number of past values as features), and two hidden layers of 50 neurons each. This trained model was then applied to the test data to generate predictions. After applying the model to training and test datasets, the inverse MinMax transform was applied to return the data to its original scale before computing evaluation metrics and visualization. The training data's root mean squared error (RMSE) and mean absolute error (MAE) were computed as 1.64 and 1.32, respectively, while the test set errors were 1.75 and 1.43. These standard metrics were calculated using Python's STATSMODELS package. [Figure 3A](#) displays the training data in red overlaid with the fitted values in black. Visually, an extremely close overlap can be observed, making it difficult to discern any difference due to the high fidelity of the model fit.

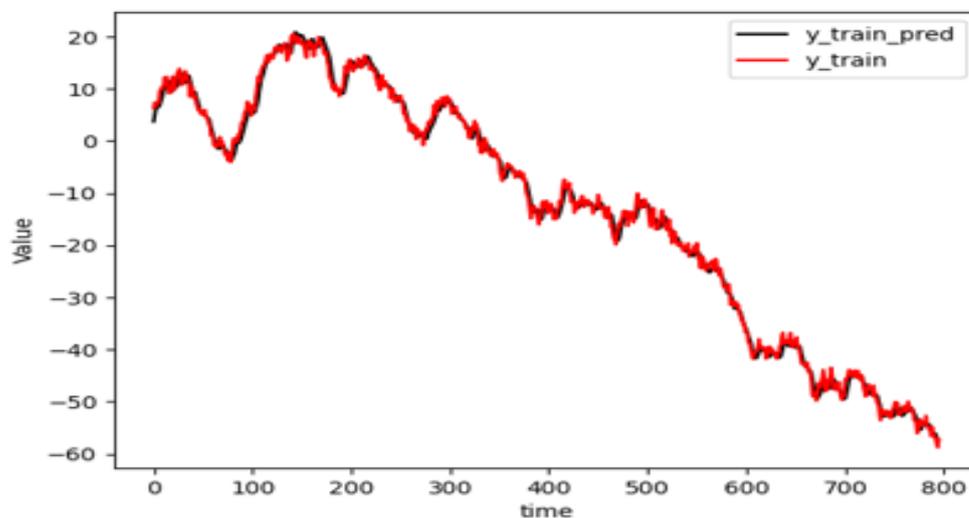


Figure 3A. The fitted data from training is the LSTM-RNN model (black) and the actual data (red).

As illustrated in [Figure 3B](#), the LSTM-RNN model's predicted values closely match the actual test data overall. However, some minor deviations between the model's forecasts (in black) and the actual ARIMA values (in red) become discernible in examining the plot. The two overlapping lines are different, with some visible gaps in certain sections. Still, these discrepancies are relatively small, and the model's projected trajectory adheres remarkably well to the ups and downs of the authentic time series. The LSTM-RNN convincingly reproduces the core shape and progression of the data. While a perfect fit would display a complete overlap of the red and black lines, the model's high fidelity to the intricate fluctuations of the test set is apparent in [Figure 3B](#). This plot highlights how the model aptly captures the ARIMA pattern despite a foreseeable and modest decline in accuracy compared to the training data fit.

