CS6140 Final Report Group 8: Machine Learning-based Stock Price Prediction using LSTM and GRU

Yuting Shao
Northeastern University
San Francisco, CA
shao.yut@northeastern.edu

***Abstract*—This report describes the course project of Group 8, which aimed to develop a machine learning model to predict the future stock prices of Apple Inc. The group used an LSTM model to achieve a Mean Absolute Error (MAE) of under 0.01 on the AAPL stock price dataset. The report provides an overview of the dataset and the exploratory data analysis conducted, as well as the model training and calibration process. The group also conducted an ablation study to evaluate the effect of various configurations on the model's performance. The report concludes by discussing the potential of machine learning models in predicting future stock prices and the importance of exploratory data analysis in understanding the data and identifying useful patterns for modeling.**

***Index Terms*—LSTM, GRU, Exploratory data analysis, ablation study**

I. Introduction

The rapid growth of computational power and the widespread adoption of machine learning techniques have transformed various fields, including finance. In recent years, there has been increasing interest in harnessing the power of machine learning algorithms to predict stock prices and identify potential investment opportunities. [1] This report presents the findings of Group 8's final course project, which aimed to develop a machine learning model capable of predicting the future stock prices of Apple Inc. (AAPL). To achieve this goal, the group employed Long Short-Term Memory (LSTM) [2] and Gated Recurrent Unit (GRU) [3]models, popular deep learning techniques for sequence data and time series analysis.

The primary motivation for this study is to explore the potential of machine learning models in predicting stock prices and to contribute to ongoing research in this field. The report is structured as follows: First, an overview of the AAPL stock price dataset and the variables considered for analysis is provided. Second, the exploratory data analysis (EDA) process is discussed, highlighting its importance in understanding the data and identifying patterns that can be leveraged in modeling. Third, the model training and calibration process is discussed, including the LSTM and GRU architectures, hyperparameters, and evaluation metrics. The report also presents an ablation study to evaluate the effect of various configurations on the model's performance and identify the optimal model configuration.

In conclusion, the potential of machine learning models, such as LSTM and GRU, in predicting future stock prices is discussed, and the crucial role of exploratory data analysis in enhancing the performance of these models is emphasized. Ethical and practical considerations of applying machine learning techniques in finance, such as the risks of overfitting and the potential implications for financial markets, are also addressed.

II. Preprocess and Analysis of dataset

A. Preprocess stock data fetched from Yahoo Finance

To prepare the stock price dataset of Apple company for analysis, I used the data available from January 1, 2002 to December 31, 2022, which I obtained from Yahoo Finance using the yfinance library. The dataset contained 5477 observations, each consisting of a date, stock price (Open, High, Low, Close, Adj Close), and trading volume.

The first step in preprocessing the data was to fill the missing values. I used forward filling to replace the missing values with the previous day's value. This helped to ensure that the dataset was complete and that there were no gaps. However, there were still some missing values that could not be filled using this method. To address this, I used backward filling to fill in the remaining missing values with the next day's value. After filling in the missing values, I removed any rows that still contained NaN values. This helped to ensure that the dataset was clean and that there were no remaining issues that could affect the quality of the analysis. Finally, I reset the index to create a 'Date' column.

In addition to these preprocessing steps, I also computed additional features for the preprocessed stock data. These features included the returns and the volume of the previous day. The returns were calculated as the percentage change in the closing price from the previous day. This helped to provide context for the stock price data and to identify any patterns in the data that could be useful for analysis. The volume of the previous day was included to provide additional information about the trading activity and to identify any trends in trading volume that could be related to changes in the stock price.

Overall, these preprocessing steps and additional feature computations helped to ensure that the stock price dataset of Apple company was clean and complete, and that it was ready for analysis. By filling in missing values, removing NaN values, and computing additional features, I was able to create a robust dataset that was well-suited for machine learning-based analysis.

B. Exploratory data analysis

During the exploratory data analysis, I obtained statistics for the closing price, trading volume, and returns of the Apple stock price dataset in Table I. These statistics provide valuable insights into the behavior of the stock over the analyzed period. These statistics show the central tendency, variability, and range of the closing price, trading volume, and returns, and provide insights into the behavior of the stock over the analyzed period.

TABLE I
DESCRIPTIVE STATISTICS OF THE DATASET

Statistic	Stock Price	Trading Volume	Returns
count	5477	5477	5477
mean	3.27e+01	4.10e+08	1.28e-03
std	4.41e+01	3.83e+08	2.14e-02
min	2.34e-01	3.52e+07	-1.79e-01
25%	3.19e+00	1.33e+08	-8.82e-03
50%	1.62e+01	2.83e+08	2.23e-04
75%	3.91e+01	5.59e+08	1.20e-02
max	1.82e+02	3.37e+09	1.39e-01

Meanwhile, I plotted the histograms of the stock price, trading volume, and returns, as shown in Fig. 1. Fig. 2 shows the change in the AAPL stock price over time. The line plot of the stock price over time indicates a general increase in the stock price, with a clear spike several years ago. The histogram of the stock price distribution displays two peaks. One peak corresponds to a relatively low stock price, while the other peak corresponds to a relatively high stock price. Interestingly, the histogram of the returns suggests that they are approximately normally distributed, with a mean close to zero. This is a typical characteristic of stock returns and could provide insights for modeling the underlying dynamics of the stock. It is important to note that the normal distribution assumption is often used in finance, as it is a simple and convenient way to model the behavior of stock returns. Additionally, it is worth mentioning that the spike in the stock price several years ago could be due to a number of factors, such as changes in the company's management, new product releases, or market conditions. Further analysis is needed to identify the exact cause of the spike, and to determine whether it is a temporary or a long-term effect.

