Predicting prices in stock and currency markets using Machine Learning algorithms

Klevis Kokoshi  
190861850Chris Phillips  
Msc Computing and Information Systems

*Abstract* — This paper delves into the integration of Machine Learning (ML) algorithms within a trading software to optimize trading performance. It begins by critically examining the theoretical framework that steered this research, highlighting the existing work in the field as well as the absence of an end-user application. Consequently, it presents a revolutionary trading software architecture, meticulously crafted by blending ML algorithms, classic indicators and sentiment analysis. Notably, the system employs a Bidirectional LSTM model in a sequence-to-sequence configuration, predicting a week's worth of stock prices based on historical data. This approach is augmented by the capability to fetch real-time stock data and allows users to select specific features for modeling. The accuracy of the model is evaluated using metrics such as RMSE, MAPE, and r2 score, providing a comprehensive view of its performance. In the subsequent sections, an in-depth analysis of the technical components of the software is provided, emphasizing the more novel characteristics that enhance trading efficiency. The paper will highlight the process of experimenting with different parameters to find the most suitable conditions to train the model. Ultimately, the paper contends that while the integration of ML in trading software can accurately predict stock prices and thus boost trading performance, the practical application and real-world effectiveness of such systems necessitate continuous refinement based on extensive testing, user feedback, and market dynamics. The efficacy of this project lies not just in the sophistication of its algorithms but also in its ability to visually interpret confluence between classic trading indicators.

Keywords: Machine learning, Trading, Financial indicators, Recurrent Neural Networks, Bidirectional Long Short-Term Memory, Sentiment analysis, Stock Price Prediction

# Introduction:

Trading stocks has long been a focal point in discussions surrounding global financial markets, serving as a potent avenue for wealth creation and economic expansion. At its heart, stock trading is a nuanced interplay of numbers, forecasts, and strategies. Yet, it's impossible to overlook the profound influence of human psychology. Emotions, ranging from fear and greed to hope, often steer investors and traders through the unpredictable terrains of the stock market, sometimes leading them to decisions that stand at odds with logical or fundamental analysis.

In our current information age, the realm of stock trading is frequently overshadowed by 'noise'. This term encapsulates the overwhelming barrage of often extraneous information, be it media speculations, market rumors, or the relentless stream of real-time data. Such noise clouds judgment, complicating the task of discerning genuine market signals from distractions. It's against this backdrop that the imperative for objective, data-driven decision-making gains prominence.

To tackle the challenges posed by human psychology and the noise in financial markets, machine learning, especially neural networks, offers a promising solution. Among these, the Long Short-Term Memory (LSTM) models have gained prominence in time series forecasting, thanks to their adeptness at recognizing patterns over extended sequences. Yet, their unidirectional data processing can sometimes miss crucial information. This is where the Bidirectional Long Short-Term Memory (BI-LSTM) model shines. By analyzing data in both forward and backward directions, it ensures a comprehensive capture of patterns, providing traders and analysts with a richer, more holistic view of market trends. Such enhanced insights can be pivotal in refining trading strategies and making more informed decisions in the volatile world of stock trading.

Drawing from my personal journey in machine learning, I've encountered the pertinent issue of overfitting more than once. This challenge is magnified in the domain of financial data analysis, characterized by its volatility and intricate nature. Recognizing these intricacies and armed with insights from past experiences, this project narrows its focus to short-term stock price predictions. The rationale is twofold. Short-term predictions can potentially sidestep the issue of overfitting, focusing on less data and the bidirectional capabilities of BI-LSTM position it as an optimal tool for this task. With BI-LSTM being the model employed, the project aims to discern short-term market trends with heightened precision, ensuring reliable predictions in the often-tumultuous financial markets.

In the complex realm of stock trading, the capacity to make well-informed decisions holds the greatest value. Although machine learning offers powerful techniques for forecasting stock prices, these projections may occasionally appear theoretical or disconnected from conventional trading knowledge. In light of this recognition, a decision was made to design a user-oriented tool that effectively combines machine learning forecasts with conventional technical indicators. This integration not only utilizes the forecasting capabilities of sophisticated algorithms but also anchors these predictions in well-established trade parameters. By employing this approach, the software offers traders and analysts a comprehensive perspective, allowing them to critically evaluate machine learning predictions within the framework of well-established technical indicators aswell as typical error scores employed in evaluating ML models. This technique guarantees that consumers may leverage the advantages of both machine learning, with its advanced capabilities, and traditional stock analysis, which offers a familiar framework. The primary objective of our program is to provide users with a wide range of information, thereby enhancing their confidence in making informed trading selections.

