Comparing Machine Learning Models for Stock Market Price Prediction*

Pranesh Ambokar
Virginia Tech
Blacksburg, VA, USA
pambokar@vt.edu

Mohammad Heydari
Virginia Tech
Blacksburg, VA, USA
heydari@vt.edu

Roshan Ravindran
Virginia Tech
Blacksburg, VA, USA
roshan14@vt.edu

Leo St. Amour
Virginia Tech
Blacksburg, VA, USA
lstamour@vt.edu

Abstract

The stock market impacts many facets of modern society, including education, technology, and retirement. To maximize the value the stock market can provide, stock market prediction has become an exciting and relevant problem for financial analysts and researchers. Research has explored various stock market analysis methods, resulting in a taxonomy of prediction techniques ranging from statistical analysis to pattern recognition. Machine learning is a promising solution for reasoning about stock market trends and informing financial decisions. This paper compares linear regression and Long Short-Term Memory for making stock price predictions. Using a large data set of historical stock prices between 2018 and 2024, we use each algorithm to analyze its accuracy and performance characteristics. These properties allow us to adequately compare and contrast the strengths and weaknesses of each algorithm and draw conclusions about their applicability in stock market prediction. We present the results of a day-trading simulation to compare the real-world applicability of each model. Our evaluation indicates both models are capable of making accurate closing price predictions. However, our evaluation demonstrates that linear regression has a surprising edge in accuracy and real-world day-trading applicability.

CCS Concepts

• **Computing methodologies** → **Model development and analysis**; *Learning linear models*; *Neural networks*; Supervised learning.

Keywords

Stock market forecasting, linear regression, LSTM, model analysis

ACM Reference Format:
Pranesh Ambokar, Mohammad Heydari, Roshan Ravindran, and Leo St. Amour. 2024. Comparing Machine Learning Models for Stock Market Price Prediction. In *CS 5805: Machine Learning, Fall 2024.* ACM, New York, NY, USA, 8 pages.

*Authors listed alphabetically

CS 5805: Machine Learning, Fall 2024, Virginia Tech

1 Introduction

The stock market impacts many facets of modern society, including education, technology, and retirement [6]. To maximize the value the stock market can provide, stock market prediction has become an exciting and relevant problem for financial analysts and researchers. Research has explored various stock market analysis methods, resulting in a taxonomy of prediction techniques ranging from statistical analysis to pattern recognition [17]. Machine learning has emerged as a promising solution for reasoning about stock market trends and informing financial decisions [6, 10, 12, 18].

Broadly, machine learning aims to solve two stock market prediction problems: (1) price forecasting and (2) direction classification. For the former, a model predicts the future price of a stock based on its historical performance. For the latter, a model classifies a stock as "buy," "sell," or "hold," based on performance indicators and buy/sell signals. Stock prices are inherently volatile and fluctuate due to complex and dynamic market forces, making accurate price prediction and reliable classification challenging. Success in either area could enable investors to make more informed decisions, potentially buying stocks at optimal times and maximizing returns by selling opportunistically.

This project focuses on the first prediction problem, approaching stock analysis as a price prediction task. We compare analyze and compare two machine learning algorithms previously applied to stock market price forecasting. Specifically, we compare and contrast linear regression and Long Short-Term Memory (LSTM). We use an open-source implementation of each provided by `scikit-learn` and `tensorflow`, respectively. We train and test each algorithm to analyze its accuracy and performance characteristics using a large data set of historical stock prices between 2018 and 2024 acquired from Kaggle [11] and the Yahoo Finance API. These properties allow us to adequately compare and contrast the strengths and weaknesses of each algorithm and draw conclusions about their applicability in stock market prediction.

In our evaluation, we analyze each model's ability to predict a closing price for an individual day, given its opening price and the high and low value the price achieves throughout the day. Our empirical results suggest that both linear regression and LSTM produce high levels of accuracy, as demonstrated by the mean squared error (MSE) and R^2 achieved by our models. However, in terms of performance, LSTM requires significantly greater computational time and resources than linear regression. Additionally,

we conducted a day-trading simulation to assess and compare the real-world implications of each model. Our simulation suggests that while both models perform better than random, linear regression has an edge over LSTM.

The rest of this paper is organized as follows: Section 2 provides background information on linear regression, LSTM, and their previous applications in stock market prediction; Section 3 describes the methodology behind our evaluation; Section 4 discusses the results and implications of our analyses; and Section 5 provides concluding remarks.