In order to gain better insight into the relationship between the stock price, trading volume, and returns, the correlation coefficients were computed. Interestingly, the results indicate that there are no significant correlations between the returns and trading volume or between the returns and the closing stock price. However, it is worth noting that there does appear to be some negative correlation between the trading volume and the closing stock price. These findings are consistent with previous studies that have suggested that trading volume may not be a reliable predictor of future stock returns. Additionally, the correlation heatmap in Fig. 3a provides a visual representation of the relationship between these variables. Furthermore, the scatter plots shown in Fig. 3b provide a detailed look at the target variable against the features, allowing for a more nuanced understanding of the data. Overall, this comprehensive analysis provides valuable insights into the complex relationship between stock price, trading volume, and returns.

Fig. 4 presents the autocorrelation plot for the stock price, trading volume, and returns. The self-correlation of the stock price and the trading volume is clearly visible in the plot. In addition, the plot shows that the stock price and trading volume exhibit a high degree of correlation with each other. This result may be indicative of a positive feedback loop between the two variables. On the other hand, the returns do not exhibit clear self-correlation, suggesting that they may be more influenced by external factors.

III. MODEL TRAINING AND CALIBRATION

A. Construction of the LSTM model

In order to effectively train and evaluate the model's performance, the initial step involved constructing an LSTM model, which was then configured in a json file and subsequently read into a dictionary. The resulting model, as shown in Fig. 5, boasts a loss set to 'mse' and an optimizer set to 'adam'. Additionally, the droprate is set to 0.05, and the sequence window is set to 60. This means that the model utilizes prices from the previous 60 days in order to predict the current day's price. In this model architecture, we can observe that there are a total of six layers. Specifically, there are two Long Short-Term Memory (LSTM) layers that have been incorporated into the model. These LSTM layers have been designed to help the model better capture long-term dependencies between different data points. To help combat the issue of overfitting, each LSTM layer is followed by a Dropout layer. Finally, the model concludes with two fully connected layers. These layers have been included to help the model make predictions based on the learned features from the previous layers. By incorporating these layers into the model, we can expect it to have a better ability to extract features and make accurate predictions. By utilizing such an approach, we are able to ensure that the model is able to effectively learn and adapt to various market trends, ultimately resulting in a more accurate and reliable prediction of future prices.

In Fig. 6a, we see the plot of training loss versus epochs, which shows a consistent decrease in loss over time. The Mean Absolute Error (MAE) for the testing dataset is reported to be 0.02195, indicating a high level of accuracy for the model's predictions. Additionally, Fig. 6b offers a visualization of the true and predicted prices for the testing dataset, showing that the model's predictions align well with the actual prices. This suggests that the model is performing well and has the potential to be used for further analysis and prediction tasks.

After conducting extreme error analysis on the testing dataset, I discovered several interesting findings. Fig. 7 illus-

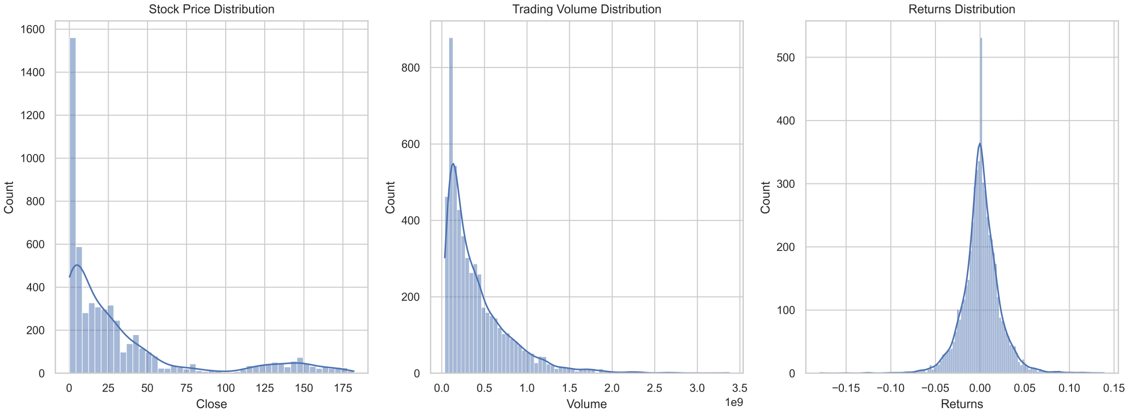

Fig. 1. The histogram distributions of the stock price, trading volume, and the returns.