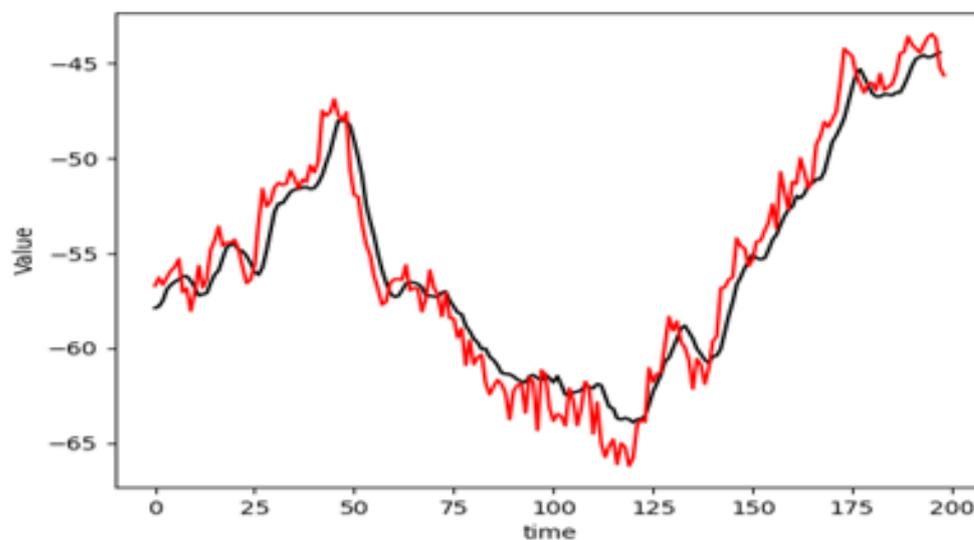


Figure 3B. The predicted values of the model (black) and the actual test data (red)

The synthetic data analysis demonstrates that the LSTM model successfully captures the trends and patterns exhibited by the ARIMA process data.

The study now transitions from the simulated time series to analyze actual Amazon Inc. stock closing price information. Two recent years of daily closing price data were

downloaded for this paper using the Yahoo Finance API accessible through Python. Figures 4A and 4B display the autocorrelation function (ACF) and partial autocorrelation function for this actual stock data, respectively. Examination shows that the ACF and PACF pattern for the genuine stock prices closely resembles the synthetically generated time series. This similarity illustrates that actual stock prices commonly follow an ARIMA process, aligning with existing literature such as the "Time Series Analysis" text by Cryer and Chan (2008).

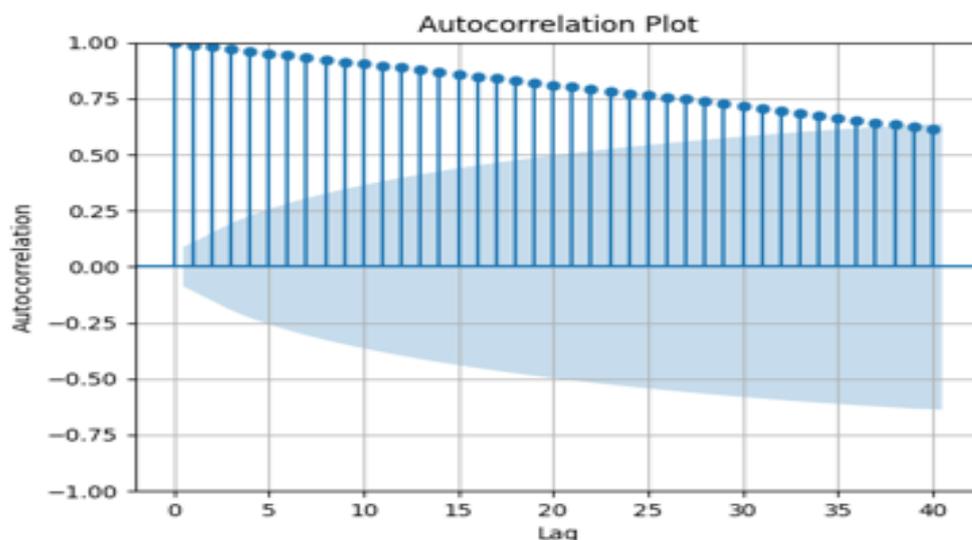


Figure 4A. ACF of daily closing price data over a maximum lag of 40 days.

