Stock Forecasting with Sentiment Analysis and Deep Learning: A Big Data Management Framework

Adam Green
Department of Math
Statistics
University of Guelph
Guelph Canada

Pansy Ahmed
Department of Math
Statistics
University of Guelph
Guelph Canada

Abstract

Due to the complexity associated with application of financial time series forecasting, stock price prediction has received extensive academic attention in recent years. A wide variety of influential factors including inflation, monetary policy and social trends introduce risk to the process of investing in stocks. Past research in stock forecasting has focused on both probabilistic and deep learning models, with successes in the application of the Facebook Prophet model and long short-term memory (LSTM) models. Our research proposes a solution to predicting stock price movement by integrating sentiment analysis of news headlines pertaining to target companies with financial metrics to train a hybrid model combining both the Facebook Prophet model and an LSTM framework. This framework is accompanied by a focus on scalability to ease the process of data processing and future model training. Based on the proposed framework, forecasting models were trained to predict the following day close price of a target stock, given a look-back window of 10 trading days. Three final models were produced for evaluation: a base LSTM model, a hybrid model trained on a single stock and a hybrid model trained on multiple stocks. This research provides the basis for future work in developing such hybrid forecasting models. The work aims to understand the benefit of a hybrid model over a base LSTM model as well as the benefit derived from training the hybrid model on data of multiple stocks vs. data of a single stock. While the results did not indicate the hybrid model could provide a profitable trading strategy or improvements over base LSTM, further tuning could improve model accuracy to the point of profitability. The data management pipeline promotes deeper analysis of market sentiment using alternative news sources while improving the ability to train the model using data from an expanded company set.

1 Introduction

A range of macro- and micro-economic factors influence the stock market, such as monetary policy, political developments and institutional investor expectations[14]. However, the precise consequences of these elements are variable making stock price movement difficult to anticipate when performing investment forecasting. The high-level problem motivating this work is that of quantifying abstract patterns in stock market movement as we aim to reduce the high predictive error often intrinsic to stock forecasting models. We hope to produce results with high enough accuracy to motivate further development of our predictive model. The general question guiding this work is - *can an optimized model for time-series forecasting with prediction of seasonal trends accurately predict stock close prices when combined with market sentiment analysis?* From a big data perspective, the main problem addressed in this work is the scalability of sentiment analysis and data processing to reduce the time required to optimize trained models. We aim to understand the benefit to model training derived from introducing Spark and PostgreSQL as management tools in the data processing pipeline.

Our work has potential benefit in that it serves as a benchmark for future research in optimizing hybrid forecasting models. There is also further benefit derived from the potential to support real-world trading strategies. With further tuning, our model could be applied as a standalone indicator for daily buying and selling or be used as part of a compound indicator in trading. The potential pay-off of our work is significant, regardless of final model results, as the novelty of the model promotes further exploration into the role of sentiment analysis and Prophet forecasting in financial asset modelling.

2 Problem Definition and Outcomes

In this research, the problem we aim to solve is 3-fold. First, we hope to understand whether a single deep learning model (LSTM) for stock price forecasting can be improved through hybrid model methods, specifically, combination with Prophet and market sentiment analysis. The second key problem we hope to address is whether training a forecasting model on multiple stocks generates improvement in prediction accuracy beyond what is achieved by a model trained on a single stock. Finally, we aim to solve the issue of model scalability with respect to financial text processing for the purpose of sentiment analysis. Based on these problems, key questions were formulated to direct our research and model evaluation:

1. Does the use of a hybrid model that combines Prophet and LSTM outperform a base LSTM model?

2. Can accuracy and applicability of the hybrid model be improved through expanding the training dataset to include multiple companies?

Evaluation of the above questions will be conducted through analysis of model mean square error (MSE) and mean absolute error (MAE) in predicting the close price of the following day for a target stock. Our models will be evaluated based on internal comparisons of prediction accuracy on test data, rather than comparisons to the outcomes of existing literature. Our research does, however, fit well in the current state of the field as the specific questions outlined above have yet to be investigated in similar stock forecasting research[5, 13, 12, 6, 9]. Justification for our first proposed research question lies in the belief that LSTM models generally provide lagging predictions of price movement, often missing swings in stock price. Inclusion of Prophet forecasting in a hybrid model is believed to improve the potential for leading predictions. Further, our second research question was formulated with the belief that while the price of separate stocks are not correlated, the patterns of stock movement based on technical trading indicators and market sentiment are similar. By including multiple companies in the hybrid model, it is hoped that the model can more effectively learn these technical trading patterns.