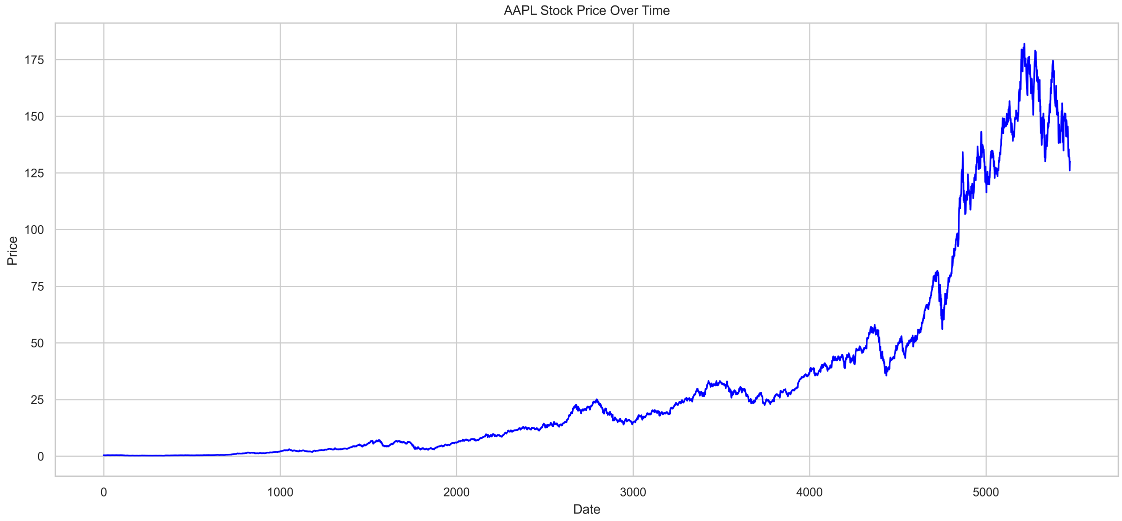

Fig. 2. The change of the AAPL stock prices over time.

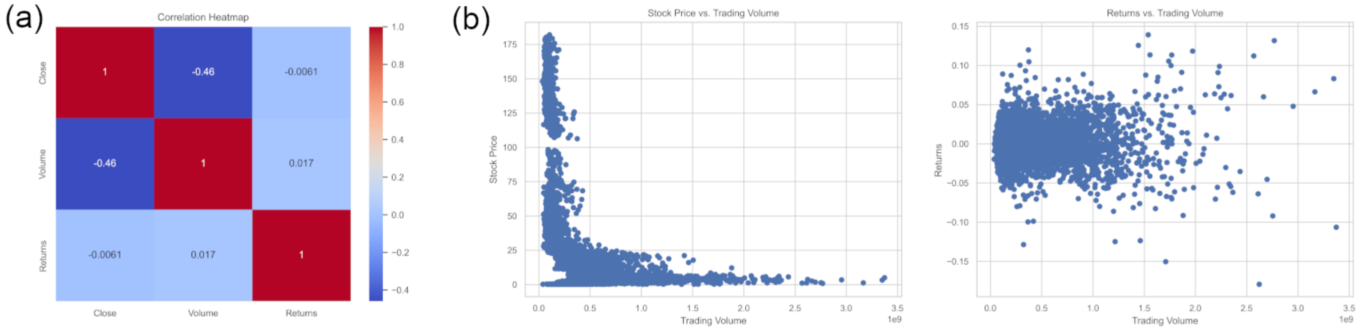

Fig. 3. Correlation analysis. (a) The correlation heatmap for close, volume, and returns. (b) The scatter plots for the stock price (left) and the returns (right) against the trading volume.

