Understanding the Impact of Epoch Levels on LSTM-Based Stock Price Forecasting

Sophia Wong
Gilbert Classical Academy
1016 North Burk Street, Gilbert, AZ 85234
sophiaw26750@gmail.com

Tony Du, Ph. D.
CSIS department, Bradley University, Peoria, IL
1501 W. Bradley Ave, Peoria, IL 61625
tdu@fsmail.bradley.edu

August 28, 2025

Abstract

The aim of this study is to understand the relationship between epoch levels and the Long Short Term Memory (LSTM) algorithm. This study will use an LSTM to predict future stock prices of Microsoft and Apple, where every test will have a different epoch value. The research results show that the LSTM heavily relies on epoch levels without any direct or inverse relationship. By adjusting the epoch levels, it significantly impacts the model's predictions, demonstrating the vitality of finding the most optimal epoch level without over-fitting or under-fitting the LSTM predictions. Research results show that LSTM models require precise evaluation with its epoch coefficients in order to optimize the machine learning predictions of forecasting stock price movements.

Keywords: Long Short-Term Memory (LSTM), Recurrent Neural networks (RNNs), Epoch levels, Time-series forecasting, Microsoft, Apple, Stock price prediction, Overfitting, and Underfitting.

1 Introduction

Recurrent Neural Networks (RNNs) are artificial neural networks that specialize in processing sequential data. By using internal memory, RNN can obtain temporal dependencies within input sequences. This makes RNNs an effective model for numerous tasks, specifically time series forecasting. Long Short-Term Memory (LSTM), one of the variations of the RNN architectures, has gained recognition in the past couple years due to their ability to overcome limitations RNN presented, specifically vanishing gradient problems. Furthermore, unlike RNN, LSTM models long-range dependencies for accurate predictions in the volatile financial market (Stryker 2024; Marghani 2025; Deressa 2025). LSTMs began to be widely adopted in stock price prediction due to their architecture, which includes memory cells gating mechanisms that facilitate retention and management of information over extended periods, allowing LSTMs to learn complex patterns and relationships in historical stock data. This enhances their predictive accuracy compared to other deep learning models, such as Convolutional Neural Networks (CNNs) and standard feedforward networks, of which both lacks the memory capability (Donges 2024; Limited 2025b).

In recent years, many studies have revealed the vital role LSTMs have in stock market forecasting. Their successful performance due to LSTMs unique architecture that is specialized to address the limitations of RNNs, specifically RNNs inability to learn long-term dependencies in sequential data, making LSTMs more effective for tasks such as time series forecasting. LSTMs architecture consists of an input, forget, and output gate that manages information flow, directly bypassing the vanishing gradient and inability to retain information in long sequences that basic RNNs encounter (Pilla and Mekonen 2025; Leccese 2019). Through its gates, LSTM is able to capture short-term fluctuations and long-term trends by selectively retaining or discarding information, in which this flexibility is essential for stock market predictions (Pilla and Mekonen 2025; Kumbure et al. 2022). For example, LSTMs can remember previous price patterns while integrating new data, granting them the ability to adapt with the fluctuating market (Yash Khodke 2025). In the context of predicting future stock prices for this research, the LSTM networks were employed to receive historical data in order to identify patterns within the input sequences to generate accurate predictions that reflect dynamics of the stock market by understanding LSTMs ability to maintain context across multiple steps.

1.1 Recurrent Neural Networks

Recurrent Neural Networks (RNN) is a deep learning model that retains information from previous time steps to inform future predictions. There are several kinds of RNNs, with each one addressing a challenge in modeling sequential data. The Simple RNNs is a basic structure with a hidden state that is updated every step based on the current input and the previous hidden state. Although it can capture short-term dependencies, they struggle with long-term dependencies due to vanishing gradient problems (Tamura 2020). Another example of an RNN is the Gated Recurrent Unit (GRU), where it possesses less gates, but features an updated mechanism (Limited 2025b). The most critical branch of RNN is Long-Short-Term Memory (LSTM), which holds three critical gates-the input, output, and forget gates-that enhances its ability to maintain and manage information over extended periods, excellent for sequence learning tasks (Donges 2024; Naufal and Wibowo 2023).

RNN models, specifically LSTM networks, contrast other deep learning models due to their specialized abilities of handling sequential data and time series forecasting. For example, if RNN and LSTM was compared with CNN, even though CNN would outperform RNN and LSTMs in its ability to process spatial tad and recognizing patterns in mage through pooling layers, CNN continues to lack the ability to retain memory of the previous inputs, making this model less suited for this research (Corporation 2025). As seen in Supplementary Fig. 7, it shows the structure of an RNN, in which the hidden layers continuously reanalyzes data and calculations, where the outputs of the hidden layers are stored in memory. This hidden state stores previous inputs, allowing them to effectively model sequential nature of data (Stryker 2024). Moreover, comparing RNN and LSTM with feedforward neutral networks only highlights LSTMs heightened ability to preproccess historical context.