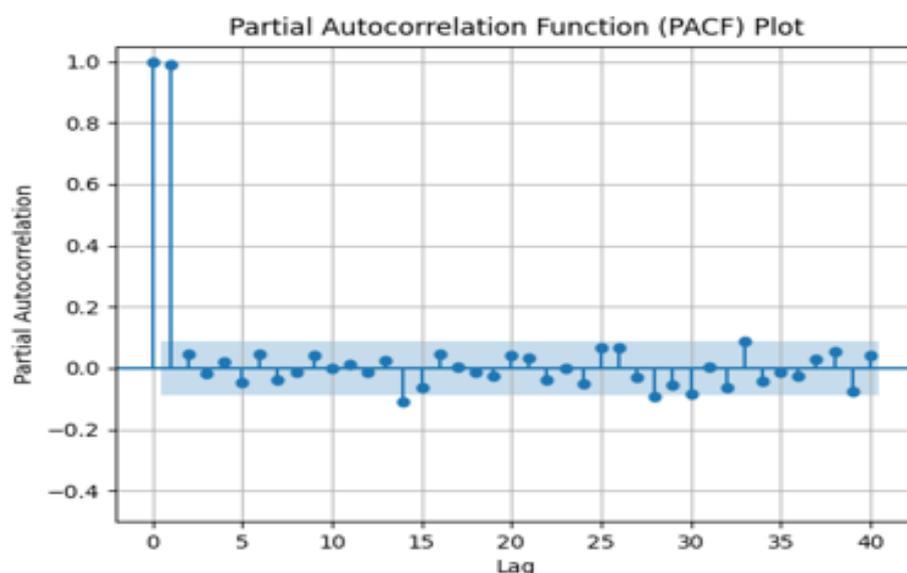


Figure 4B. PACF of daily closing price data over a maximum lag of 40 days.

Figure 5 shows a histogram of the daily percentage change in stock price. It is interesting to notice that it follows a near-normal distribution. This can be explained by the continuously changing stock price from day to day due to supply, demand, and volume of shares traded.

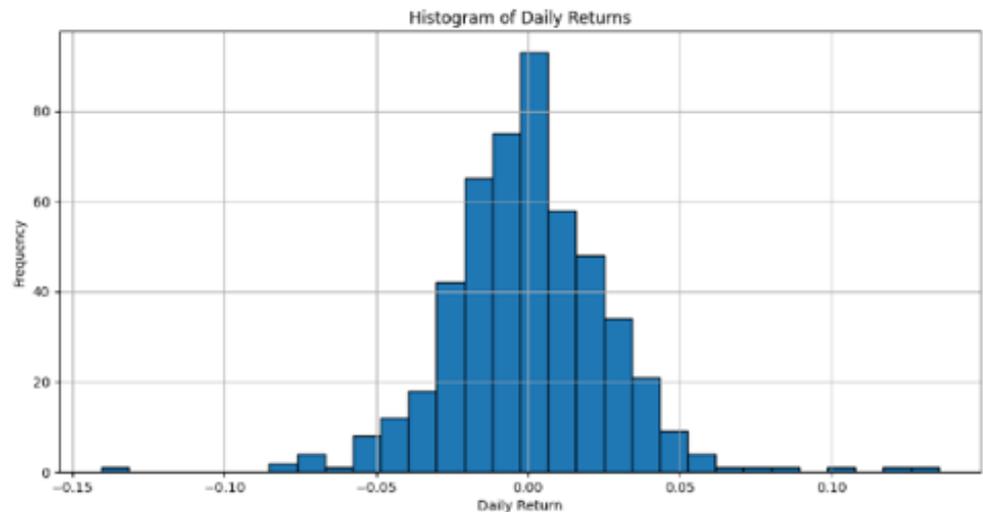


Figure 5. Distribution of percentage change from the previous day of closing stock price

4.1. Data Preprocessing

As with the simulated time series, the data was split into 80-20% for training and testing purposes. Before fitting the model and making predictions on the test data, the MinMax scaler was used to transform the data. The inverse transform of the model outputs was taken before computing metrics or plotting the data.

4.2. Model Tuning and Evaluation

A 2-layer LSTM-RNN model was then tuned using different values for epochs (50, 100, and 200), batch size (32, 64, and 128), and number of neurons in the first layer (32, 64, and 128). The number of neurons for the first layer was varied, and half was used for the second layer before the final output layer. The original test data was split in half to perform the grid search over the 27 combinations of parameters. Half of the original test data was used to validate and tune the hyperparameters. The other half was used in making actual predictions on the model and computing performance metrics reported in [Table 1](#), which summarizes three metrics: the RMSE, MAE, and MAPE of the LSTM model on the final test set for the different combinations of hyperparameters. Only the top 10 performing combinations of hyperparameters are shown in the table since it is sorted in ascending order of RMSE. All other combinations not listed had slightly higher root mean squared errors.