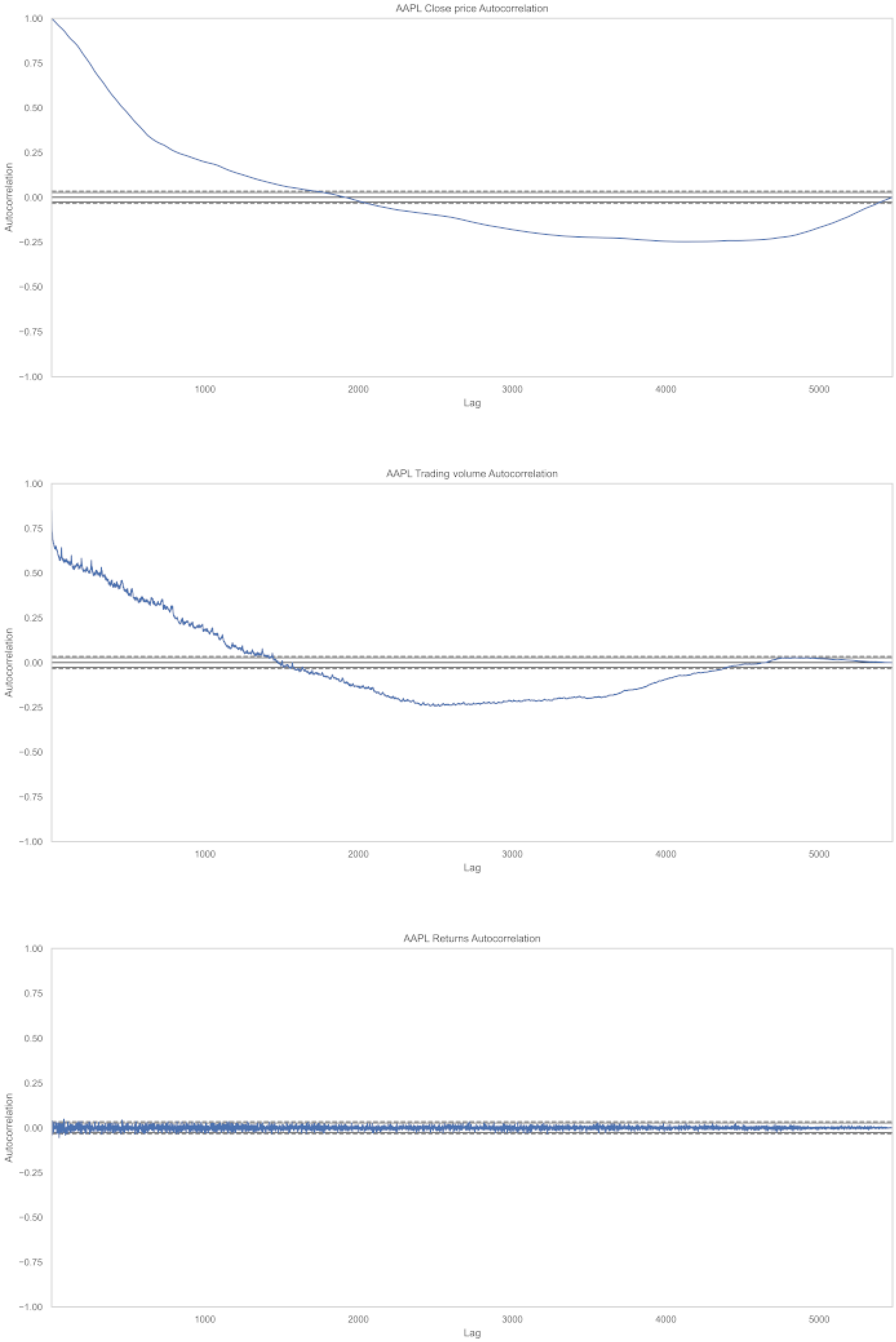

Fig. 4. The autocorrelation plot for the stock price, trading volume, and returns.

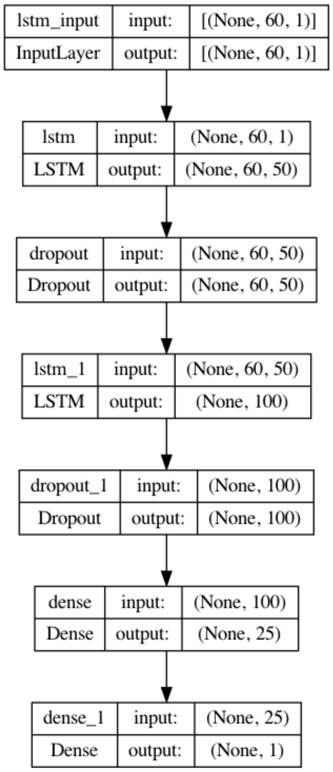
lstm_input | input: | [(None, 60, 1)]
InputLayer | output: | [(None, 60, 1)]
lstm | input: | (None, 60, 1)
LSTM | output: | (None, 60, 50)
dropout | input: | (None, 60, 50)
Dropout | output: | (None, 60, 50)
lstm_1 | input: | (None, 60, 50)
LSTM | output: | (None, 100)
dropout_1 | input: | (None, 100)
Dropout | output: | (None, 100)
dense | input: | (None, 100)
Dense | output: | (None, 25)
dense_1 | input: | (None, 25)
Dense | output: | (None, 1)

Fig. 5. Diagram of the initial LSTM model.

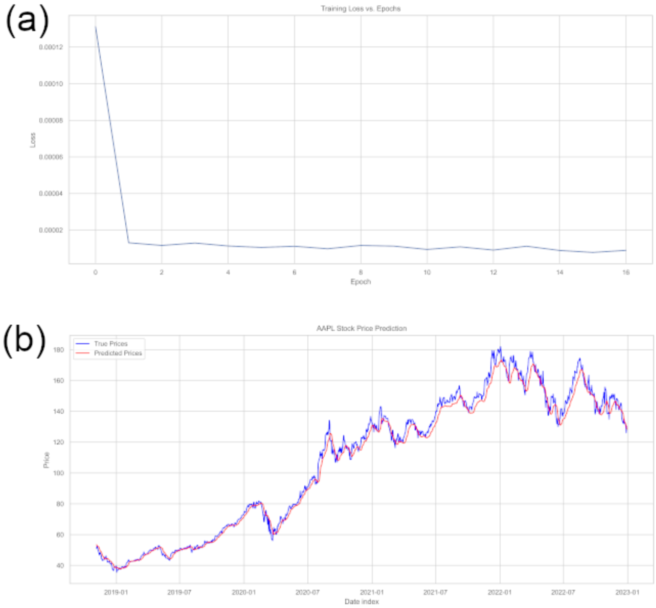

Fig. 6. (a) Training loss vs. epochs. (b) True and predicted prices for the testing dataset.