2 Literature Review

A financial market's inherent volatility and non-linear nature have long made stock market forecasting difficult. However, traditional statistical methods have given way to complex machine-learning techniques in research, yielding ever more advanced tools for stock price predictions. This paper compares two machine learning algorithms for stock market prediction. Specifically, we selected linear regression and LSTM as our evaluation subjects. Linear regression was chosen for its simplicity, interpretability, and usefulness as a baseline for comparing more sophisticated models. Its outcomes provide a benchmark for assessing how well sophisticated methods work. But its drawbacks, such as its inability to handle volatile and non-linear data well, highlight the need for more advanced methods [8]. LSTM was chosen because of its proven capacity to capture long-term dependencies and model sequential data, which is essential for stock market prediction. Research shows that LSTM performs better than other machine learning models and conventional statistical techniques, especially when dealing with noisy and non-linear financial time series [1, 6, 15]. This study intends to contribute to the continuous investigation of efficient techniques for stock market forecasting by contrasting these two approaches and assessing their trade-offs in accuracy, computational complexity, and practical usability. We next describe algorithmic details of linear regression and LSTM and provide additional details on their applicability to stock market prediction in prior literature.

2.1 Linear Regression

Linear regression is a fundamental supervised modeling technique that attempts to establish a linear relationship between dependent and independent variables. In its simplest form, linear regression models the relationship between a scalar response and one or more explanatory variables by fitting a linear equation to the observed data [4]. The model assumes that the relationship between variables can be described by a straight line, represented as:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n + \epsilon \tag{1}$$

where y is the dependent variable, x_1 through x_n are independent variables, β_0 is the y-intercept, β_1 through β_n are the coefficients that describe the relationship between each independent variable and the dependent variable, and ϵ represents the error term [8].

The model determines the optimal values for these coefficients by minimizing the cost function (sum of squared residuals between predicted and actual values). This optimization process makes linear regression particularly appealing for its computational efficiency and mathematical tractability. Additionally, the model's interpretability allows analysts to understand the relative importance of different features through the magnitude and sign of their corresponding coefficients [7].

Linear regression is computationally efficient for high-dimensional data, requiring only a single matrix operation to compute the optimal parameters. This efficiency and determinism ensure consistent results and enable rapid model validation across different market scenarios. Furthermore, the model's analytical properties allow for straightforward calculation of confidence intervals and prediction bounds, providing valuable insights into prediction uncertainty when modeling stock prices. In practice, these bounds can help quantify the reliability of price forecasts, making linear regression particularly useful as both a baseline model and an analytical tool [17].

Because of its ease of use and interpretability, linear regression has long been a fundamental statistical technique for simulating connections between variables. Using factors like opening and closing prices and concentrating on short-term trends, studies like those conducted by Emioma and Edeki (2021) used least-squares linear regression to forecast stock prices. However, the dynamic and non-linear structure of stock market data degrades the efficiency of linear regression, even though it captures simple relationships well [4, 8].

2.2 Long Short-Term Memory

LSTM networks are a specialized type of recurrent neural network (RNN) designed to effectively capture long-term dependencies in sequential data [5]. Unlike traditional RNNs, which often suffer from the vanishing gradient problem, LSTMs incorporate gating mechanisms that regulate the flow of information and maintain the integrity of the memory cell over extended time periods [19]. The gates enable LSTMs to retain relevant information and discard irrelevant data, which makes them particularly suitable for handling complex temporal patterns.

To address the intricacies of financial forecasting, increasingly sophisticated methods such as artificial neural networks (ANNs) and RNNs were created with the advent of machine learning [6, 16]. A subclass of RNN called LSTM networks was developed to overcome drawbacks, including the vanishing gradient issue that plagues conventional RNNs. LSTMs are useful for modeling time-series data, such as stock prices, because they use memory cells to store information over extended periods. According to research by Bhandari et al., LSTMs perform noticeably better than conventional techniques like autoregressive integrated moving average (ARIMA) and simple RNNs, successfully capturing the noisy and non-linear nature of stock market data [1]. Similarly, studies by Saboor et al. (2020) and others applied LSTM models to indices like the S&P 500 and NIFTY 50, achieving higher accuracy in predicting stock prices and validating the model's applicability in financial forecasting [1, 15].

In stock market prediction, LSTMs are employed to analyze time-series data by utilizing the information from previous time steps [1]. The architecture of LSTMs allows them to capture intricate trends in historical stock prices, which can be instrumental in forecasting future price movements. Moreover, LSTMs are highly flexible and can integrate various input features, including technical indicators,

trading volumes, and macroeconomic variables, enhancing their predictive capabilities [15]. Advanced architectures, such as stacked LSTMs and bidirectional LSTMs, have been explored to improve model performance by capturing hierarchical representations, allowing LSTMs to better adapt to the noisy financial time series data [5].