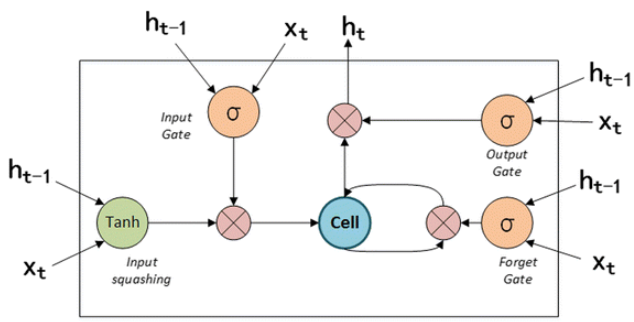

Figure 1: Structure of an LSTM unit (Li et al. 2017)

1.2 LSTM Model Architecture

LSTMs ability to counter vanishing and exploding gradient issues, specifically with the input, output, and forget gates, ensures a stable gradient flow throughout long temporal spans, which contributes to their surge in popularity modern studies (Noh 2021). Specifically, the LSTM network utilized in this analysis was designed to handle long-range dependencies in historical data, a common characteristic of stock price movements (Srivatsavaya 2023).

The LSTM model incorporates several layers: hyperparameter, learning rate, and batch size, which tuned to record LSTMs result accuracy. One of the main focuses of the model is the epoch—a complete pass through the entire dataset for model training in machine learning. With deep learning models like LSTM, the batch size is fine-tuned to improve model performance (Limited 2025a).

LSTM incorporates a memory cell and a forget gate, input gate, and output gate (shown in Fig. 1). This design allows LSTMs to maintain and manage information. The forget gate plays a vital role in determining which information from the previous cell state should be discarded. It takes two inputs—the current input x_t and the previous cell output h_{t-1}—and produces a value between 0 and 1. A value of 0 indicates that the information will be forgotten, while a value of 1 means it will be retained. The forget gate is defined as:

$$f_t = \sigma(\mathbf{W}_f \cdot [h_{t-1}, x_t] + \mathbf{b}_f) \tag{1}$$

where $\mathbf{W}_f$ represents the weight matrix associated with the forget gate, and $\mathbf{b}_f$ is the associated bias (Deressa 2025).

The input gate regulates the addition of new information to the cell state. Similar to the forget gate, it uses the sigmoid function to filter the values that should be retained and creates a candidate vector through the $\tanh(x)$ function. The input gate and candidate cell state are defined as:

$$i_t = \sigma\left(\mathbf{W}_i \cdot [h_{t-1}, x_t] + \mathbf{b}_i\right) \tag{2}$$

$$\hat{C}_t = \tanh(\mathbf{W}_c \cdot [h_{t-1}, x_t] + \mathbf{b}_c) \tag{3}$$

where i_t decides how much of the new information to add, and $\hat{C}_t$ represents the new candidate values (Naufal and Wibowo 2023).

The output gate determines the next hidden state by regulating the cell state information that will be passed on to the next cell. It first applies the $\tanh$ function to the current cell state, followed by a filtering process through the sigmoid function:

$$o_t = \sigma\left(\mathbf{W}_o \cdot [h_{t-1}, x_t] + \mathbf{b}_o\right) \tag{4}$$

The output of the output gate is then multiplied by the cell state to form the final output (Deressa 2025).

1.3 Applications

RNNs have found widespread application across various fields such as Natural Language Processing (NLP), speech recognition, and tie-series forecasting (Donges 2024; M 2025). Furthermore, RNNs ability to understand order and context are crucial for its ability in machine translation, allowing to analyze sentence structure and context; text generation, where chatbots can hold conversations and generate different text formats; music generation, generating music patterns by learning from preexisting pieces while generating new melodies; anomaly detection, where it learns normal patterns to detect anomalies; sequence study of the genome and DNA, where analyzing sequential data identifies and predicts gene functions (Kalita 2022).

With studies that require a handle on long-term dependencies within sequential data, LSTMs offer accurate results by intaking the sequential and time-dependent data; a few applications of LSTM include, but are not limited to, its high applicability in materials science, such as predicting Cohesive Zone Model (CZM) parameters recommender systems (Dai et al. 2024); learning behavior patterns to suggest personalized suggestions; time series forecasting, analyzing data to weather patterns (GeeksforGeeks 2019). While many uses of LSTM overlap those of RNNs models, such as text generation, anomaly detection, and speech recognition, studies have been conducted to compare the performance of RNNs and LSTM, revealing LSTMs higher validation and prediction accuracy than that of standard RNN (Xu 2023). Specifically, LSTM networks gained significant traction in applications, particularly in the financial sector with stock price predictions. Their networks are widely used to predict stock prices due to their models ability to retain complex temporal patterns in financial data. LSTM can analyze historical stock prices and various financial indicators (i.e. opening and closing prices, trading volume, and market capitalization) to forecast future prices (Zhixin Zhang 2025; GitHub 2023).