The paper will demonstrate empirical testing, conducting experiments to find the optimal parameters for the BI-LSTM model. The configuration that yields the best results evaluated with typical error metrics and otherwise will be then put to the test, assessing its performance on stocks and currencies not yet trained on. Ultimately the paper will assess the efficacy of BI-LSTM, predicting stock and currency prices given its ability to handle a significantly noisy dataset. All the while providing a user-friendly experience made possible through the medium of a tailored web application.

The aims can therefore be summarised by the following:

* Experiment on different parameters to find suitable ones to train the BI-LSTM model on.
* Demonstrate the ability of the BI-LSTM model aswell as other(s) to help identify short term market trends and make predictions. (Even in stocks and currencies it has not been trained on).
* Create a unique user-friendly tool that enables users to utilise machine learning in conjunction with established technical indicators and knowledge.
* Evaluate the BI-LSTM model’s performance using typical error metrics and comparing it to an ‘un-optimized’ XGboost model.
* Investigate the use of news articles in providing sentiment scores on a given stock or currency.

# Literature Review:

The use of Long Short-Term Memory (LSTM) networks, a type of recurrent neural network (RNN) has shown significant promise in predicting stock market trends. A study by Nabipour et al. (2020) compared nine machine learning models and two deep learning methods, including LSTM trained with ten years of historical data from four stock market groups. The study found that for continuous data, LSTM outperformed other prediction models by a considerable margin.

In another study, Patil et al. (2020) proposed a novel approach using graph theory along with deep learning and traditional machine learning techniques for stock market prediction. Although the study did not specifically focus on LSTM models, it highlighted the potential of combining different machine learning techniques for improved prediction accuracy and thus forms the basis for implementing XG boost as well as BI-LSTM in this project. The user will be able to see the plotted predicted/actual stock price with the BI-LSTM model and can then choose to view the predictions forecasted by the XGboost model.

A study by Aslam et al. (2022) demonstrated the effectiveness of an ensemble model combining LSTM and Gated Recurrent Unit (GRU) for sentiment analysis and emotion detection on cryptocurrency related tweets, outperforming both machine learning and state of the art models. This was particularly interesting as it highlights the potential of sentiment analysis in making informed decisions when trading. It is also specifically interesting in cryptocurrency given that it is a relatively new ‘emerging market’ and speculation and emotions are significantly more influential in shifting price and trends whereas traditional commodities and stocks are heavily controlled by market makers. This was convincingly explained by Aste (2019) whose analysis of almost two thousand cryptocurrencies traded during the first six months of 2018, uncovers a complex and rich structure of interrelations where price and sentiment influence each other instantaneously and otherwise.

Another study that demonstrated the strength of sentiment analysis in predicting market trends or movements was a study by Zhang, et al. (2011) that found market sentiment, as measured by the Twitter mood, could predict the daily ups and downs of the Dow Jones Industrial Average. They found an accuracy of 87.6% in predicting the daily up and down changes in the closing values of the Dow Jones Industrial Average and a reduction of the Mean Average Percentage Error by more than 6%.

Remani et al. (2023) predicted the price of Bitcoin using LSTM, XGBoost and sentiment analysis on dynamic streaming data, they concluded that LSTM ‘price prediction performance is enhanced with sentiment score in combination with LSTM technique’. This conclusion aligns with other related works that also suggest LSTM networks have the potential to be used in financial forecasting. However, their mean absolute error, root mean square error, and r-squared (r2) error numbers did not align with this conclusion. The addition of sentiment analysis data with the LSTM model worsened the mean absolute error score, the root mean square error, and brought the R2 error closer to the required range (0-1) but was still outside of the normal range. Given the scope of our project and the previous research that emphasized the advantages of sentiment analysis, my project will integrate sentiment analysis as an independent metric, rather than using it to train the machine learning model. This is to prevent the issue of overfitting and due to the ambiguous nature of sentiment analysis which can struggle with context and sarcasm.

Pramudya, (2020) conducted a study to assess the effectiveness of different indicators in generating precise buy and sell signals for the LQ45 index. The study examined the MACD, Bollinger Bands, and Relative Strength Index, with the results indicating that merging various indicators can yield a better understanding of financial data - thus facilitating well-informed decision-making. The primary objective of this project is to train and showcase machine learning models. However, it is equally crucial to enable end users to observe the correlation between the predictions generated by the model and the traditional indicators that they are familiar with.