Recent studies have further explored the application of LSTM networks in stock market prediction in different market conditions and datasets. For example, Borovkova and Tsiamas [2] proposed an ensemble of LSTM neural networks for high-frequency stock market classification. Their approach uses a diverse set of technical analysis indicators as inputs and employs an online weighting mechanism that adjusts the influence of individual models based on their recent performance. The results showed that their model outperforms benchmark models, such as lasso and ridge logistic classifiers, in terms of predictive performance. In another study, Pawar et al. (2018) [14] implemented LSTM-based RNNs for stock market price prediction focusing on portfolio management. They compared the performance of their LSTM models against traditional machine learning algorithms, including regression, support vector machines (SVMs), Random Forests, Feed Forward Neural Networks, and Backpropagation. The results indicated that the LSTM models achieved superior accuracy, highlighting their ability to capture complex temporal dependencies and nonlinear relationships in historical stock data. Additionally, the study discussed the impact of customer sentiment on stock trends, suggesting that integrating sentiment analysis with LSTM models could further enhance prediction accuracy. Chen et al. (2015) [3] conducted a case study on the Chinese stock market, utilizing LSTM networks to predict stock returns. Their methodology involved transforming historical stock data into 30-day sequences with multiple learning features and labeling based on a 3-day earning rate. The findings revealed a significant improvement in prediction accuracy from 14.3% with random prediction methods to 27.2% using the LSTM model.

The computational complexity of LSTMs is generally higher than that of other methods due to their deep architecture and the need for extensive training. However, the trade-off is often analyzed and justified by the significant improvements in predictive performance and the ability to model complex dependencies within the data [19]. As a result, LSTMs have become a cornerstone in financial forecasting by offering a powerful means to derive actionable insights from historical stock data. Their superior performance in capturing nonlinear patterns and integrating diverse input features underscores their applicability and effectiveness in financial forecasting tasks.

2.3 Applications in Stock Market Prediction

Many studies have investigated the application of machine learning techniques, including linear regression and LSTM, for stock market prediction. In an extensive literature review, Kumbure et al. (2022) examined the data, markets, and machine learning methods used in stock market forecasting research from 2000 to 2019. They found that the most widely used prediction techniques in recent years are the neural networks, particularly LSTMs and SVMs. The US stock market and Asian markets such as Taiwan and China were the most frequently studied. Technical indicators were the dominant input features, while some studies also incorporated macroeconomic variables, fundamental financial data, and market sentiment extracted from news and social media. The review highlighted the growing sophistication and diversity of machine learning approaches to stock prediction [10].

Patidar et al. (2023) compared the performance of linear regression and LSTM models in predicting stock prices of specific companies. Their user-centric approach aimed to assist newcomers in making informed decisions about selecting, retaining, or selling stocks. The study utilized historical price data sourced from Yahoo Finance and achieved mean absolute errors (MAE) of 1.25 for linear regression and 0.82 for LSTM, demonstrating the superior accuracy of LSTM in capturing complex temporal dependencies. The authors emphasized the potential of machine learning techniques for reliable market prediction, even using only historical data [13].

Orsel and Yamada (2022) conducted a comparative study of linear Kalman filters and various LSTM architectures in predicting daily prices of individual stocks (Microsoft and Tesla) and market indices (S&P 500 and Russell 2000) over ten years from 2011 to 2021. Their results showed that a linear Kalman filter could predict next-day prices reasonably well for low-volatility stocks like Microsoft. At the same time, LSTM models significantly outperformed the Kalman filter for high-volatility stocks like Tesla. Among the LSTM variants tested, bidirectional and convolutional LSTM (CNN-LSTM) models achieved the best performance. This highlights the importance of capturing past and future dependencies and extracting relevant features from the input sequence. Interestingly, the study found that LSTM models trained on certain stock types could generalize to similar stocks without requiring retraining, suggesting the potential for efficient transfer learning in stock prediction [12].

Shen et al. (2012) applied SVMs to predict the movement direction of major US stock indices, including the S&P 500, Dow Jones Industrial Average, and NASDAQ Composite. By selecting input features through correlation analysis, they discovered that the prior day's index values from certain foreign markets, such as Germany's DAX, had strong predictive power for the US market. The SVM models achieved accuracy levels of 74 to 78% in predicting the daily direction of the indices. The study demonstrated the value of incorporating global market information and using feature selection techniques to improve the performance of machine learning models for stock prediction [18].

Jha et al. (2024) compared the performance of linear regression, LSTM, and hybrid models in predicting the stock prices of Google and Tesla. This study found that LSTM models consistently outperformed linear regression, achieving lower mean squared error and better capturing the non-linear dynamics of the stock prices. However, a hybrid model combining an ARIMA model, a generalized autoregressive conditional heteroskedasticity (GARCH) model, and a feed-forward neural network achieved the best overall performance. This study highlighted the potential of integrating classical time series models with deep learning architectures to leverage their complementary strengths for stock prediction [9].