3 Related Work

While our focus is not on comparisons of our accuracy results directly to those produced by other works due to differences, understanding how our research relates to existing projects is useful in identifying the motivation behind our problem and proposed solution. Significant research has been conducted on the application of statistical and neural network predictive techniques on stock market forecasting for a variety of global markets[7, 10, 3]. *Table 1* shows a sample of models used in other research and the basic outcomes discovered.

Model	Outcome
LSTM[3]	Combining features of one stock and features of Shanghai Securities Composite Index will increase performance
LSTM and ANN[7]	The LSTM has a high ability to distinguish between market fluctuations and accidental fluctuations.
LSTM and GRU[10]	Models will improve through analysis of financial news as a feature

Table 1: Summary of Models Used in Related Work and Their Outcomes

Chen et al. made use of an LSTM model to predict securities on China's stock exchange[3]. Features including opening price, closing price, daily high and daily low were used to train the model. This model gave a positive result, however, financial news sentiment was not considered in this study. Further, we found that Bahadur Shahi et al. discussed how to align the sentiment score to daily stock data[10]. We combined these methods in our implementation to reflect the successes of each.

In order to identify the most accurate models to include in our hybrid implementation, we found a range of related research. Auto-regressive integrated moving average models (ARIMA), general artificial neural networks (ANN), and LSTM have been explored to predict the stock price of Dell through 2010. Comparisons were carried out between ARIMA and ANN as well as ANN and LSTM. ANN did a better job taking into account the volatility of stock prices over a period of time when compared to ARIMA. LSTM was more effective than ANN because of its ability to distinguish between true market fluctuations and accidental fluctuations [7]. Given this result, we felt an LSTM model was an appropriate decision for our deep learning framework.

Further, Xiaochun et al. made use of stacked long-term and short-term models on an S&P 500 index dataset from January 2000 to May 2016. The results showed that the stacked LSTM model produced the most accurate prediction over a single model[16]. This finding indicates that hybrid models have viability in improving predictions over base models.

Mohammed Ali Alshara compared Prophet and LSTM models for stock price prediction and explains the potential for success achieved through use of the Prophet model[1]. However, LSTM produced better standalone results when compared to Prophet[1]. This research supported our model goals for a Prophet-LSTM hybrid model in stock forecasting to gain the benefits of each individual model.

Finally, past work has identified the potential benefit of sentiment analysis, seasonal forecasting and deep learning in stock forecasting. However, no work has combined all three methods to form stock price predictions. We aim to do this while answering the key research questions outlined above.

4 Methodology

To answer our questions of interest, the project was divided into five phases: Data collection, data pre-processing, model architecture, training and model evaluation. The logistics of these steps and tools used are outlined below.

4.1 Big Data Management Tools

While our research targets a solution for stock forecasting, a data management framework was created to handle scalability of our solution and promote further development of our models. The primary goals in the formation of the data pipeline were to allow integration of additional news/sentiment sources in addition to easing the process of training future hybrid models. These goals were met through use of Apache Spark for distributed data processing and a PostgreSQL database for storage of relational data. Apache Spark is an open source, distributed computing engine built to achieve efficient parallel computing with high fault tolerance[11]. For the purposes of this research, a Spark cluster was deployed including a master node with two worker nodes to run data processing steps in a scalable format. Additionally, PostgreSQL was selected as the primary database management system for this work as all necessary data was structured in nature and the data pipeline did not require high read capabilities or the need for horizontal scalability.

Regarding the data pipeline, retrieval of news headlines for sentiment analysis was the most computationally expensive step. This process was purposefully separated from model building, allowing the scraping and processing framework to operate in the background, preparing data without limiting the speed of model production. Data retrieval could be run at regular intervals to expand the company database and allow for training of future models. Additionally, the high computational need of this process was addressed using Spark for data processing, as previously stated. This enables integration of expanded datasets in the future with respect to sentiment analysis. Additional news sources or tweets could be included in the processing pipeline without issue. Our current models were built using a reduced historical look back period for easing the proof-of-concept process, however, future development would benefit from including expanded datasets from a longer historical period.