Li (2022) comparative study of LSTM variants in the prediction of Tesla’s stock price found BI-LSTM to be the least accurate of the LSTM variants when compared to Vanilla, Stacked and CNN LSTM. After 10 experiments on a Tesla dataset between 2020-2022 their BI-LSTM model had an average 48.29 RMSE score and an accuracy of just under 70% with other models all above 70%. In the second dataset with financial data between 2018-2022, BI-LSTM outperformed all other LSTM variants by the RMSE metric and had a 70.1% accuracy prediction instead of 67.76% accuracy in the ‘Vanilla LSTM’. Further reading into Li’s study found that they did not utilize scaling or seem to tune parameters to find the optimal conditions to train their models. Interestingly the study demonstrated that BI-LSTM performed better in predicting stock prices when trained and tested on more data.

# Datasets processing

The data used for the candlestick graph as well as the training, testing, future predictions and evaluations of the model was sourced using the ‘yfinance’ library. This library provides reliable historical stock data from Yahoo finance, in this instance fetching data for specific stock symbols starting from January 1st, 2022, to the present day. In addition, news articles related to the selected stock are fetched using the NewsApiClient from the newsapi library. This data is used to output sentiment scores, expressing the confidence captured in the title of articles about a given stock. The titles are analyzed using the ‘TextBlob’ library to output sentiment scores. From the articles fetched, the polarity of each title is calculated an average polarity score is calculated. This score is converted to be a % out of 100 and output in the application in different colors based on the sentiment.

The stock data sourced from yfinance includes the data columns of ‘Open, High, Low, Close and Volume’. The user can select any combination of these as the input for the machine learning models. This allows user flexibility and gives them more choice with what they want to feed the model to base the predictions off.

Furthermore, the code enables the computation and visualization of typical technical indicators, namely Simple Moving Average (SMA), Exponential Moving Average (EMA), Moving Average Convergence Divergence (MACD), Relative Strength Index (RSI), and Bollinger Bands. These indicators are popular tools in technical analysis and provide valuable insight into price movements.

# Methodology:

The development of this web application was undertaken using Visual Studio Code, a powerful code editor developed by Microsoft. VS code optimized the software development process, particularly due to its syntax highlighting and integrated debugging, it was easier to spot errors and aided in the testing of the software. It further aided with the importing of libraries and ensuring that everything was installed correctly using the built-in terminal to run commands. These small differences between the python IDLE and VS code enhanced the development experience of this stock and currency predicting application.

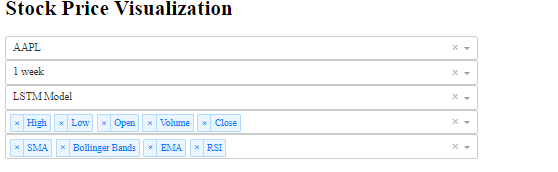
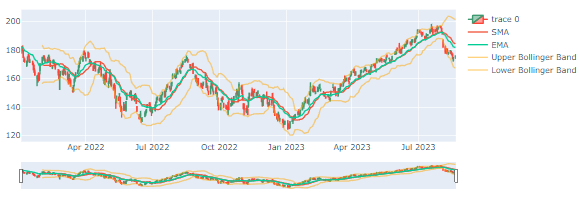
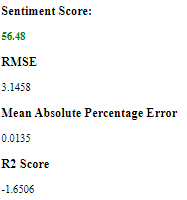
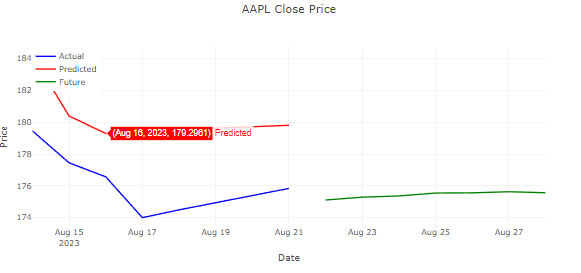
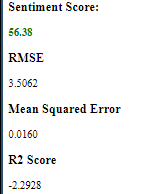
Predicting stock prices is challenging given that there are many factors that can influence price movements, these include but are not limited to, market sentiment, economic data as well as specific company news. The aim of the project was to as accurately as possible, predict the close of the stock price in the short term and present the predictions and fluctuations in a comprehensive manner using a BI-LSTM model. To do this, an experiment was conducted to find the optimal parameters for the model.

This experiment evaluated different configurations for time series forecasting using the BI-LSTM model, with variations in the financial data history (30, 60, 100, 180 and 365 days), steps (2 and 4), and batch sizes (2 and 4). Each iteration produced visual plots comparing the predicted vs. actual closing prices, along with RMSE and MAPE scores to quantify the prediction accuracy.