Kumbure et al. (2022) proposed an improved stacked LSTM architecture for predicting intra-day price movements of the CSI 300, a major Chinese stock index. They first decomposed the raw price data using wavelet transforms to extract multi-scale features and then applied a genetic algorithm-based fuzzy model to select

the most relevant features for prediction. The selected features were fed into a two-layer LSTM network for training and forecasting. This approach reduced the root mean squared error (RMSE) by over 50% compared to standalone LSTM, convolutional neural network (CNN), and other machine learning models. The study demonstrated the importance of effective feature engineering and architecture design in enhancing the performance of deep learning models for stock prediction, especially in high-frequency trading [10].

The literature demonstrates that while classical methods like linear regression remain viable for certain stock prediction tasks, LSTM neural networks and their extensions have emerged as state-of-the-art techniques by their ability to capture complex non-linear patterns and long-term dependencies in market data. Hybrid approaches integrating machine learning with traditional time-series analysis have also shown promise in improving prediction accuracy. Careful feature engineering and architecture optimization techniques, such as stacked and bidirectional LSTMs, are key factors in maximizing the potential of AI in stock market prediction. As the field continues to evolve, more sophisticated and effective prediction models will likely be developed using deep learning and domain-specific knowledge advances.

3 Methodology

In this section, we describe our evaluation methodology. We discuss the processes for preprocessing the data, validating and training the models, and testing them. We also discuss the evaluation metrics we collect to compare the models.

3.1 Data Collection and Preprocessing

Large-cap and mid-cap stocks were the primary market categories on which we concentrated our investigation to preserve computational viability and guarantee thorough market representation. In particular, we gathered historical data for the top 50 businesses by market capitalization, which included 50 mid-cap firms like Williams-Sonoma and Illumina as well as industry titans like NVIDIA, Apple, Microsoft, and Google. The selective method is used for analytical reasons. Most market capitalization comprises large-cap stocks, which also tend to exhibit more consistent trading trends. On the other hand, mid-cap equities frequently exhibit distinct trading traits and greater volatility. Additionally, both groups include reputable historical data from well-established businesses, reducing the possibility of data quality problems with smaller or more recent market participants.

We collected daily trade metrics for each stock, such as opening price, high price, low price, closing price, adjusted closing price, and trading volume, using the Yahoo Finance API and the yfinance Python package. Every stock's history data was stored in a separate CSV file. The data collection covers several market cycles and significant economic events, starting with stock enlistment data and ending in November 2024. The quality and integrity of the data were examined (e.g., standardizing date formats and column names, confirming the completeness of the data for each stock, etc.).

Before using the dataset for modeling, several preparation procedures were carried out to guarantee data integrity and prepare it for analysis. The dataset was first divided into three subgroups for testing, validation, and training. Seventy percent of the data was in the training set, fifteen percent was in the validation set, and the remaining fifteen percent was in the testing set. Our partitioning technique preserved the data's chronological sequence to avoid leaking information between the subsets. The integrity of the evaluation process was maintained, representing genuine prediction scenarios, by ensuring that the training and testing sets did not overlap temporally.

Second, we chose our models' input and output characteristics. The closing price was the target feature; the input characteristics were the opening price, daily high, and daily low. The study's goal of assessing short-term market performance aligned with this feature selection, which emphasized daily stock price prediction. The relevance of the data utilized for modeling was ensured by selecting input features based on their direct influence on stock movement for a single day.

Lastly, we used min-max scaling to normalize all numerical features. Scaling the feature values to a range of 0 to 1 standardized the dataset and increased training efficiency. By ensuring that each feature contributed equally to the model, normalization prevented scale disparities from causing any one variable to dominate.

3.2 Models, Validation, and Training

We evaluate two machine learning models: linear regression and Long Short-Term Memory (LSTM) networks. We implement each model using industry-standard Python libraries, ensuring reproducibility, optimization, and reliability. The linear regression model was implemented using the `LinearRegression` class from the `scikit-learn` library, a widely used tool for statistical modeling and machine learning.

The LSTM network was implemented using `TensorFlow` and `Keras`, two widely-used libraries for deep learning applications. The architecture of the LSTM model used in this study consisted of three primary components: an input layer, hidden layers, and an output layer. The input layer processed the scaled stock price features, including the opening price, daily high, and daily low. These scaled features ensured that the model could learn efficiently without being biased by differences in the magnitude of the input data. The hidden layers included a single LSTM layer with 100 units, which enabled the model to retain long-term dependencies in the data. To mitigate overfitting, the architecture incorporated a recurrent dropout rate of 0.1 within the LSTM layer and an additional dropout layer with a rate of 0.3. These regularization techniques helped improve the generalizability of the model. The output layer predicted the closing price for a given day based on the processed input features and the learned temporal dependencies.

We used the validation set to fine-tune the LSTM hyper-parameters and optimize the model's performance. Key hyper-parameters included the number of units in the LSTM layer, dropout, and learning rates. The learning rate was set to 0.005, which was found to balance the trade-off between convergence speed and stability. This careful fine-tuning process ensured that the LSTM model could effectively capture the complex, non-linear relationships in the stock price data while maintaining robust performance across different validation scenarios. In contrast, the linear regression model had no hyper-parameters to fine-tune. After fine-tuning the LSTM

hyper-parameters, we trained each model using the training data set. During this evaluation phase, we collected the time required to train and fine-tune each model.