1.4 Experiment Study

The research studies aims to uncover the effects of hyperparameter tuning and data requirements of LSTM. Many studies in the last decade have applied LSTM to predict stock price predictions, but fail to acknowledge the risky relationship of over-training or under-training an AI model. Recent deep-learning model studies primarily focus on the results of LSTMs' predictive power but fail to recognize that an increase in epoch levels does not ensure elevated accuracy in the AI's result. By setting a fixed epoch or training number for the LSTM model, research studies typically overlook the faulty and unpredictable relationship between increasing epoch levels and the model's accuracy. One such example of this can be seen with Phuoc et al. 2024, where the study utilizes LSTM modeling to predict Vietnam's stock prices. However, the upon setting the epoch level to 1000, the study fails to mention tests taken in order to prevent possibly overfitting, neglecting the risks of biasing their data (Phuoc et al. 2024). Furthermore, there is a lack of understanding regarding what should the required epoch value be for a specific research topic, and how previous epoch values may contrast with those of another study's target of prediction. The aim of this study is to fine-tune the model epoch to hone the most accurate data result for stock predictions in order to uncover the relationship between epoch and AI accuracy. Hyperparameter tuning controls the learning rate of an AI model. This tuning process can be complex and time-consing, requiring experimentation to achieve optimal results. Inadequate tuning leads to suboptimal model performance, complicating the prediction process (Anishnama 2023; Technologies 2024).

The application of LSTM networks in predicting stock prices presents challenges and limitations that must me addressed and acknowledged.

Deep learning and AI training such as LSTM aims to develop models that can learn and make predictions on large amounts of data. In order to utilize large datasets, these AI models relies on training: the number of iterations in deep learning and AI training are crucial in the process and can greatly impact the performance of AI models. LSTM, relies significantly on the number of epoch and batch size to affect the accuracy and computational efficiency of the training process. An epoch is a single pass through the entire training dateset, used to measure the number of times the model has seen the entire data set in order to train and generate accurate data. Epoch is a vital hyperparameter to set correctly. If it is too small, the model may result in underfitting. If the number of epoch is too large, the model will overfit the training data, leading to poor performance. It is important to find the ideal number of epochs for a given training process through experimentation and monitoring the performance of a model's datasets (SabrePC 2025).

The significant challenges is the propensity of LSTMs to overfit and underfit. Overfitting occurs when a model learns to capture noise instead of underlying data patterns, leading to poor generalization. This issue is exacerbated in stock price prediction due to complex and nonlinear nature of the stock market. The deep learning model's precision also relies heavily on its epoch level, which allows it to learn trends and memorize specific data points (Anishnama 2023; Srivatsavaya 2023).

2 Research Methodology

Data Collection. The research applies altering the epoch levels of the LSTM algorithm to forecast future prices of the stock market. To accomplish the above research objective, the price history is collected from yahoofinance.com. The specific list of the stocks will be Apple and Microsoft. Each dataset was kept within recent months upon publication to prevent distorted shifts in the market environment. To ensure quality, the dataset was thoroughly analyzed for missing values, resulting in a clean dataset suitable for modeling (Lee 2024).

The python language was used to calculate the LSTM predicted prices affected by the altering epoch levels, and Google Sheets was used to collect the model data and evaluate the data. The LSTM model is built on the basis of Sklearn and Keras support libraries.

Research Methodology. This study analyzes the model of Long Short Term Memory and to find the most effective epoch number in order to predict future stock prices. The data is divided into two sets: the training and testing sets. The training set includes gathering historical data from the start date of January 1, 2025 to the last listing date before the predicted data with each dataset being thoroughly preprocessed. The preprocessing phase involved filtering and keeping the dataset within recent months to prevent huge misleading, inaccurate, or irrelevant changes in the business, environment, or technology (Josh 2022). Each dataset would then be filtered to focus on the 'Close' price and linear transformation of the original data through min-max normalization, enhancing the computational efficiency and model performance (Zhixin Zhang 2025; Zhenglin Li 2023). This transformation ensured that all data values were scaled between 0 and 1.

The testing set is a list of data with the model's predicted stock prices, and this will be used to evaluate the model's performance. Training sets and testing are independent, ensuring objectivity in the process of evaluating the performance of model.