As described in the abstract, the project aims to visualize stock price data, forecasting future stock prices using either the BI-LSTM or XGBoost model. The absence of end-user software encouraged the development of our web interface. The dash framework was imperative in enabling this web application development, utilizing dropdown menus to allow users to select specific stocks, time intervals, prediction models as well as technical indicators. In addition to enabling these key features, changes in inputs reflect immediately in the output. There is no buffer time (except for when the BI-LSTM model is performing its machine learning) and this too contributes to the user experience. The graphs and dropdowns were integrated using Dash’s ‘dcc’ and ‘html’ modules, as well as plotly, a popular library for plotting interactive graphs. These graphs display the stock predictions and allow the users to zoom and hover over data points to get precise values, offering greater insight into the stock’s behaviour.

Key to the success of a stock prediction application is its flexibility and ease of use. The dropdown menus and the ability to change between stocks, the selection of data columns (open, high, low, close volume) as well as the responsive candlestick charting ensure that users can quickly gain insight from the machine learning model and see the confluence with technical indicators.

Also essential to the success of the project was the ability to import real-time stock data, compute and present technical indicators, perform predictions and evaluate them in real time via RMSE, MAPE, and R2 score. These metrics were computed using the sklearn library. Real time financial data was obtained through the yfinance (yahoo finance) module. The Pandas library and TensorFlow aided in data manipulation, specifically in computing technical indicators and to design and train the BI-LSTM model respectively. Additionally, the Keras module, integrated with TensorFlow, provided a high-level neural network API that streamlined the process, enabling the stacking of layers in the BI-LSTM model. The web application allows users to select the last week of actual vs predicted prices, or the last two weeks. This, in conjunction with the error scores, allow the user to evaluate the performance of the model over these timeframes, giving them even more information. The use of the mentioned libraries facilitated a working and responsive web application, fig. 1, 2 and 3 (resized for the purpose of this screenshot) demonstrates the ability to visualize results of the BI-LSTM model aswell as seeing confluence with technical indicators.



*Figure 1, 2, 3*

The evaluation metrics namely RMSE, MAPE and R2 are suitable choices for stock price prediction. RMSE or root mean square error, is the square root of the average squared different between predicted and actual values, this gives a higher weight to larger errors. The MAPE or Mean Absolute Percentage Error is the average of the squared difference between predicted and actual values. It also penalizes larger errors and is a common metric used in machine learning models; In both the RMSE and MSE metrics, the smaller the number the better the model is at predicting values compared to the actual values. The R2 metric is a statistical measure that represents the proportion of the variance for the stock price, explained by independent variables in the models. R2 thus provides an understanding of how accurately the model’s predictions match the real data. An adequate range here is between 0-1 with 1 indicating that the model has captured all variability in stock prices (the larger the number, within the range, the better). These metrics are commonplace in time series stock prediction and are suitable for a regression task demonstrating deviations from the actual price and accounting for variability in predictions.

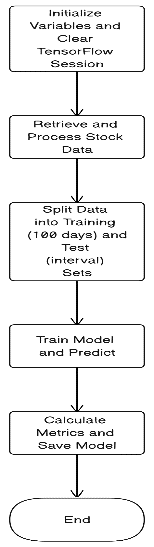
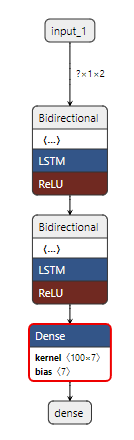


Figure 4 Figure 5

Figure 4 Figure 5

Following the research conducted on time series forecasting in stocks using the LSTM model, our project employed a Bidirectional Long Short-Term Memory neural network. Whilst LSTM emerged as a popular form of a Recurrent Neural Network used to predict stock and currency prices, a decision was made to consider a bidirectional approach. This decision was motivated by the potential of Bi-LSTM’s to understand patterns that may have been missed with the unilateral counterpart. Thus, providing a rich context and potentially enhancing prediction accuracy.

In order to feed the BI-LSTM model, the data sourced from yfinance needed to be split into overlapping sequences suitable for LSTM models. In each sequence, past sock prices are paired with the stock’s closing price for the future 7 days. This data is subsequently divided into training and testing sets based on the users defined time interval, 1 week or 2 weeks. In order to optimize the model’s performance, the application makes use of the MinMaxScaler to scale the closing prices between 0 and 1. This ensures that the BI-LSTM model operates faster and negates the issue of exploding and vanishing gradients. This is a typical issue with RNN models, when the gradient of the loss function becomes very small, the deep neural networks stops learning or learns very slowly. Similarly, when the gradient is ‘exploding’ or increasing, the RNN can become unstable. Once the data is prepared, the BI-LSTM model is trained, utilizing the early stopping function to again, prevent overfitting.