3.3 Testing and Evaluation Metrics

The testing and evaluation phase was critical in assessing both models' performance and practical utility. Both models were tested on an unseen testing set to ensure that the evaluation reflected their ability to generalize to new data. During the testing phase, we provided the testing set of each *individual* stock to both models. After making predictions, we collected several performance metrics to quantify the accuracy and performance of the models.

The first metric used was the mean squared error, which measures the average squared difference between the predicted and actual closing prices. By squaring the errors, this metric penalizes larger discrepancies more heavily, making it effective for evaluating how well the models performed in minimizing significant prediction errors. A lower MSE indicates a model's ability to make predictions that closely match the actual values. The second metric, mean absolute error, captures the average magnitude of the prediction errors without considering their direction (positive or negative). mean absolute error (MAE) provides an intuitive understanding of the model's accuracy, as it directly reflects the average deviation of the predictions from the actual closing prices. This metric is particularly useful in scenarios where understanding the overall magnitude of error is more important than penalizing larger errors. The third metric, R^2 (R-squared), indicates the proportion of variance in the actual closing prices that the model could explain. A higher R^2 value signifies that the model captures the underlying trends in the data effectively, making it an essential measure of goodness-of-fit. When testing each stock, we collected the time it took for each model to make predictions. Table 1 summarizes the metrics for each model. We report the average MSE, MAE, and R^2 values across all stocks.

In addition to these quantitative metrics, we conducted a day-training simulation to evaluate and compare the models' real-world applicability. We randomly selected stocks from the testing set and used the models to predict their closing price for a single day. Based on the predicted prices, we identified which stock would produce the highest yield (i.e., the largest difference between the closing and opening price). We then compared the prediction to the stock that produced the greatest yield. We calculated prediction accuracy based on 100 trials. We repeated this process, incrementing the number of stocks chosen by one from one to 30. To evaluate and compare each model's applicability, we graphed the average model accuracy achieved when making predictions from various numbers of stocks. We augmented these graphs with a line depicting the accuracy achieved by a random guess. Figures 3 and 4 present the results of the day-trading simulations. The ability of each model to correctly identify the highest-yielding stock is a measure of its practical utility.

The evaluation process combined these performance metrics with the day-trading simulation to capture the models' predictive accuracy and practical utility. This comprehensive approach ensured that each model's strengths and limitations were thoroughly assessed, offering a nuanced understanding of their applicability in stock market forecasting.

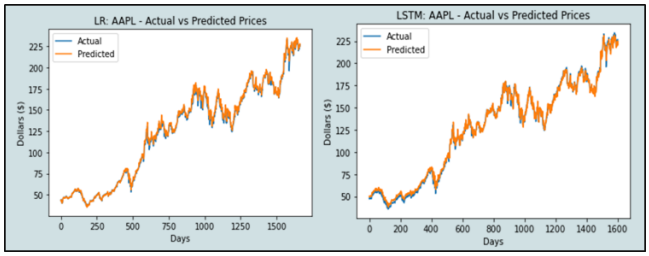

Figure 1: Apple stock prediction results

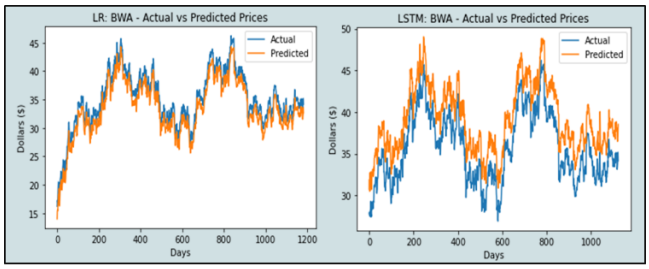

Figure 2: BorgWarner stock prediction results

4 Analysis

Next, we discuss the results of our evaluation and describe how the models' accuracy, performance, and practical implications compare. Comparing results of linear regression and LSTM models for stock price prediction reveals interesting patterns across different market segments. The evaluation focused on two categories: high market cap stocks, exemplified by AAPL (Apple Inc.), and mid-market cap stocks, represented by BWA (BorgWarner Inc.). Figures 1 and 2 depict the accuracy of our models when predicting closing prices for AAPL and BWA, respectively. Accuracy graphs for all 100 stocks can be viewed on our project website[1].

Both models demonstrated strong predictive capabilities for high market cap stocks, with clear and stable trends emerging in their predictions. The linear regression model achieved particularly impressive results, showing an average R^2 value of 0.977, indicating that the model could explain 97.7% of the variance in stock price movements. While still performing well, the LSTM model achieved a slightly lower average R^2 value of 0.935. Considering how the problem was framed, both models provided high accuracies. Both models showed a remarkable ability to make accurate future predictions, particularly during periods of steady market movement.