4.2 Data Collection

The primary data source for technical financial indicators was the yfinance Python library which generates historical datasets for a range of daily stock metrics[2]. This data underwent feature engineering to develop a number of additional metrics not provided by the yfinance package (outlined below). We compiled historical financial data from January 1st, 2016 to March 1st, 2023 for five target stocks (Tesla, Google, Microsoft, Netflix and Amazon). *Figure 1* shows the trend in closing stock price for each of the five companies over the indicated date range. All gathered financial data was stored in the project database for improved future access.

Regarding data used for sentiment analysis, historical news headlines pertaining to each of the five target stocks were retrieved from the Financial

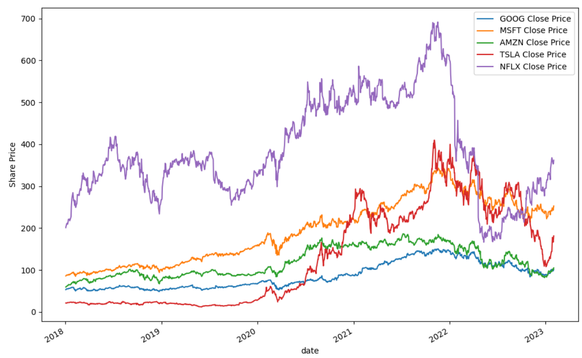

Figure 1: Change in Closing Stock Price Over Time for Target Companies

Post using the Beautiful Soup Python package[8]. Headlines were compiled in the database over the date range of January 1st, 2013 to March 1st, 2023, highlighting the success of our data processing framework. These headlines served as the basis for market sentiment analysis in this work, while our framework also allows for future development to integrate news data from multiple sources.

4.3 Data Pre-processing

To prepare data for modelling, pre-processing was completed with respect to both the financial dataset and news headlines data. Regarding financial data, several features were engineered to reflect key indicators commonly used in stock trading. These engineered features included 10-, 20-, 50- and 100-day moving averages as well as exponential moving averages, Bollinger Bands and relative strength index values (RSI) which represent common technical indicators used commonly by human traders to discover patterns in stock movement. RSI is an oscillating value that represents price momentum of a stock. High RSI values indicate the stock is overbought while low values indicate the stock is oversold.

Regarding news headline data, analysis of headline sentiment was completed by passing headlines to the Valence Aware Dictionary and Sentiment Reasoner (VADER). The outcome of this process was a label for each headline comprised of a compound score ranging from -1 to 1. Sentiment scores were averaged by day with days having no headlines being given a default score of 0. These average daily values were then multiplied by the square number of headlines gathered for the day and scaled back to a value between 0 and 1. This process was completed to ensure sentiment scores based off a single headline had a lower weight in the model compared to scores based off many headlines. After processing of sentiment was complete using Spark, final daily sentiment scores for target companies were stored in the project database.

After processing and storing all data, subsequent use in the modelling process involved reading target data from the database and merging financial metrics with sentiment scores on the date index to form the final data-sets used in model production.

4.4 The LSTM Model

The proposed hybrid model encompasses two forecasting implementations combined to achieve an increased level of rigour. The first was an LSTM model based on existing literature, and created using the TensorFlow Keras package for Python[5, 13, 12, 6, 9]. This model was used to make predictions of stock close price at time *t+1* given a look-back period from *t-9 to t*. The underlying structure of the LSTM model is broken down in *Figure 2* to represent updating the model at a single time point[15]. *ct* represents the current state of the cell with each time point taking the previous cell state (*ct-1*) the previous hidden state (*ht-1*) and the current data update (*xt*) as inputs[15]. The cell state is updated based on the activity of the input, output and forget gates which determine what information is stored in the model's memory at each time point and how the cell states are updated[15]. The model output at each time point is the cell state (model weights) as well as the hidden state which represents a filtered version of the cell state intended to retain important information from previous time points. The overall output of this model is daily predictions for the following day's closing stock price. For the purposes of this research, a random search function was implemented to determine model structure parameters. The details of the final model are outlined below.