To execute the machine learning model, the project had to initialize variables and clear the TensorFlow session to ensure previous runs did not affect subsequent ones. Then, the stock data is retrieved from yfinance and processed, splitting the data into training (100 days) and test (user input interval) as shown in Fig, 4. The first layer of the model encompasses 50 units and utilizes the Rectified Linear Unit activation function (‘relu’), one that is widely used in neural networks providing computational efficiency as well as enabling the model to learn from errors and make corrections. This first layer returned a sequence to be fed into the second Bi-LSTM layer, which also comprised of 50 units and the relu activation. This second layer produced a fixed-size vector containing the features of the input sequences. Following these layers, a dense layer with seven units was integrated which was responsible for generating final predictions, see fig. 5.

The model is then compiled and optimized using the ‘Adam’ optimizer. Ando and Takefuji (2021), found that Adam (Adaptive Moment Estimation) is one of the promising techniques for parameter optimization of deep learning. The model then presents the last week’s prediction vs actual prices and forecasts the closing prices for the following 7 days, providing the user with machine learning propagated insights into stock or currency price movements and outputting the error metrics.

The application also made use of sentiment analysis to provide the user with even more information to make a well-informed decision. The sentiment analysis was achieved using the NewsAPI to source news articles, focusing on articles that included the stock symbol and terms like ‘prediction’ or ‘news’. The titles of these news articles were then analysed using the TextBlob library which provided a polarity score ranging from -1 (negative) to 1 (positive). The average polarity score for the selected stock was taken and transformed to a scale of 0 to 100. Testing found that this did indeed print different sentiment scores for each stock that mostly corresponded with the performance of the stock, however, scores ranged from 49 (Nike) to 56.48 (Apple). Based off these scores, a decision was made to color the output, below 45 would be negative, 45-55 would be considered neutral, with 55+ being positive.

# Testing and Evaluation:

In the system testing phase of our software development, a significant issue was discovered pertaining to the selection of 'volume' as a factor in training the model. The implementation of this feature resulted in bursting gradients previously described in our methodology. This was because stock and currency prices can have very large trading volumes. This led to a significant increase in the training loss of the model and thus resulted in very inaccurate predictions. In order to tackle this difficulty, we converted the volume column into a volume percentage change column instead. This change successfully addressed the issue of the exploding gradient, and this can be seen in fig. 6 & 7, resulting in a more stable training procedure and improved predictive performance of the model whilst allowing volume to still be a choice of data column.

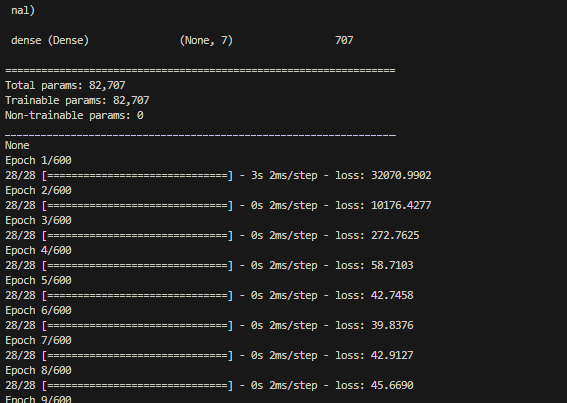
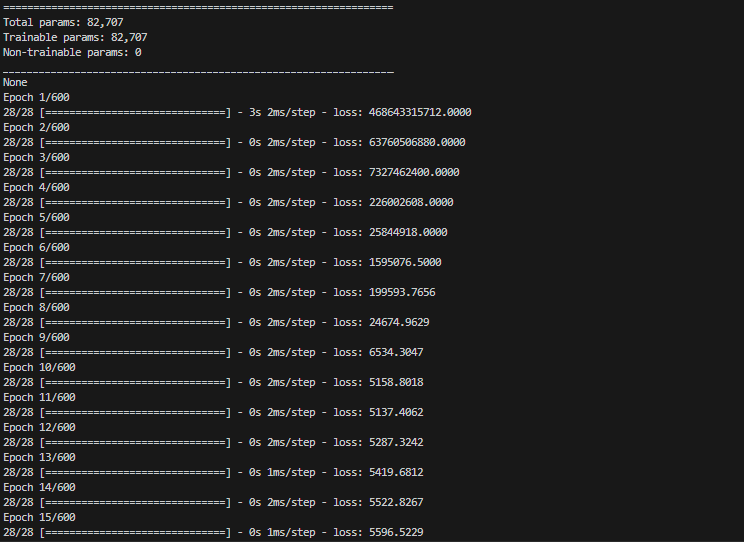


Figure 6

Figure 7

Sentiment scores were all similar with the exception of ‘Nike’ stock which exhibited a below 50 sentiment score, this is perhaps because the content from NewsApi article titles is not enough to determine the sentiment of any given stock. These articles may have been too neutral given that using either titles or descriptions yielded similar results.