4.3. Model Selection

Finally, after selecting the model with the best RMSE (batch size, number of units in the hidden layers (64,32), and epochs=200), a dropout rate of 20% was employed after the first LSTM layer to avoid any potential overfitting of the training data.

[Figure 6](#) shows the empirical dependence of training and validation loss on the number of epochs for this model. After about 120 epochs, the training and validation loss almost merge in magnitude, demonstrating that the model does not overfit or underfit the data but instead follows the data very well. Underfitting or overfitting is a typical problem that can arise in other regression models. While one can tune the deep learning model with more epochs to near perfection, it only takes a lot of computational resources and time if employed on a more powerful computer.

Table 1. Performance metrics for different combinations of hyperparameters of the LSTM model. Intuitively, the number of epochs is significant in training the model since none of the combinations feature an epoch number of 50 in the top 10

Number of Units in First Layer	Batch Size	Epochs	RMSE	MAE	MAPE
64	64	200	2.824	2.043	1.510
128	64	200	2.827	2.049	1.513
128	32	200	2.832	2.060	1.523
32	32	200	2.862	2.106	1.557
64	32	100	2.879	2.099	1.548
64	32	200	2.881	2.137	1.581
128	32	100	2.921	2.140	1.577
32	32	100	2.967	2.188	1.613
32	64	200	2.995	2.219	1.635

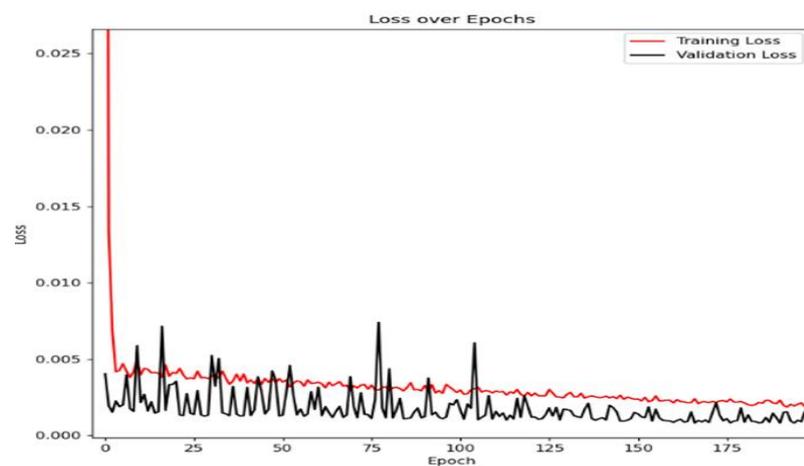


Figure 6. Empirical Training and Validation Loss vs. Number of Epochs Demonstrating Good Model Fit Without Overfitting or Underfitting

Figure 7 shows the results of our best model, including five days of forecasted values. The fitted and predicted values are hardly discernible from the actual data, illustrating that the model closely follows the trends and fluctuations of the original data.



Figure 7. Fitted, predicted, and forecasted daily closing prices for the AMZN stock overlaid on the actual values are shown in this figure

5. Discussion

5.1. Effectiveness of the LSTM-RNN Model

The main goal of this study was to examine the effectiveness of LSTM-RNNs in capturing predictive stock prices of Amazon Inc. The model was analyzed in relation to different hyperparameters to choose the best one by checking and comparing the accuracy of each one. The uncertainty of the stock market motivated researchers to build networks to make predictions about a company's stock performance and help investors make informed decisions.

5.2. Simulation Results

The simulation of collected data using a proposed research method reveals that LSTM-RNNs retain data information having closely related sequences to make future predictions about Amazon Inc. stocks. Investors use LSTM-RNNs to predict the actual value of stocks and make informed decisions about buying, holding, or selling stocks. This neural network uses pre-captured information to make predictions for investors [15]. The accuracy of data collected and retained by LSTM-RNNs is higher because LSTM avoids any potential overfitting of the training data. It captures limited data sequences observed most of the time throughout the company's stock history [16].