4.5 The Prophet Model

In addition to the LSTM model, Prophet was trained on daily stock close prices of Microsoft. This model is intuitive in nature with auto-feature engineering to model seasonal trends. The general formula of the Prophet model is given below.

$$y(t) = g(t) + h(t) + s(t) + e_t$$

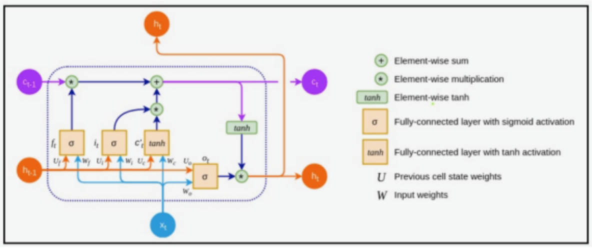

Figure 2: LSTM Framework

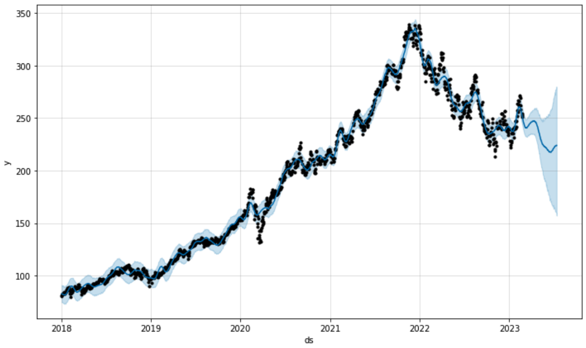

Figure 3: Prophet predicted stock price and corresponding CI based on the stock price of Microsoft

$g(t)$ represents the trend forecast, $s(t)$ represents periodic trends for weekly and yearly change, $h(t)$ gives the effect of holidays and e_t is an error term[4].

While the model is intuitive, we used cross validation and mean squared error to determine the best hyper-parameters. When the parameter 'change point prior scale' is set to 1.0, the model assumes a moderate level of flexibility, striking a balance between sensitivity and stability. When the seasonality mode is set to multiplicative, the model assumes that the seasonal component of the time series is proportional to the overall trend. When setting 'seasonality prior scale' to 0.01, the model assumes a weak seasonality prior, expecting the seasonal component to vary only slightly across the time series. By carefully tuning these parameters, the prophet model can provide accurate and reliable forecasts for a wide range of time-series data. *Figure 3* shows the prediction of the next 100 periods, and for our work, the final prophet model only makes predictions for weekdays as trading does not occur on weekends.

Due to the nature of Prophet predictions, to produce consistently accurate forecasting, a training loop was created to continuously train a Prophet model, predict price change for the following ten days and then use the true price values for those ten days to retrain a new model for the next ten-day prediction. This iterative process was completed for the entirety of the target date range.

4.6 Hybrid Architecture

To combine the models outlined above, outright price predictions given by Prophet and the LSTM model(s) on Microsoft training data were compiled into a single data-set. This dataset was used to train a final LSTM model with the initial Microsoft data labels used as target data. The outcome was daily predicted close price for the following trading day. The output of this model will be in the same form as that of the initial LSTM network, allowing 1-to-1 comparison's of model accuracy between the hybrid models and the base LSTM model. It should be noted that using Microsoft as training data for the LSTM while also using predictions on the Microsoft data to train the hybrid model could result in over-fitting the model on this dataset. While we are aware of this potential fault, it is believed that this could improve the model's predictive capabilities on future Microsoft price movement. Future work could explore the change in predictive power and generalizability achieved by training the hybrid model on price predictions from alternative stocks.

4.7 Model Training

To account for the previously outlined research objectives, the model training process took place in two steps. First, a single LSTM model was trained on the financial data and news sentiment scores for Microsoft. This training process was implemented with a random parameter search function to determine ideal hyper-parameters for the LSTM network. The parameters identified during this process were used throughout other experiments for subsequent LSTM models. The final architecture consisted of a single LSTM layer with 98 neurons and a sigmoid activation function, as well as a single fully-connected dense layer. Training of the LSTM was optimized to minimize mean square error of predictions.