When testing each stock / currency listed in the software to evaluate the performance of the predictor on different category stocks and currencies. I encountered an issue of ‘KeyError’ when attempting to plot the graph for predicted vs actual price. This could have been due to the fact it was in the early hours of the day, perhaps because the yfinance data column was not ready for the following day. Fig. 8 and 9, below demonstrate the error and the solution found. A variable named ‘valid\_dates’ was used to ensure that the latest data frame had valid entries for the given dates, with only these being plotted.

A screenshot of a computer

Description automatically generated

Figure 8



Figure 9

Upon gathering results for different stocks and currencies, I came across an issue. Refreshing the application resulted in a different output for the error metrics, this was an issue given that the financial data could not have changed that much every refresh to warrant such changes in errors. To fix this, I initiated a random seed to generate at the start of each code, selecting the same number for each to hopefully ensure the same results in each run. I found that closing the application fully and restarting it would generate the same results on the initial run, but subsequent runs following a refresh would change the output. Using the VS code terminal, I analyzed the output of the date time index and found that subsequent runs following a refresh used a different time index and this prompted a change in the code. Now every time the application is launched, the BI-LSTM model and XGB\_model would be set to None. In addition to this, I created a function to reset the LSTM model on stock change to ensure no data carried over influencing the model’s predicitons. Following these changes, the application worked as intended, with all user input functions, drop downs, plotting of graphs and interactivity working seamlessly. The testing of the model itself is evidenced and discussed in the results section.

# Results

With the aim of optimizing the BI-LSTM’s prediction accuracy, a series of experiments were conducted by adjusting the key parameters: days, steps and batch size, as shown in figure 12. This experiment involved fetching data from yfinance, preparing the data for training and testing the BI-LSTM by splitting it, defining and training the model as in the methodology and calculating error scores for each iteration in the experiment loop. These experiments demonstrated that Shorter historical data (30 and 60 days) yielded worse RMSE and MAPE scores, despite the model needing to make short term predictions. This could be because models trained on shorter histories could be overly sensitive to recent changes, potentially leading to volatile and reactionary predictions as well as missing potentially significant price movements prior to this time period.

A screenshot of a computer

Description automatically generatedThe 100-day model effectively traced stock price movements, achieving the best RMSE (3.21) and MAPE (0.0084) scores with parameters of steps=4 and batch\_size=4. Contrary to literature suggesting better accuracy with larger datasets, our 180 and 365-day experiments showed increased error, possibly due to noisy data or overfitting. For Microsoft stock, the model predicted with a relative error of 6.42% (RMSE / Price Movement) \* (100). reflecting the stock's volatility. Given these optimal error scores, the BI-LSTM model will be trained using these parameters.

Figure 10

A value of 4 for the step parameter means that the model is using a window of 4 days to predict future prices. That’s for every prediction, it takes the last 4 days price and basis the prediction off this. Although this is very short term, the idea here is that the model will be able to accurately predict short time price movements, whilst being trained and validated on the most recent 100 days of data for the selected stock. Typically, this 100-day window will likely capture earning reports (from last quarter), market-wide events and general trends and this forms the basis of choosing a step of 4 to make predictions. Each experiment also plotted a graph to visualize the accuracy of predictions. These parameters had the best error scores and the predicted close price crossed over with the test close price on 5 different occasions, figure 11.

A graph with blue and orange lines

Description automatically generated

Figure 11

It’s important to note that due to the nature of trading, especially due to human psychology, fear of missing out and speculation (people buy stocks when they think it will go higher due to demand). Stocks can pump (shoot up in price) or dump at very short notice according to the latest news. Inspired by the sudden increase in price for Microsoft stock and then subsequent downwards movement, research found that MSFT’s earnings in July 2023 for Q4 were $700 million above estimates. This could perhaps explain the price movement and further support these parameters. The 30 and 60-day windows could have had worse RMSE scores because they did not fully account for this price movement. A step of 4, a batch size of 4 and a 100-day training window enables the model to spot price trends and movements accurately and be computed quicker.