The test set data from the model would be compared to true data from Yahoo Finance to calculate the total Mean Squared Error (MSE) for every complete test with a set epoch. The Mean Squared Error offers a numerical analysis of the model with an output of a series data, which reflects the overall model performance at a certain epoch (Vision 2025).

The model consists of the following coefficient: $batch\_size = 32$. That is, the number of samples used in one pass through the network has a direct impact on the accuracy and computational efficiency of the training process (SabrePC 2024).

The model consists of the variable $epochs = x$, where x will be changed by an interval of ten to analyze the number of times a model will loop without overfitting or early stopping the predicted prices.

Fig. 2 and Supplementary Fig. 8 are examples of the graphs and methods this study will use. Seen in Fig. 2, the data will be plotted as follows: the x-axis will by the epoch levels, increasing to analyze the model's predictive accuracy. The y-axis will graph the accuracy of the model's predictions, providing visual aid to analyze the Mean Squared Error results. The lower the MSE is, the more accurate the forecasted prices are. Therefore, the target of this study is to find the absolute minimum on the study's graph between epoch levels and MSE values.

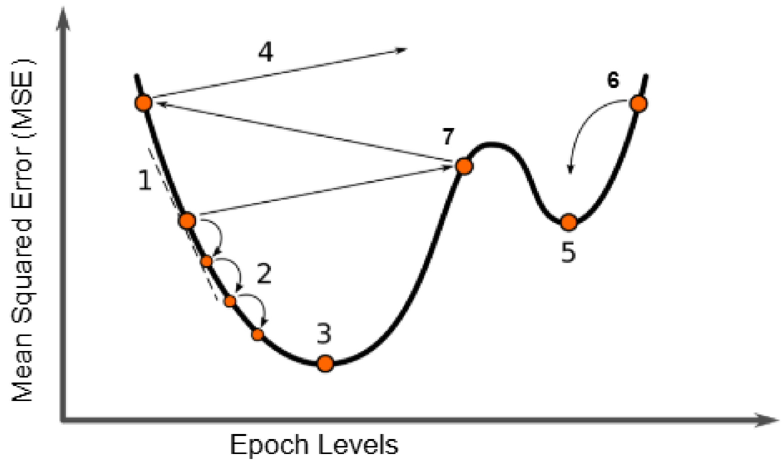

Figure 2: Example Graph (Rother 2023)

Seen in Supplementary Fig. 8, each increase in an epoch level is considered a learning step. As the study increases the learning step, the MSE may reach a minimum: the most accurate outputs from the model. However, the learning steps must refrain from overstepping as seen in Fig. 2: the progression from point one to seven to four highlights the risk that excessive epoch increases may cause the optimal point of predicted values to surpass the minimum error, thereby passing the most accurate prediction range. Another risk is understepping, where the step sizes are too small, misusing study time. Lastly, there is the risk of discovering the relative minimum instead of the absolute minimum, which can be observed in the example plot: point five in Fig. 2. The relative minimum may resemble the most optimal epoch, but in actuality, there is a more optimal epoch level for accurate results.

For this study, to avoid understepping or overstepping, the epoch level will start at ten, increasing every step size of ten until one-hundred epochs. Ten was chosen to avoid misusing study time while ensuring the model is not overstepping an excessive amount.

After building and performing the LSTM model with the training set, the LSTM model will forecast stock prices for the corresponding observations of the test set, that is, an estimated half a month after May. One complete test is when a whole half month has been recorded with a fixed epoch. Once a complete test has been recorded, the epoch level will increase by ten, and the program will be rerun. All the forecasted data will be recorded and compared with the true stock price, resulting in an evaluation. The same days will be predicted with the exception that epoch level will be increasing every full test. The stock prices forecasted from each complete test will be compared with the true stock price of each trading session. The accuracy of each complete test's prediction is evaluated based on the following formula,

$$A_j = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{5}$$

where A_j is the accuracy of the model's forecast for share j. The lower the calculated A_j is, the more accurate the model is; y_i is the true price of j at the ith session in the test; $\hat{y}_i$ is the forecasted result for the price of stock j at the with trading session in the test set; n is the number of sessions in the test set.

Once the data has been fully collected, the MSE from each corresponding epoch level will be set in a table and plotted, where epoch is the independent variable and the MSE is the dependent variable. From this qualitative and quantitative data, the datasets will be analyzed and discussed to understand the models performance.

3 Results and Discussion

The data provided shows the Mean Squared Error (MSE) values obtained from training a Long Short-Term Memory (LSTM) model over different numbers of iterations.