Our evaluation demonstrated that, in many cases, our models yielded better accuracy on high-cap stocks than mid-cap stocks. This trend is exemplified by comparing the predictions for AAPL with BWA predictions. While we only discuss AAPL and BWA, other high and mid-cap stocks demonstrate similar trends. Both graphs in Figure 1 show minimal differences between the predicted and actual closing values. In contrast, the graphs in Figure 2 exhibit

[1] https://1stl0ve.github.io/CS5805-project/project.html#accuracy-graphs

Table 1: Performance Metrics Comparison between Linear Regression and LSTM

Metric (Average)	Linear Regression	LSTM
R squared	0.977	0.935
MSE	1.45e-8	3.89e-08
MAE	0.000076	0.000110
Training time	Less than 1 second	1.17 hours

a more significant gap. High market cap stocks often follow a similar steadily increasing trend. We observed more complex patterns and increased prediction volatility when examining the models' performance on mid-market cap stocks. We hypothesize that this increasing trend makes their prediction easier and can be one of the reasons why the models are performing slightly better for high-cap stocks.

Further, our evaluation suggests that the LSTM model has lower accuracy on mid-cap stocks than linear regression. As demonstrated by Figure 2, while both models produce predictions following the curve of the actual closing prices, the LSTM model over-predicted the prices, especially for stocks experiencing sudden price changes. This trend suggests a potential overemphasis on historical patterns.

Surprisingly, the linear regression model showed competitive performance with the LSTM model despite its simplicity, particularly in capturing day-to-day price movements. The MSE for linear regression was 1.45e-8, compared to LSTM's 3.89e-8, suggesting better accuracy in short-term predictions. The MAE results gave more insights into both models' high prediction accuracy. The linear regression model achieved an MAE of 0.000076, while the LSTM model showed a slightly higher MAE of 0.000110.

As expected, the computational efficiency between the two models starkly diverged. Linear regression completed its training in less than one second, while the LSTM model required approximately 1.17 hours for training completion. The substantial difference in computational requirements raises important considerations for practical day-trading applications, especially in scenarios requiring frequent model updates or real-time predictions. Further analysis of the prediction accuracy across different market conditions revealed interesting patterns. Both models maintained relatively stable performance during periods of high market volatility, though with slightly increased error rates. The linear regression model showed particular resilience in maintaining consistent prediction accuracy across different market conditions, while the LSTM model demonstrated marginal improvements during periods of market stability.

These results suggest that the simpler linear regression model might be more practical for real-world applications, particularly in scenarios requiring rapid decision-making. A preliminary analysis of the prediction error distributions revealed that both models performed better during market uptrends than downtrends. The linear regression model showed a more symmetrical error distribution. In contrast, the LSTM model demonstrated slightly better accuracy in predicting upward price movements, possibly due to its ability to capture momentum in trending markets. Therefore, in the first

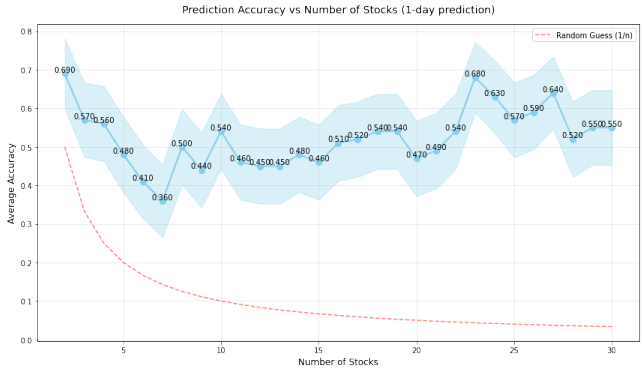

Figure 3: Linear regression model's accuracy in finding the highest yielding option between N=2 to 30 stocks

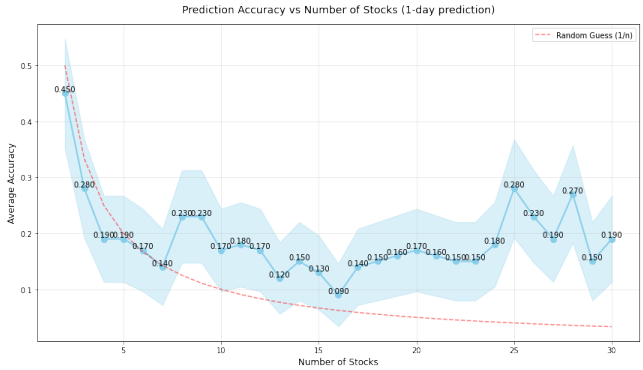

Figure 4: LSTM model's accuracy in finding the highest yielding option between N=2 to 30 stocks

look, the findings challenge some conventional assumptions about the superiority of complex neural network architectures over simpler statistical methods in financial forecasting. Nevertheless, the results make sense considering the daily time frame of the stock price analysis. The linear regression model's strong performance, particularly in accuracy and computational efficiency, suggests that simpler models might be more practical for certain stock market prediction tasks, especially when dealing with stable, large-cap stocks.