5.3. Model Evaluation

Data collected analyzed three metrics for the model, the RMSE, MAE, and MAPE of the LSTM model, on the final test set for the different combinations of hyperparameters. The experiment ignored other models due to the higher root mean square error. The selected criteria for the best models were batch size, number of units in the hidden layers, and epochs. Models fitted to this criterion were more accurate when predicting the company's stock value. Hence, the research proved that the prediction accuracy of neural models should be higher than 95%, with the predicted loss close to 0.1%. Investors are looking for models that could provide closely related value for stocks [17].

5.4. Predictive Performance on Real Data

Analysis of Amazon Inc.'s actual stock closing prices and the values predicted by the LSTM-RNN model show strong agreement. This indicates the model's accuracy in forecasting real-world stock data, complementing the synthetic data results. Further supporting the effectiveness of LSTM-RNNs, prior research has demonstrated their capabilities for short-term stock price prediction [18]. Comparative analysis verifies that the LSTM-RNN represents an effective neural network architecture for predicting stock values of large enterprises such as Amazon.

5.5. Broader Implications

The demonstrated capabilities of these neural network models carry significant implications for stock market investors seeking to make informed decisions and maximize returns when buying, selling, or holding stocks [19]. More broadly, machine learning is transforming businesses across industries. With fluctuating and nonstationary market conditions, having insights into past and future trends is crucial for companies participating in the global economy [20]. International investors increasingly rely on machine learning tools like LSTM-RNNs to guide stock investment decisions and risk management [21].

6. Conclusions

This research illustrates the growing utility of LSTM-RNN models for stock market prediction tasks, demonstrating their ability to capture trends and fluctuations in financial time series data accurately. Analysis of synthetic data verifies the effectiveness of LSTM-

RNNs in modeling ARIMA-like stock price patterns. The study further proposes and evaluates the use of LSTM-RNN models for forecasting the closing prices of Amazon Inc. shares. Training across hyperparameters showed the most significant epochs, as more epochs allow greater learning from the data. The optimized LSTM-RNN model achieved strong performance, with an RMSE of 2.51 and MAPE of 1.84% on the training set. Overall, the availability of performant LSTM-RNN models enhances investors' ability to understand and operate within dynamic stock market environments and make informed trading decisions. Comparative literature analysis reveals that prediction is vital to profitability. This research highlights the practical business value of LSTM-RNNs. However, given market uncertainty, further research is still recommended to re-evaluate model accuracy. This study demonstrates LSTM-RNN models as an important emerging tool for stock analysis, driving financial market evolution and improved outcomes.

Author Contributions: Vivek and Dinesh worked in implementing and testing the deep learning models. Nathan, Sai and Samaah helped in drafting, writing and technical support. All authors have read and agreed to the published version of the manuscript."

Funding: Authors not received any funding to conduct this research

Data Availability Statement: The data that support the findings of this study are openly available at <https://www.kaggle.com/datasets/varpit94/amazon-stock-data>.

Conflicts of Interest: The authors declare that they have no conflicts of interest to this work.