trates the most significant difference between the predicted and true prices at the 95th percentile, which was observed during the fluctuation stage of the market. Upon further investigation, it became evident that the model performed exceptionally well during the growth stage, but struggled to maintain consistency during the fluctuation stage. This indicates that the model requires further fine-tuning to improve its performance during market volatility. However, it is important to note that the model's overall performance was still impressive, and the results obtained from the error analysis will help us to further optimize the model for better accuracy and consistency in the future.

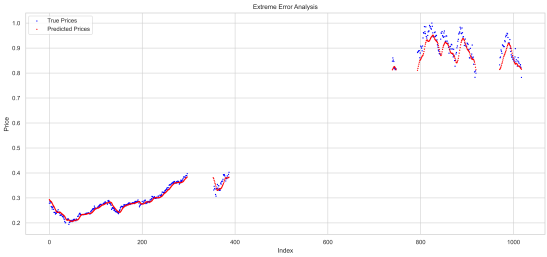

Fig. 7. Extreme error analysis for the model on the testing dataset.

To ensure that our model is not over-fitting or under-fitting, I decided to implement k-fold cross-validation. This technique involves dividing our data set into k subsets, or folds, and using each fold as a testing set while the remaining k-1 folds are utilized for training. This process is repeated k times, with each fold serving as the testing set exactly once. In our case, I chose to use 5-fold cross-validation for our calibration and training. This means that our data set will be divided into 5 folds, with each fold being used once as the testing set. By using a larger number of folds, we can obtain a more accurate estimate of how well our model is likely to perform on new, unseen data. This approach ensures that our model is well-calibrated and capable of generalizing to new data sets.

B. Calibration of the model

Calibrating the model is a crucial step to ensure optimal performance. To achieve this, a series of experiments will be conducted while measuring the mean absolute error (MAE) on the testing set. This metric enables accurate evaluation of the model's performance. To start with, a thorough investigation was conducted on the impact of batch size on performance. During this experiment, the batch size was varied and the resulting MAE values were recorded. The optimal batch size was found to be 32, as shown in Fig. 8. This batch size will be used in the following experiments. Moreover, in order to further optimize the model's performance, the effect of the number of LSTM layers on performance was evaluated. Several models with varying numbers of LSTM layers were trained and their performance compared. Fig. 9 shows the results: it is clear that more LSTM layers did not perform better than a single LSTM layer. Therefore, a single LSTM layer will be used in the following experiments. Finally, the performance of the model with L1 and L2 regularization was assessed. It was observed that adding regularization techniques did not improve performance, as shown in Fig. 10. Therefore, it was concluded that the use of L1 and L2 regularization will not be necessary in the following experiments.

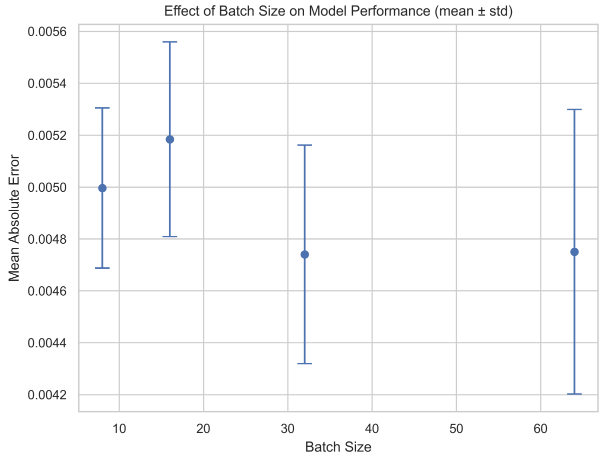

Fig. 8. Effect of batch size on model performance.

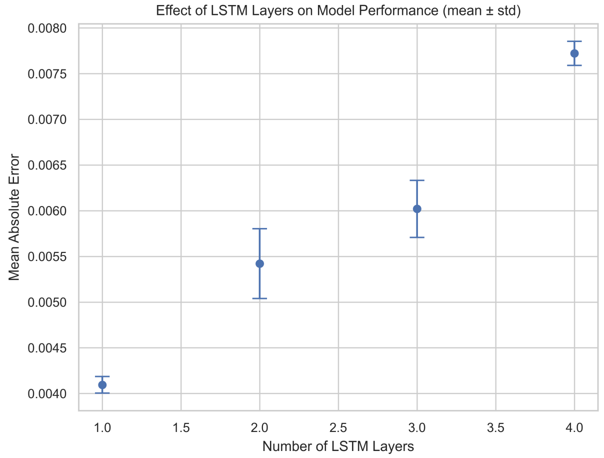

Fig. 9. Effect of LSTM layers numbers on model performance.