Epoch Level	MSE
10	73.3831933
15	130.8781667
20	69.8542267
30	53.9022733
40	68.4442600
50	47.0135133
60	59.6022600
80	53.9494867
70	52.8801067
90	54.5785400
100	59.9972333

Table 1: Apple stock: LSTM accuracy in relation with Epoch Levels

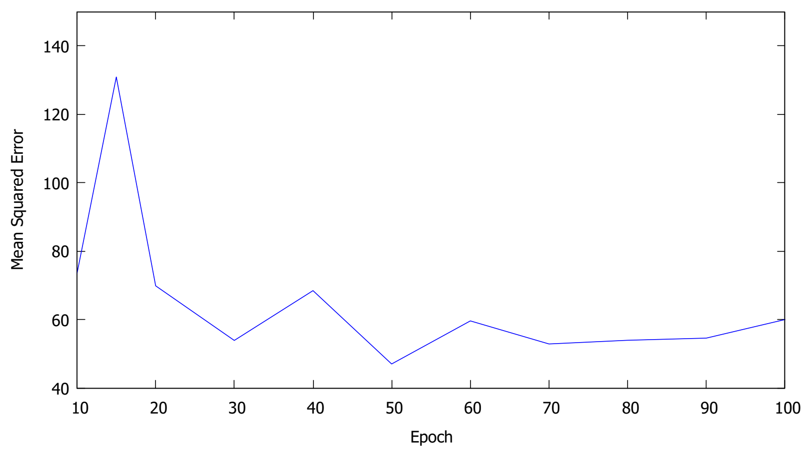

Figure 3: Apple stock: LSTM accuracy in relation with Epoch levels.

The data from Fig. 3 and from Table 1 demonstrates a variety of MSE values with varying performances from 47.01 to 130.88. This range suggests that the model's predictive accuracy fluctuates as the number of epoch levels increases or decreases, which is an expected behavior during LSTM training and testing due to the model's sensitivity to training dynamics and potential overfitting or underfitting.

Apple stock. Specifically, from Fig. 3 and table 1, the highest MSEs occur at the beginning iterations, indicating poor model performance at this stage, likely due to the insufficient training iterations for LSTM to adequately learn the stock's patterns. Conversely, the lowest MSEs recorded are around the higher ranges of epoch. However, the study reveals there is no clear relationship between the increase of epoch levels and LSTMs performance. This variability is consistent with LSTM behavior, where increasing iterations beyond the optimal point may lead to overfitting and diminishing returns, especially if the dataset is complex or noisy.

Microsoft stock. Analyzing table 2, it mirrors the previous analysis, the lower numbers of epochs resulted in a significant increase in its evaluation, showing poor performance. However, instead of 50 epoch displaying optimal results, 40 epoch reveals the best recorded results overall of all tests. Revealing that not only does epoch require a specific point before overfitting or underfitting, but it may differ from stock to stock. Furthermore, Fig. 2 reveals that with 40 epoch showing optimal result, 50 and above displays least optimal predictions, which is consistent with the behavior previously found where LSTMs performance has no direct relationship with an increase of epoch.

To further analyze this data, the MSE values align with a typical LSTM training outcome in tasks, where it heavily depends on factors of datasize, model architecture, and learning rate. The observed fluctuations suggest that the model's performance is highly sensitive to the number of epochs that may have designated benchmarks from stock to stock to prevent overfitting.

Epoch Level	MSE
10	283.6480467
20	192.3785067
30	279.0968333
40	17.1309667
50	156.2978267
60	65.1808400
70	85.5211333
80	84.1874533
90	76.0156600
100	71.4491933

Table 2: Microsoft stock: LSTM accuracy in relation with Epoch levels

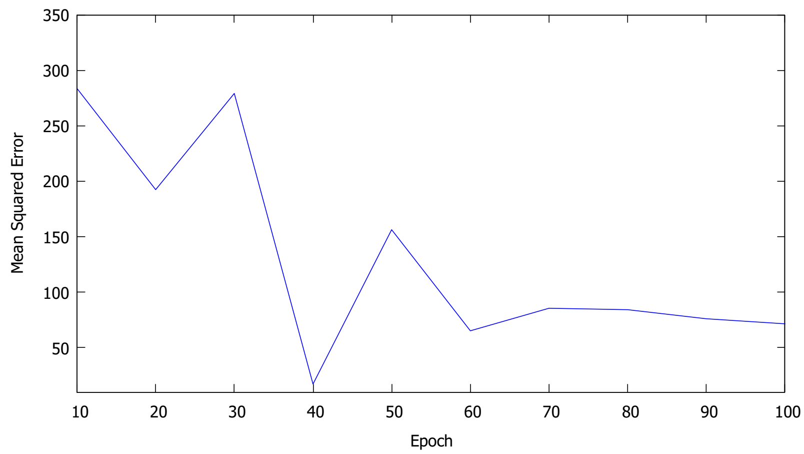

Figure 4: Microsoft stock: LSTM accuracy in relation with Epoch levels

The data provides a realistic depiction of LSTM training dynamics, highlighting the importance of selecting the appropriate number of epochs to minimize the error. The results support the need to carefully set hyperparameter optimization and validation techniques to achieve consistent and reliable models with the optimal epoch number for a robust model performance in practical applications.