Table I: One Week Time Interval – Stock Close Price Column

| Stock / Currency | BI-LSTM Model Error Score | | |
| --- | --- | --- | --- |
| RMSE | MAPE | R2 |
| Microsoft | 3.5371 | 0.0095 | 0.1367 |
| Apple | 3.7345 | 0.0142 | -2.1543 |
| Tesla | 15.3856 | 0.0523 | -0.6186 |
| Nike | 3.2338 | 0.0239 | -0.3468 |
| BTC-USD | 2293.5183 | 0.0649 | -1.0861 |
| Visa | 2.5508 | 0.0094 | -5.1341 |

Table II: Two Week Time Interval – Stock Close Price Column

| Stock / Currency | BI-LSTM Model Error Score | | |
| --- | --- | --- | --- |
| RMSE | MAPE | R2 |
| Microsoft | 5.5769 | 0.0138 | 0.2945 |
| Apple | 5.0600 | 0.0215 | 0.6274 |
| Tesla | 22.2702 | 0.0730 | -1.0828 |
| Nike | 2.8613 | 0.0200 | -0.3702 |
| BTC-USD | 1857.2198 | 0.0444 | -0.6147 |
| Visa | 2.4500 | 0.0088 | -1.1142 |

The results of using the BI-LSTM model on the selected stocks and currencies indicates that when tested on the last week and two weeks of data (whilst being trained on the last 100), it performed best on Visa and Nike, Microsoft and Apple stocks. This is in comparison to the Tesla stock and Bitcoin currency, this assessment is based on the model error scores, exhibiting closer predictions through RMSE/MAPE and explaining the variance in the data with the r2 scores. The model performed similarly on Microsoft as it did on the experiments suggesting that the reproducibility of the code is adequate. There was a slight variation in RMSE and MAPE scores, (3.53 and 3.21 RMSE, and 0.0084 and 0.0095 MAPE) these variations could be due to the different data being used across the days the experiments code was ran and the application itself. In the cases of Nike and Visa, the prediction accuracy benefitted from a larger time interval, whilst in the others, the model produced worse error scores. The plotting of the test and predicted values for Visa demonstrates the best prediction the model produced, figure 12, (One Week) and figure 13 (Two Weeks).

A graph with a red line

Description automatically generated

Figure 12

A graph with red and blue lines

Description automatically generated

Figure 13

The models poor performance on Bitcoin can be attributed to its high volatility and complex price movement. The timeframe of testing coincided with the largest bout of volatility since November 2021, as per Bitcoin Magazine (2023). The model’s performance on Tesla when compared stocks like Visa can also be attributed to it’s high volatility in comparison, Visa moved at its largest range, 5 points across the time period of two weeks, whilst Tesla moved from $267 to $215 at it’s greatest range. This is something to note for future work.

Table III: One Week Time Interval – Stock Close Price Column

| Stock / Currency | Unexperimented XGBoost Model Error Score | | |
| --- | --- | --- | --- |
| RMSE | MAPE | R2 |
| Microsoft | 10.6750 | 0.0241 | 0.0364 |
| Apple | 6.6614 | 0.0329 | -20.3234 |
| Tesla | 36.0742 | 0.1349 | -12.3862 |
| Nike | 3.5457 | 0.0262 | -0.8691 |
| BTC-USD | 1104.9156 | 0.0252 | 0.1563 |
| Visa | 8.4439 | 0.0336 | -14.5449 |

When compared to an unexperimented XGboost model, our BI-LSTM model outperformed in terms of prediction accuracy on all stocks except Bitcoin. The XGBoost model was able to better capture the variance between the data and learn from its errors in a more coherent manner than the BI-LSTM model resulting in a better RMSE and MAPE.

# Conclusion

The project aimed to experiment on parameters, train a BI-LSTM model using these, utilise the model to make accurate stock price predictions and evaluate these predictions using typical error metrics and to assess its performance on unseen stocks, both related to the initial stock Microsoft, and otherwise. This was successfully encompassed within a user-friendly application which facilitates the use of machine learning models for people with little to no experience with machine learning. The findings of the study suggest that the BI-LSTM model could predict relatively accurately within a one-week time period, and in some cases improved when predicting a two-week time period with smaller relative mean squared errors and being able to explain the variants in the predictions (better r2 scores). The model performed better on stocks with less volatility and complexity, suggesting future work with the model should experiment with different parameters and weightings to capture the complexity of stocks and currencies such as Tesla and Bitcoin.

The project filled a void in the market, it introduces an interactive web application that brings together the power of machine learning through predictions as well as the traditional financial indicators commonly used in day-to-day trading. The integration of sentiment analysis sought to provide even more information for user’s to make an informed financial decision however, given the similarity between sentiment scores, the chosen approach does not seem suitable and could certainly benefit from further development. Although the project demonstrated an ability to predict certain stocks with good accuracy, financial markets are inundated with noise and require constant attention. The project shows promise but it's important that future work works on consistent refinement, central to this will be extensive user testing as well as more experiments with a vast array of features.