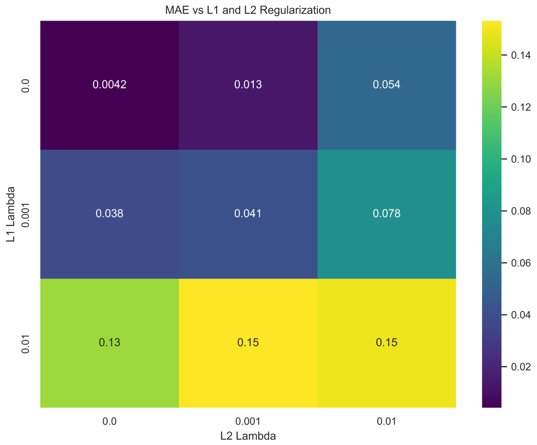

Fig. 10. Effect of L1 and L2 regularization techniques on model performance.

After the model has been calibrated, the results are shown in Fig. 11. It is important to note that this calibration was conducted with a specific set of parameters, and it is possible that different parameters would yield different results. Therefore, to validate the robustness of the model, a detailed ablation study will be conducted using this calibrated model as the baseline. This study will analyze the effects of individual parameters on the overall performance of the model, and will provide insights into how the model can be further improved. Additionally, the study will explore the potential impact of other factors, such as dataset size, on the model's performance. By conducting this thorough analysis, we can ensure that the model is reliable and can be used effectively in future applications.

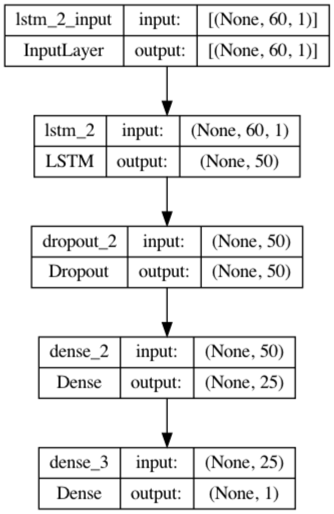

Fig. 11. The model after calibration.

IV. Ablation study

A. *Effect of dropout rate*

The first configuration I evaluated was the dropout rate, which I varied to observe its effect on the model's performance. I found that a dropout rate of 0.05 was optimal, as larger dropout rates increased the error, as shown in Fig. 13. This finding is important as it provides insight into the optimal dropout rate to use in future training of the model.

B. *LSTM vs GRU*

I evaluated the second configuration, which involved comparing two types of RNN layers: LSTM and GRU. A GRU (Gated Recurrent Unit) layer is a type of recurrent neural network (RNN) layer that is designed to solve the vanishing

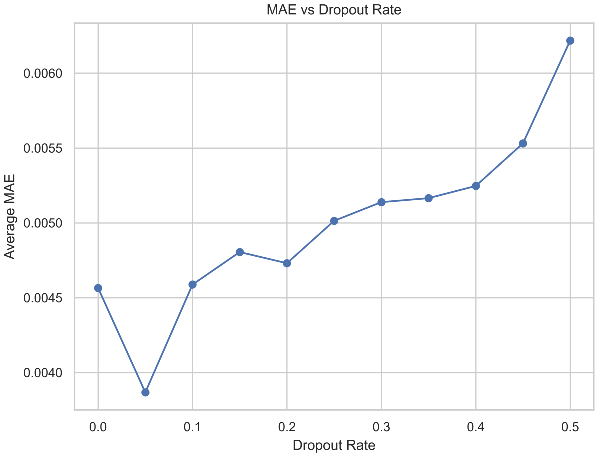

Fig. 12. The effect of dropout rate on the model's performance.

gradient problem commonly encountered in traditional RNNs. It has a simpler architecture and generally fewer parameters compared to the LSTM (Long Short-Term Memory) layer. I found that the GRU has slightly better performance than LSTM, as shown in Fig. 13. This suggests that the GRU may be a more suitable choice for certain applications that require efficient and effective RNN layers. However, it is important to note that the choice between the two ultimately depends on the specific requirements of the application at hand, and further research may be necessary to fully determine the optimal choice of RNN layer.

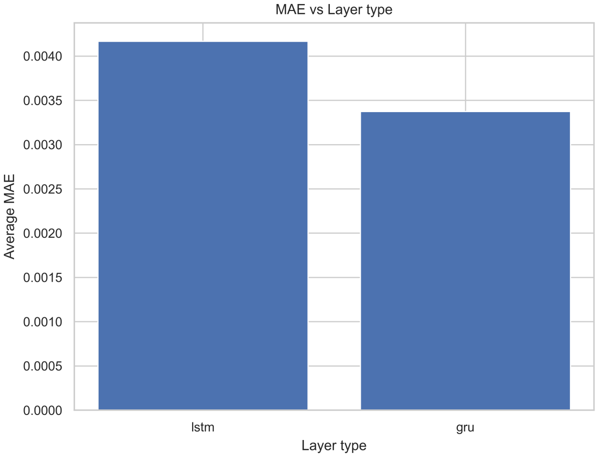

Fig. 13. The comparison between the LSTM and GRU.