As shown in Figure 3, the linear regression model's performance in the day-trading simulation remained remarkably stable even as the analysis expanded from two to 30 stocks. This stability in prediction accuracy across varying numbers of stocks and consistently outperforming random guessing shows the value of the developed models in real-life applications. A detailed breakdown of the prediction results showed that accuracy was highest when comparing stocks within similar market sectors, suggesting that both models effectively captured sector-specific market dynamics.

As shown in Figure 4, the LSTM model also converged on an average accuracy that outperforms random as the number of stocks analyzed increases. However, the difference between the predicted accuracy and random chance is smaller than that of the linear

regression model. Additionally, LSTM predictions were worse than random until the number of stocks exceeded seven. The day-trading simulation shows that both models can provide practical predictions relevant to day trading. However, the linear regression model has an edge over LSTM in making accurate predictions.

Multiple Stock Comparison analysis demonstrates particular value for day traders comparing numerous options simultaneously. The ability to predict daily closing prices across varying numbers of stocks (N = 2 to 30 or more) offers practical utility for portfolio managers and day traders who need to evaluate multiple positions quickly. The consistency in performance when analyzing larger sets of stocks suggests scalability for institutional trading operations where multiple securities are evaluated simultaneously.

5 Conclusion

This study compared the performance of linear regression and Long Short-Term Memory (LSTM) networks for predicting stock prices. Our evaluation reveals some surprising and insightful outcomes. Despite its simplicity, linear regression consistently made more accurate predictions than LSTM. Linear regression achieved an average R^2 value of 0.977, demonstrating its ability to explain a high proportion of variance in the actual stock prices. Surprisingly, LSTM, despite being a more sophisticated model designed for capturing non-linear relationships and temporal dependencies, achieved a slightly lower average R^2 score of 0.935. This result highlights the relevance of simpler models, even in complex domains like financial forecasting.

Linear regression's strong performance may be attributed to the stable nature of stock price trends. This claim is supported by the models' higher performance when predicting high-cap stocks compared to mid-cap stocks. Its deterministic nature and ability to process data rapidly made it an ideal choice for short-term predictions in less volatile scenarios. These advantages were particularly evident when predicting prices for large-cap stocks, where trends tend to follow more predictable patterns. On the other hand, the LSTM network, known for its advanced architecture and capacity to model complex temporal patterns, did not meet expectations in this study. Its slightly lower R^2 value indicates it did not fit market trends as effectively as linear regression. While LSTM is generally well-suited for handling volatile and non-linear data, its advantages were less pronounced in this context.

The trade-offs between the two models became evident when considering practical application scenarios. Linear regression proved to be a faster and more stable option, making it well-suited for situations requiring quick, reliable predictions. In contrast, while LSTM offered the potential to uncover deeper insights in more complex data, its computational demands and marginally lower accuracy than linear regression made it less practical for short-term stock price predictions. The results of this study challenge the assumption that complex data sets require complex models for optimal performance. Further, it demonstrates that the most intuitive model might not always be well-suited for the task.

Our results encourage several future research directions. First, we plan to extend the evaluation to study the model's accuracy over longer periods. This approach uses a predicted closing price as the input for the next day's opening price. The models could simulate longer-term forecasts by iterating this process over multiple days. We suspect that longer-term forecasts would yield larger margins of error. This evaluation would provide deeper insights into the models' performance in real-world, multi-day scenarios.

Second, we plan to expand the dataset to include a wider range of stocks across different market capitalizations, industries, and global markets. We can evaluate how well the models generalize across various scenarios by introducing diverse market conditions. This expanded dataset would also allow us to test the models under different levels of market volatility, helping to refine their limitations further. This direction offers the potential to better understand the practical applicability of the models.

Statement of Work

Leo wrote most of the code for pre-processing the data, training, and testing the models. Mohammad collected and cleaned the data set and wrote the code for the day-trading simulation. Roshan wrote the code for fine-tuning the hyper-parameters. Pranesh identified relevant evaluation metrics and coordinated the presentation slides. All students contributed to the presentation slides and the final report. Specifically, Mohammad wrote the analysis section, Pranesh wrote the literature review and methodology section, Roshan wrote the conclusion, and Leo wrote the abstract and introduction and edited the final report.

References

[1] Hum Nath Bhandari, Binod Rimal, Nawa Raj Pokhrel, Ramchandra Rimal, Keshab R Dahal, and Rajendra KC Khatri. 2022. Predicting stock market index using LSTM. *Machine Learning with Applications* 9 (2022), 100320.