# Future Work

Considering the results and the promise of the model performing better on the two-week timeframe for some stocks according to error scores, future work could test the model’s performance across larger time frames.

The use of the NewsApi for sentiment analysis did not seem sufficient, thus future work could make use of multiple data sources such as the twitter Api to learn from stronger opinions and assess sentiment through a richer context about stocks and currencies. Moreover, future work could integrate sentiment analysis as a feature for the BI-LSTM model to also take into consideration when making predictions.

Whilst the BI-LSTM and XGBoost models function effectively, there is a potential to expand the application by incorporating more models and showcasing error scores. This would enable easier model comparisons and increase flexibility of the application. A future version could further offer preset and 'advanced' configurations, achieved by modifying our mlflow experiments code, transforming this application into a comprehensive stock prediction tool and machine learning configuration tool.

Furthermore, integrating technical indicators, commonly used by traders, could enhance the model's predictive accuracy. It would be interesting to see the impact the inclusion of technical indicators as a feature would have on the prediction accuracy of the model.

##### References

Abadi, M. et al., 2016. TensorFlow: A system for large-scale machine learning. *12th USENIX Symposium on Operating Systems Design and Implementation* (OSDI '16).

**Ando, R. and Takefuji, Y., 2021. A Randomized Hyperparameter Tuning of Adaptive Moment Estimation Optimizer of Binary Tree-Structured LSTM. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 12(7)**

Aslam N. F. Rustam, E. Lee, P. B. Washington, and I. Ashraf, "Sentiment Analysis and Emotion Detection on Cryptocurrency Related Tweets Using Ensemble LSTM-GRU Model," *IEEE Access,* vol. 10, 2022, pp. 39313-39324.

Aste, T. "Cryptocurrency market structure: connecting emotions and economics," Digit Finance, vol. 1, 2019, pp. 5–21.

Bollen, Johan & Mao, Huina & Zeng, Xiao-Jun. (2010). *Twitter Mood Predicts the Stock Market. Journal of Computational Science.*

Chollet, F., 2015. *Keras*. Available at: https://keras.io [Accessed: July 23 2023].

[CNBC (2023) 'Microsoft (MSFT) Q4 earnings report 2023', *CNBC*. Available at: https://www.cnbc.com/2023/07/25/microsoft-msft-q4-earnings-report-2023.html (Accessed: 05 August 2023).

LeClair, D. (2023, August 21). Bitcoin Volatility Returns With 15% Drop. Bitcoin Magazine Pro. URL: https://bitcoinmagazine.com/markets/bitcoin-volatility-returns-with-massive-bitcoin-price-drop.

Li, J. "A Comparative Study of LSTM Variants in Prediction for Tesla’s Stock Price," *BCP Business & Management,* vol. 34, 2022, pp. 30-38. [Online]. Available: 10.54691/bcpbm.v34i.2861.

McKinney, W., 2010. Data Structures for Statistical Computing in Python. *Proceedings of the 9th Python in Science Conference,* pp.51-56.

Nabipour, M. Nayyeri, H. Jabani, S. S., and A. Mosavi, "Predicting Stock Market Trends Using Machine Learning and Deep Learning Algorithms Via Continuous and Binary Data; a Comparative Analysis," *IEEE Access*, vol. 8, 2020, pp. 150199-150212.

Patil, P. C.-S. M. Wu, K. Potika, and M. Orang, "Stock Market Prediction Using Ensemble of Graph Theory, Machine Learning and Deep Learning Models," Proceedings of the 3rd International Conference on Software Engineering and Information Management (*ICSIM '20*), Association for Computing Machinery, New York, 2020, pp. 85–92.

Pedregosa, F. et al., 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, pp.2825-2830.

Pramudya, R. "Technical Analysis To Determine Buying And Selling Signal In Stock Trade," International Journal of Finance & Banking Studies (2147-4486), vol. 9, no. 1, 2020, pp. 58–67. [Online]. Available: https://doi.org/10.20525/ijfbs.v9i1.666

Ramani, K. M. Jahnavi, P. J. Reddy, P. Venkatachakravarthi, P. Meghanath, and S. K. Imran, "Prediction of Bitcoin Price through LSTM, ARIMA, XGBoost, Prophet and Sentiment Analysis on Dynamic Streaming Data," *2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS),* Coimbatore, India, 2023, pp. 1514-1518.

Stojanovic, R., 2019. *yfinance: Yahoo! Finance market data downloader*. Available at: <https://github.com/ranaroussi/yfinance>