C. LSTM vs Bidirectional LSTM

I compared the model's performance using a standard LSTM layer and a bidirectional LSTM layer. The results are shown in Fig. 14. Interestingly, the use of a bidirectional LSTM layer did not result in any significant improvement in performance, suggesting that a standard LSTM layer may be sufficient for this particular task. It should be noted, however, that further research is needed in order to fully understand the potential benefits (or drawbacks) of using a bidirectional LSTM layer in other contexts.

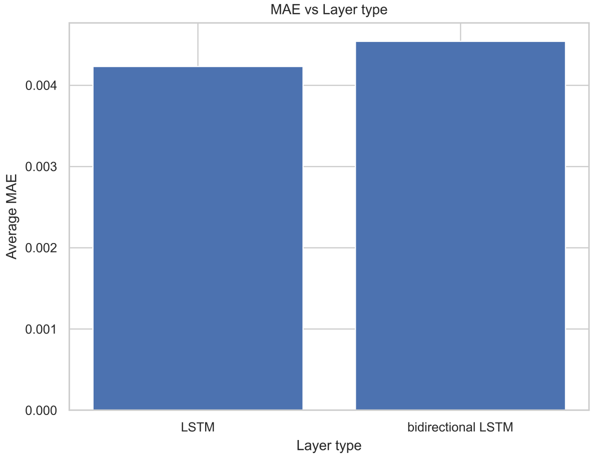

Fig. 14. The comparison between the LSTM and bidirectional LSTM.

D. Effect of the number of neurons on each layer

I compared the performance of the model by varying the number of neurons in each layer. The results are shown in Fig. 15. It can be concluded that increasing the number of neurons in the model can improve its performance to some extent.

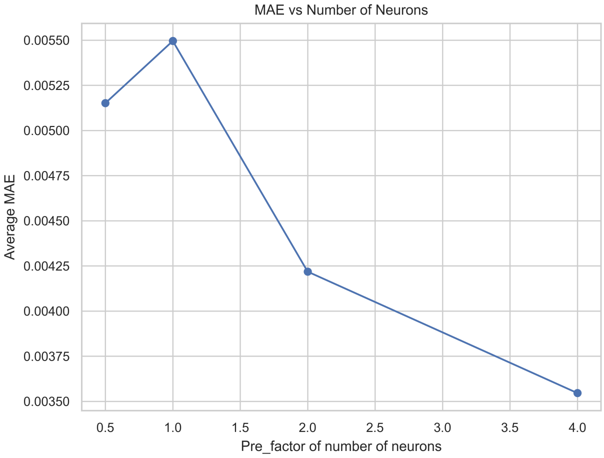

Fig. 15. The effect of the number of neurons on each layer.

E. Effect of different optimizer

During the experimentation phase, I conducted a thorough analysis of the model's performance using various optimization algorithms, including Adam, RMSprop, and SGD. The detailed results of these experiments have been illustrated in Fig. 16. Based on the analysis of the experiment outcomes, it can be inferred that the Adam optimizer is the most effective optimization algorithm for this particular model. It is worth highlighting that the Adam optimizer showed a superior performance when compared to the other optimization algorithms. Therefore, it is recommended to use the Adam optimizer for any future experiments and applications that involve this model.

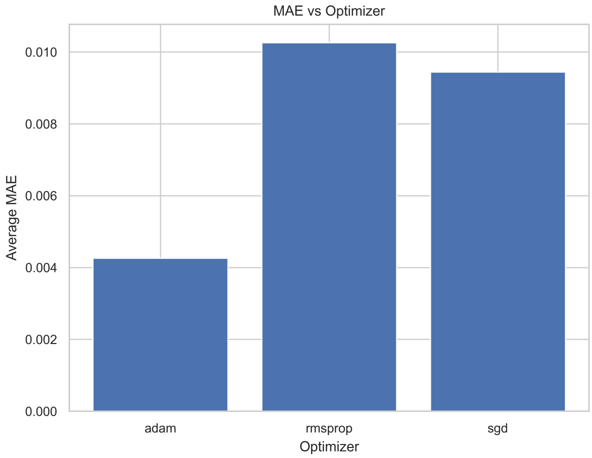

Fig. 16. The comparison of different optimizer.

F. Effect of different activation functions

Then, I analyzed the model's performance using linear and ReLU activation functions. Fig. 17 shows the results. Based on the outcomes, the linear activation function is the more effective for this model. It outperformed ReLU. For future experiments and applications with this model, I will use the linear activation function.