After training, the model was evaluated on a held out validation dataset pertaining to Microsoft as well as a Google dataset. Further, the LSTM model was used to form final predictions on the daily close price of original Microsoft training data. These predictions were appended to a secondary dataset to be implemented in hybrid model training. Next, Prophet was trained on the financial data for Microsoft and price forecasts made by the model were appended to the secondary dataset. This secondary dataset was then used to train the hybrid LSTM model.

The second portion of model training pertained mainly to the LSTM framework. Four LSTM models were trained on financial information and daily sentiment scores for four target stocks (MSFT, AMZN, NFLX, TSLA). After training the models, predictions from these models were combined with predictions from a trained Prophet model in the same manner outlined above. The resulting dataset contained predictions on Microsoft training data from each of the four LSTM sub-models and the Prophet model. This data was used to train a final expanded hybrid model. This model was again evaluated on Microsoft and Google test data before results were compared to those obtained from the previously noted models.

5 Results

After all three primary models were fit on appropriate training data, testing of predictive accuracy was finalized using Microsoft validation data from the held-out 10% split as well as a second evaluation data-set pertaining to Google. The primary metric used for evaluation was mean absolute error, while models were also evaluated on mean square error. The formulas used to calculate these values are provided:

Mean Absolute Error: $\sum_{i=1}^{D} |x_i - y_i|$

Mean Square Error: $\sum_{i=1}^{D} (x_i - y_i)^2$

These values were calculated using scaled predictions to allow direct comparisons between accuracy on Microsoft validation data and Google test data. The scores of each model using predictions on the Google test data are provided in *Table 2*. The base LSTM model achieved the best scores while the expanded hybrid model performed the poorest on the Google data.

6 Discussion and Evaluation

Based on the results outlined above, evaluation of our key research questions was completed to understand the potential benefit of a hybrid model over a base LSTM model as well the benefit derived from an expanded company set in training the hybrid model. First, in comparing the base LSTM model vs. the single-company hybrid model, the LSTM model achieved greater accuracy with respect to both primary evaluation metrics. *Figure 4* provides a representation of predictions given by the base LSTM model as well as the single-company hybrid model. It is clear that base model predictions follow true target values more closely that hybrid model predictions. Despite this, it would be of value to review the ability of the hybrid model to predict stock price change in a profitable way. Due to the nature of the stock market, the ability of the hybrid model to reflect trends in price change, even if exact values are incorrect could prove effective. The hybrid model prediction trend appears more regularized which could explain the poor scores observed when comparing evaluation metrics. Further testing should be conducted, however, given the existing model,it can be concluded that the hybrid model was unsuccessful in providing price change predictions that were more accurate than the base LSTM model.

Model	Prediction Accuracy (Google)
LSTM	Mean Square Error: 0.0006 Mean Absolute Error: 0.017
Single Company Hybrid	Mean Square Error: 0.003 Mean Absolute Error: 0.050
Expanded Hybrid Model	Mean Square Error: 0.035 Mean Absolute Error: 0.162

Table 2: Summary of Model Accuracy Scores on the Google Test Dataset

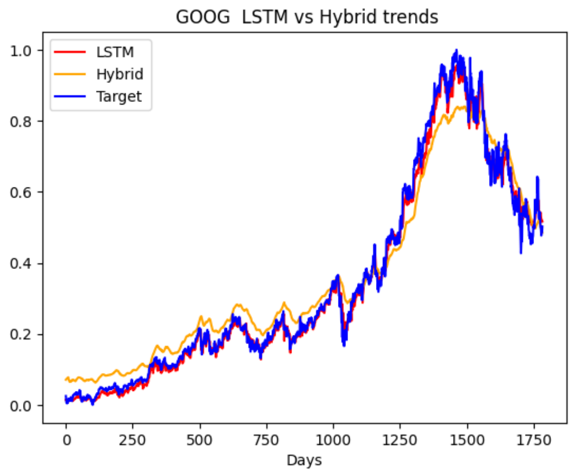

Figure 4: Google Stock Price Predictions Using the Base LSTM Model and Single Company Hybrid Model January, 2016 to March, 2023