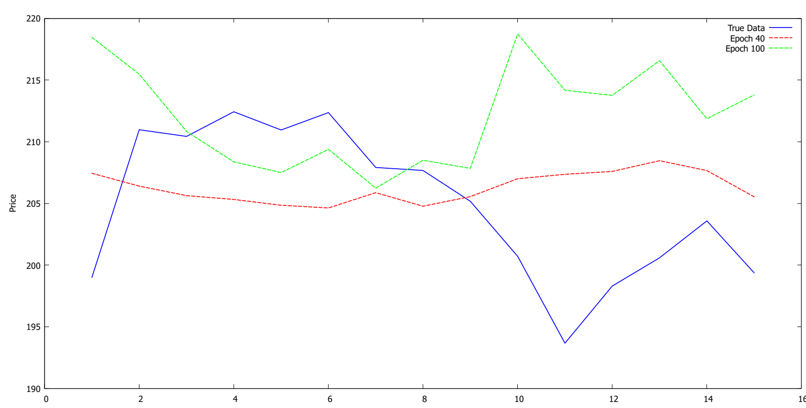

Figure 5: Compare forecast price and actual price Apple

Fig. 5 aligns with the typical learning curve of LSTM, where lower numbers of epoch is insufficient training iterations for LSTM to adequately learn from Apple's historical data and predict the stock's patterns.

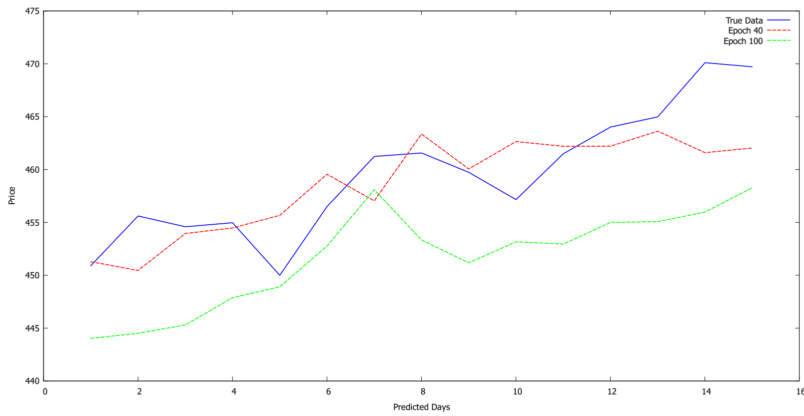

Figure 6: Compare forecast price and actual price Microsoft

Fig. 6 reveals a unique depiction of LSTMs training dynamics, where epoch 40 outperformed all epoch levels despite being one of the lowest training. Although most studies use higher epoch for more accurate predictions, Fig. 4 reveals that 40 epoch significantly outperforms 100 epoch despite having a lower learning rate, showing the lack of correlation between epoch and LSTMs predictive ability.

4 CONCLUSION

In this study with the aim of evaluating the application of LSTM algorithm and epoch level to forecast the stock market, using historical data of Yahoo Finance, reveals the vitality of altering epoch levels has on Long Short Term Memory. The forecast results show the different epoch requirements each stock may require and the predictive accuracy of using the LSTM algorithm. The study results show an altering predictive level for every stock studied and the strong reliance of LSTM's predictive power depending on the adjustments with epochs levels. With changes of epoch levels, it shows how the predictive power of LSTM, where each stock alters depending on the amount of training each model receives. Therefore, there is no set number of epochs that could maximize the predictive power for every stock. There is the risk of overfitting and underfitting with epoch, and the requirement of finding the precise number of epochs for perfect fitting.

It is a common trend in recent years to implement machine learning algorithms to economic (i.e. financial) problems in order to predict future pecuniary trends. There are many other complex machine learning algorithms that would require higher understanding but also higher performance. By combining these various machine learning techniques, it could enhance LSTM's performance as well as facilitate the finding of accurate epochs. This study only used one machine learning model; hence, an extension to this research could be improving the accurate number of epochs through other learning methods.

The gathered amount of historical data heavily affects the accuracy of the model's predictions. Another extension could be using other machine learning algorithms to filter the most accurate amount of historical data to prevent inaccurate or irrelevant information for the machine learning training. This also extends into the amount of days predicted with this algorithm. Only half a month was collected for this research purpose, which heavily affects the data calculated with the MSE. With an increase of days, there could be an increase or decrease in the MSE, altering the analyzed performance levels of the epochs.