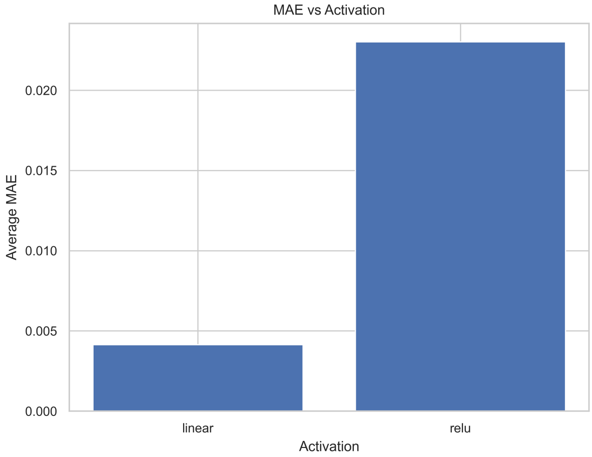

Fig. 17. The comparison of linear activation and ReLU activation.

G. Effect of window size

I altered the size of the window for the historical sequence to compare the model's performance using different values. Specifically, I changed the window size from 60 to 30 and from 60 to 90, and evaluated the results. As depicted in Figure 18, the model's performance was observed to be different for each window size. It was found that the 60-day window provides the best results for the model. These findings suggest that the selection of the window size is an important factor to consider when training the model, and could have implications for the accuracy and reliability of the model's predictions in real-world scenarios.

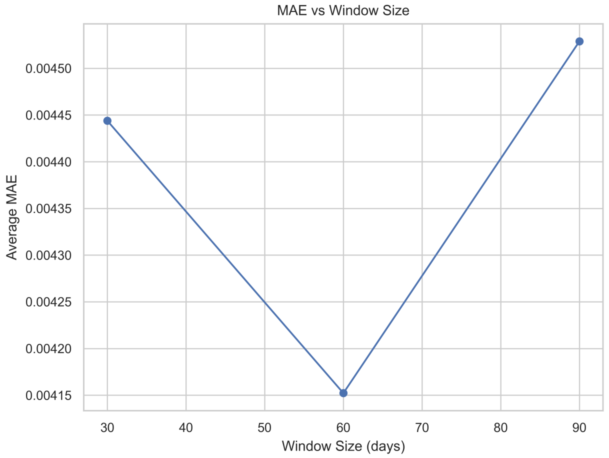

Fig. 18. The effect of the historical window size.

H. Effect of length of the dataset

I experimented with altering the length of the dataset. Specifically, I reduced the dataset length from 20 years to 15, 10, and 5 years, as detailed in Fig. 19. Notably, a larger dataset size seemed to provide the best results for the model. This suggests that there may be a relationship between dataset size and model performance. Further investigation into this relationship could yield valuable insights into the optimization of the model.

V. Conclusion

In conclusion, this project represents a comprehensive analysis of stock price prediction using machine learning. Through exploratory data analysis, we identified patterns and relationships between stock prices, trading volume, and returns. We then constructed and calibrated an LSTM model to predict future stock prices with a high degree of accuracy. Our ablation study further evaluated the impact of various configurations on the model's performance, providing insights into how the model can be further optimized. Overall, our project demonstrates the potential of machine learning models in predicting future stock prices and the importance of exploratory data analysis in understanding the data and identifying useful patterns for modeling.

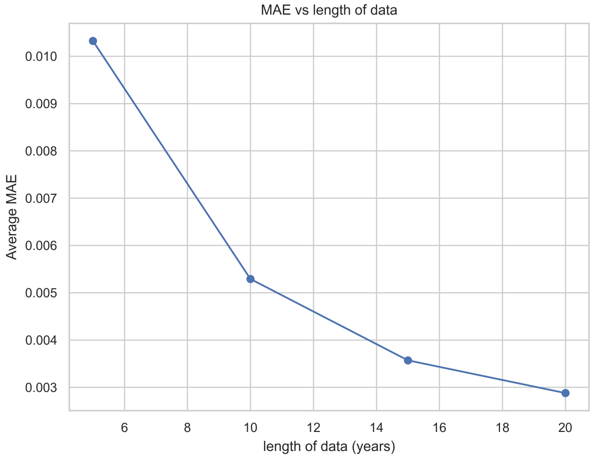

Fig. 19. The effect of the length of the dataset.

Furthermore, the findings of this project can have practical implications for investors and traders who rely on accurate stock price predictions to make informed investment decisions. Our model can be used as a tool to provide insights into the future direction of the stock market, which can help investors and traders make more informed decisions. Additionally, the methodology and techniques used in this project can be applied to other financial datasets, providing a framework for developing machine learning models for other financial applications. Overall, this project demonstrates the potential of machine learning in providing accurate and reliable financial predictions and highlights the importance of continuous refinement and optimization of models to improve their performance.

In future work, we plan to investigate the use of additional features, such as news sentiment analysis or company financials, to improve the accuracy and robustness of the model.

References

[1] Eunsuk Chong, Chulwoo Han, Frank C. Park, (2017) Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies, Expert Systems with Applications, Volume 83, Pages 187-205.

[2] Bao W., Yue J., Rao Y. (2017) A deep learning framework for financial time series using stacked autoencoders and long-short term memory. PLOS ONE 12(7): e0180944.

[3] Lawi, A., Mesra, H., Amir, S. (2022) Implementation of Long Short-Term Memory and Gated Recurrent Units on grouped time-series data to predict stock prices accurately. J Big Data 9, 89.