[2] Svetlana Borovkova and Ioannis Tsiamas. 2019. An ensemble of LSTM neural networks for high-frequency stock market classification. *Forest* (2019). https://doi.org/10.1002/for.2585 First published: 21 March 2019.

[3] Kai Chen, Yi Zhou, and Fangyan Dai. 2015. A LSTM-based method for stock returns prediction: A case study of China stock market. In *2015 IEEE International Conference on Big Data (Big Data)*. IEEE. https://doi.org/10.1109/BigData.2015.7363946

[4] CC Emioma and SO Edeki. 2021. Stock price prediction using machine learning on least-squares linear regression basis. In *Journal of Physics: Conference Series*, Vol. 1734. IOP Publishing.

[5] Klaus Greff, Rupesh K Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. 2017. LSTM: A Search Space Odyssey. *IEEE Transactions on Neural Networks and Learning Systems* 28, 10 (2017), 2222–2232. https://doi.org/10.1109/TNNLS.2016.2582924

[6] MEAG Hiransha, E Ab Gopalakrishnan, Vijay Krishna Menon, and KP Soman. 2018. NSE stock market prediction using deep-learning models. *Procedia computer science* 132 (2018), 1351–1362.

[7] Hani A.K. Ihlayyel, Nurfadhlina Mohd Sharef, Mohd Zakree Ahmed Nazri, and Azuraliza Abu bakar. 2018. An enhanced feature representation based on linear regression model for stock market prediction. *Intelligent Data Analysis* 22, 1 (2018), 45–76. https://doi.org/10.3233/IDA-163316

[8] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, and Jonathan Taylor. 2023. Linear regression. In *An introduction to statistical learning: With applications in python*. Springer, 69–134.

[9] Rianchal Jha, Nitin Dixit, Rakhi Arora, Rishi Soni, Vijay Prakash Sharma, and Shreshtha Kinger. 2024. Predicting Stock Market Movements with Linear Regression and LSTM Machine Learning Model. In *2024 IEEE International Conference on Contemporary Computing and Communications (InC4)*. IEEE, Bangalore, India. https://doi.org/10.1109/InC460750.2024.10649281

[10] Mahinda Mailagaha Kumbure, Christoph Lohrmann, Pasi Luukka, and Jari Porras. 2022. Machine learning techniques and data for stock market forecasting: A literature review. *Expert Systems with Applications* 197 (2022), 116659. https://doi.org/10.1016/j.eswa.2022.116659

[11] Oleh Onyshchak. 2020. Stock Market Dataset. https://www.kaggle.com/datasets/jacksoncrow/stock-market-dataset. Accessed October 29, 2024.

[12] Ogulcan E. Orsel and Sasha S. Yamada. 2022. Comparative Study of Machine Learning Models for Stock Price Prediction. *arXiv preprint arXiv:2202.03156* (2022). https://doi.org/10.48550/arXiv.2202.03156

[13] Govind Patidar, Animesh Kumbhakar, Harshit Mahabale, and Juned A Siddiqui. 2023. Stock Market Analysis using Long Short-Term Memory(LSTM) and linear regression Machine Learning Model. In *2023 IEEE 7th Conference on Information and Communication Technology (CICT)*. 1–6. https://doi.org/10.1109/CICT59886.2023.10455550

[14] Kriti Pawar, Raj Srujan Jalem, and Vivek Tiwari. 2018. Stock Market Price Prediction Using LSTM RNN. In *Emerging Trends in Expert Applications and Security*. Advances in Intelligent Systems and Computing, Vol. 841. Springer, 493–503. https://doi.org/10.1007/978-3-319-93163-1_35

[15] Khalid Saboor, Qurat Ul Ain Saboor, Liyan Han, and Abdul Saboor Zahid. 2020. Predicting the stock market using machine learning: Long short-term memory. *Electronic Research Journal of Engineering, Computer and Applied Sciences* 2, 2020 (2020), 202–219.

[16] Ramaswamy Seethalakshmi. 2018. Analysis of stock market predictor variables using linear regression. *International Journal of Pure and Applied Mathematics* 119, 15 (2018), 369–378.

[17] Dev Shah, Haruna Isah, and Farhana Zulkernine. 2019. Stock market analysis: A review and taxonomy of prediction techniques. *International Journal of Financial Studies* 7, 2 (2019), 26.

[18] Shunrong Shen, Haomiao Jiang, and Tongda Zhang. 2012. Stock Market Forecasting Using Machine Learning Algorithms. https://api.semanticscholar.org/CorpusID:16643114

[19] Ralf C Staudemeyer and Eric Rothstein Morris. 2019. Understanding LSTM–a tutorial into long short-term memory recurrent neural networks. *arXiv preprint arXiv:1909.09586* (2019).