Each of the following equations to find the relative minimum extends beyond 100 epochs, which has not been included in this study. The lowest MSEs found could be a relative minimum rather than an absolute minimum of the graphed prediction. An extension of this study includes analyzing the higher ranges of epoch levels, which may result in improved prediction performances from the model. Future studies could explore implementing other AI systems to enhance extensive prediction precision and data accuracy.

References

David Andrés. Gradient descent. 2022. URL `https://mlpills.dev/machine-learning/gradient-descent/`.

Anishnama. Understanding lstm: Architecture, pros and cons, and implementation. 2023. URL `https://medium.com/@anishnama20/understanding-lstm-architecture-pros-and-cons-and-implementation-3e0cc` a194094.

NVIDIA Corporation. Long short-term memory (lstm). 2025. URL https://developer.nvidia.com/discover/lstm.

Yanwei Dai, Jiahui Wei, and Fei Qin. Recurrent neural network (rnn) and long short-term memory neural network (lstm) based data-driven methods for identifying cohesive zone law parameters of nickel-modified carbon nanotube reinforced sintered nano-silver adhesives. *Materials Today Communications*, 39:108991, Jun 2024. doi: https://doi.org/10.1016/j.mtcomm.2024.108991. URL https://www.sciencedirect.com/science/article/pii/S2352492824009723.

Temesgen Deressa. Time series forecasting with rnn, gru, and lstm: Aapl stock price analysis. 2025. URL https://medium.com/@ttderessa/time-series-forecasting-with-rnn-gru-and-lstm-models-a-comparative-analysis-on-aapl-stock-price-cdcdbd3c5ad8.

Niklas Donges. A guide to recurrent neural networks (rnns). 2024. URL https://builtin.com/data-science/recurrent-neural-networks-and-lstm.

GeeksforGeeks. What is lstm long short term memory?, Jan 2019. URL https://www.geeksforgeeks.org/deep-learning/deep-learning-introduction-to-long-short-term-memory/.

GitHub. Stock price prediction with lstm. 2023. URL https://github.com/ftmoztl/stock-price-prediction-with-LSTM.

Josh. Limitations of historical data. 2022. URL https://support.onesaas.com/hc/en-us/articles/204756914-Limitations-of-historical-data#:~:text=Back%20To%20Top-,General%20Errors%20with%20Historical%20Data,deleted%20from%20your%20eCommerce%20system.

Debasish Kalita. A brief overview of recurrent neural networks (rnn), Mar 2022. URL https://www.analyticsvidhya.com/blog/2022/03/a-brief-overview-of-recurrent-neural-networks-rnn/.

Mahinda Mailagaha Kumbure, Christoph Lohrmann, Pasi Luukka, and Jari Porras. Machine learning techniques and data for stock market forecasting: A literature review. *Expert Systems with Applications*, 197:116659, 2022. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2022.116659. URL https://www.sciencedirect.com/science/article/pii/S0957417422001452.

Andrea Leccese. Machine learning in finance: Why you should not use lstm's to predict the stock market. January 2019. URL https://www.blueskycapitalmanagement.com/machine-learning-in-finance-why-you-should-not-use-lstms-to-predict-the-stock-market/.

Ernesto Lee. Advanced stock pattern prediction using lstm with the attention mechanism in tensorflow: A step by step guide with apple inc. (aapl) data. 2024. URL https://drlee.io/advanced-stock-pattern-prediction-using-lstm-with-the-attention-mechanism-in-tensorflow-a-step-by-143a2e8b0e95.

Shumin Li, Junjie Chen, and Bin Liu. Protein remote homology detection based on bidirectional long short-term memory. *BMC Bioinformatics*, 18, 10 2017. doi: 10.1186/s12859-017-1842-2.

Sanchhaya Education Private Limited. Epoch in machine learning. April 2025a. URL https://www.geeksforgeeks.org/epoch-in-machine-learning/.

Sanchhaya Education Private Limited. What is lstm - long short term memory? May 2025b. URL https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/.

Relinns Technologies Pvt. Ltd. 2018. URL https://botpenguin.com/glossary/recurrent-neural-network.

Siddharth M. Stock price prediction using lstm and its implementation. 2025. URL https://www.analyticsvidhya.com/blog/2021/12/stock-price-prediction-using-lstm/.

Muskan Marghani. Recurrent neural networks (rnns) explained: Architecture, applications, and limitations. 2025. URL https://medium.com/@muskanmarghani13/recurrent-neural-networks-rnns-explained-architecture-applications-and-limitations-2a2c8a40ca57.

Ghani Rizky Naufal and Antoni Wibowo. Time series forecasting based on deep learning cnnlstm-gru model on stock prices. *International Journal of Engineering Trends and TEchnology*, 71:126–133, 2023. ISSN 2231–5381. URL https://ijettjournal.org/Volume-71/Issue-6/IJETT-V71I6P215.pdf.