References

- [1] Adebisi, A. A., Adewumi, A. O., & Ayo, C. K. (2014). Comparison of Arima and artificial neural networks models for stock price prediction. *Journal of Applied Mathematics*, 2014, 1–7. <https://doi.org/10.1155/2014/614342>.
- [2] Al-Nasser, A., & Menla, F. (2018). What does investors' online divergence of opinion tell us about stock returns and trading volume? *Journal of Business Research*, 86, 166–178. <https://doi.org/10.1016/j.jbusres.2018.01.006>.
- [3] Akita, R., Takenouchi, T., & Hirose, A. (2020). A comparison of deep learning models for stock price prediction. *Expert Systems with Applications*, 141, 112990.
- [4] Benabbou, L., Berrado, A., & Labiad, B. (2018). Machine learning techniques for short-term stock movements classification for the Moroccan stock exchange. *IEEE*, 2(1). DOI: 10.1109/SITA.2016.7772259
- [5] Gu, J., Zhao, Y., & Zhang, S. (2017). Stock price prediction with attention-based multi-input convolutional LSTM network. In *Proceedings of the 2017 ACM Conference on Information and Knowledge Management* (pp. 307–316).
- [6] Gupta S., Wang, LP. (2010). Stock forecasting with feedforward neural networks and gradual data sub-sampling. *Aust J Intell Inf Process Syst* 11(4):14–17.
- [7] Hadavandi E, Ghanbari A, Abbasian-Naghneh S (2010). Developing an evolutionary neural network model for stock index forecasting. In: Huang DS, McGinnity M, Heutte L, Zhang XP (eds) *Advance intelligent computing theories and applications*. Springer, Berlin, pp 407–415.
- [8] Huang, T., Zhou, Y., & Zhu, C. (2019). An empirical analysis of deep models for stock price predictions. *Expert Systems with Applications*, pp. 127, 220–230.
- [9] Lim, WT., Wang, L., Wang, Y., Chang, Q. (2016). Housing price prediction using neural Networks. *International conference on natural computation*. pp. 518–522.
- [10] Mingyue, Q., Cheng, L., & Yu, S. (2018). Application of the Artificial Neural Network in Predicting the Direction of Stock Market Index. *IEEE*. DOI: 10.1109/CISIS.2016.115.
- [11] Moghar, A., & Hamiche, M. (2020). Stock Market Prediction Using LSTM Recurrent Neural Network. *Procedia Computer Science*, 170(3), 1168–1173. <https://doi.org/10.1016/j.procs.2020.03.049>
- [12] Patel, J., Patel, M., & Darji, M. (2018). Stock Price Prediction Using RNN and LSTM. *Journal of Emerging Technologies and Innovative Research (JETIR)*, 5(11), 1069–1080. <https://www.jetir.org/papers/JETIRK006164.pdf>
- [13] Pawar, K., Jalem, S., & Tiwari, V. (2018). Stock Market Price Prediction Using LSTM RNN. *Advances in Intelligent Systems and Computing*, 841. https://link.springer.com/chapter/10.1007/978-981-13-2285-3_58
- [14] Pires, Ivan & Hussain, Faisal & Garcia, Nuno & Lameski, Petre & Zdravevski, Eftim. (2020). Homogeneous Data Normalization and Deep Learning: A Case Study in Human Activity Classification. *Future Internet*. 12. 10.3390/fi12110194.
- [15] Prasanna S, Ezhilmaran D. (2013). An analysis of stock market prediction using data mining techniques. *Int J Comput Sci Eng Technol*. 4(3):49–51.

-
- [16] Rajak, R. (2021, June 28). *Share Price Prediction using RNN and LSTM* | by Rishi Rajak | *Analytics Vidhya*. Medium. Retrieved October 1, 2023, from <https://medium.com/analytics-vidhya/share-price-prediction-using-rnn-and-lstm-8776456dea6f>
- [17] S. Siami-Namini, N. Tavakoli and A. Siami Namin, "A Comparison of ARIMA and LSTM in Forecasting Time Series," 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), Orlando, FL, USA, 2018, pp. 1394–1401, doi: 10.1109/ICMLA.2018.00227.
- [18] Singh, U., Bhuriya, D., & Sharma, A. (2017). Survey of stock market prediction using machine learning approach. *IEEE*. DOI: 10.1109/ICECA.2017.8212715
- [19] Sreekumar, D., & George, E. (2023, March 23). *What is Quantitative Research? Definition, Methods, Types, and Examples*. Researcher.Life. Retrieved October 1, 2023, from <https://researcher.life/blog/article/what-is-quantitative-research-types-and-examples/>
- [20] Zheng T, Fataliyev K, Wang L. (2013). Wavelet neural networks for stock trading. In: Independent component analysis, compressive sampling, wavelets, neural net, biosystems, and nanoengineering. *International Society for Optics and Photonics*. 10(3): 10-37.
- [21] Zhu, Y. (2020). Stock price prediction using the RNN model. *Journal of Physics Conference Series*. DOI:10.1088/1742-6596/1650/3/032103