Seol-Hyun Noh. Analysis of gradient vanishing of rnns and performance comparison. *Information*, 12(11):442, Oct 2021. doi: https://doi.org/10.3390/info12110442.

T. Phuoc, P. T. K. Anh, P. H. Tam, and C. V. Nguyen. Applying machine learning algorithms to predict the stock price trend in the stock market – the case of vietnam. *Humanities and Social Sciences Communications*, 11:393, 2024. doi: 10.1057/s41599-024-02807-x. URL https://doi.org/10.1057/s41599-024-02807-x.

Prashant Pilla and Raji Mekonen. Forecasting s&p 500 using lstm models. January 2025. URL https://arxiv.org/html/2501.17366v1.

Kristian Rother. Gradient descent. 2023. URL https://www.academis.eu/machine_learning/supervised/linear_regression/gradient_descent.html.

SabrePC. Epochs, batch size, iterations - how are they important to training ai and deep learning models. 2024. URL https://www.sabrepc.com/blog/Deep-Learning-and-AI/Epochs-Batch-Size-Iterations.

SabrePC. Epochs, batch size, iterations - how are they important to training ai and deep learning models. 2025. URL https://www.sabrepc.com/blog/Deep-Learning-and-AI/Epochs-Batch-Size-Iterations.

Prudhviraju Srivatsavaya. Lstm — implementation, advantages and diadvantages. 2023. URL https://medium.com/@prudhviraju.srivatsavaya/lstm-implementation-advantages-and-diadvantages-914a96fa0acb.

Cole Stryker. What is a recurrent neural network (rnn)? 2024. URL https://www.ibm.com/think/topics/recurrent-neural-networks.

Yasuto Tamura. Simple rnn: the first foothold for understanding lstm. 2020. URL https://interviewdb.com/time-series-analysis/1282/.

Q3 Technologies. How lstm networks are revolutionizing time series forecasting. 2024. URL https://www.q3tech.com/blogs/lstm-time-series-forecasting/.

Encord Computer Vision. Mean square error (mse). 2025.

Zhaozhen Xu. Baeldung, Aug 2023. URL https://www.baeldung.com/cs/lstm-vanishing-gradient-prevention.

Sonali Deshpande Yash Khodke. Stock price prediction using lstm and gru. 2025. URL https://www.ijraset.com/research-paper/stock-price-prediction-using-lstm-and-gru.

Jinxin Xu Jihang Liu Yuhong Mo Zhenglin Li, Hanyi Yu. Stock market analysis and prediction using lstm: Acase study on technology stocks. 2023.

Yanrong Hu Hongjiu Liu Zhixin Zhang, Qingyang Liu. Multi-feature stock price prediction by lstm networks based on vmd and tmfg. *Journal of Big Data*, 2025. URL https://journalofbigdata.springeropen.com/articles/10.1186/s40537-025-01127-4] [https://github.com/ftmoztl/stock-price-prediction-with-LSTM.

Supplementary Material

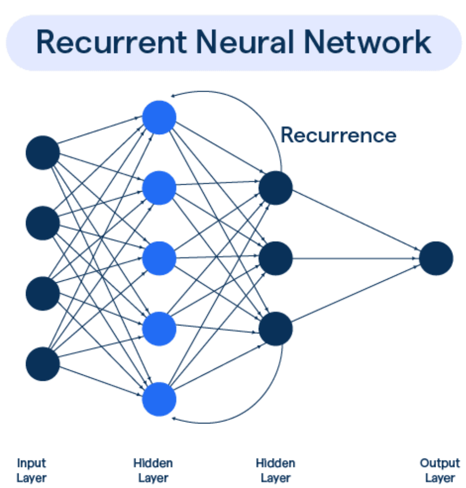

Figure 7: Structure of a Recurrent Neural Network (RNN) (Ltd. 2018).

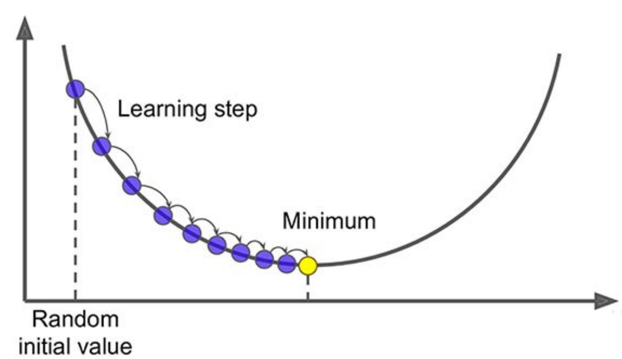

Figure 8: Gradient Descent (Andrés 2022).