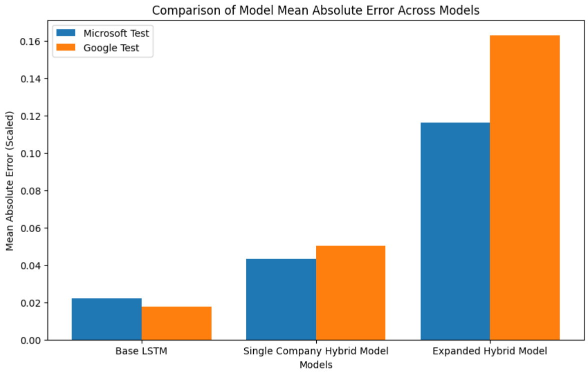

Figure 5: Comparison of Model Accuracy in Predicting on Microsoft and Google Test Data (MAE)

Regarding the second key question of interest, mean absolute error results depicted in *figure 5* indicate that expanding the hybrid model to include a wider range of stocks failed to improve predictive accuracy. The expanded model performed poorly on both the Microsoft validation data as well as the Google test data. This finding allows us to conclude that increased generalization could not be attained by expanding the set of companies used in the model training process. Given the outcomes of this research, it is likely that the high complexity of our framework had the opposite of the intended impact on model accuracy. Future work would likely benefit from reducing the number of features initially used to fit the LSTM model in addition to including Prophet data in the initial training process to eliminate the need for a second LSTM to combine LSTM and Prophet predictions. Despite the poor results of our hybrid models, we feel there is merit in exploring these models further to reduce complexity and fine tune various model aspects. These future directions are discussed below.

7 Model Application

Beyond primary research objectives in understanding model viability and implications of training procedures, we also focused on future application of the model to real-world trading. To address application of the model, a python file, operable from the command line, was created to provide outright stock price predictions of any user-defined stock using the pre-trained model. This application file remains very basic in function but provides context for a possible use-case for the model. Additional functionality could be built into the application to include selection of specific models and possibly the ability to train custom models using specified target stocks. Additional model development and testing is required before such an application can be successfully applied to real-world training, however, proof-of-concept was achieved.

8 Limitations and Challenges

The first challenge we encountered arose during data collection and pre-processing. Initially, our workflow aimed to use tweet data as a basis by which to evaluate daily market sentiment. While we were successful in cultivating such data, subsequent sentiment analysis efforts were ineffective as the data-set consisted of millions of tweets per target company with many containing irrelevant or inaccurate information. As a result, we altered our workflow to instead use financial news headlines gathered from the Financial Post [8]. While this decision solved the challenge of irrelevant text data, it also introduced a smaller issue as certain days did not have any news articles related to target stocks. For the purposes of this work, these sentiment gaps were filled with a default value representing neutral sentiment, however, future development could benefit from integrating alternative sentiment data sources to fill data gaps, made possible by our scalable processing framework. Tweet data could be used as one of these supplementary sources, possibly by applying a language model to analyze tweets for relevant information and filter unnecessary results that initially caused issues. Related to this was the specific limitation that headlines for sentiment analysis were retrieved from a single news source. Using a single source could result in sentiment inaccuracies due to potential media bias as well as missing data. This limitation could be overcome through integration of multiple sources in the same form outlined above.

The second limitation of this research related to the predictive power of the Prophet model. The model is designed to be trained and subsequently make a prediction for an extended future time period. The average error in this model increases as the prediction period extends. To minimize error while exploiting the predictive power of Prophet, our implementation requires the Prophet model to repeatedly make 10 day predictions of stock price change. These repeated 10 day forecasts required retraining the model on all available data after each 10 day period. This process ensured the Prophet model only ever predicted price change for a shortened time period and maximized the accuracy of our predictions. While this strategy improved the overall accuracy of the model, it also resulted in increased model training time and computational need. This trade-off was difficult to avoid, resulting in accuracy being prioritized over training speed. Even at its slowest, the training process for Prophet could be completed in real-time.

Finally, our current data-processing framework has limitations that could be addressed in future work. Technological issues in access to cloud-based data processing and storage services resulted in all services being locally-hosted for the duration of the project. Future work could expand our framework using a cloud service to allow widespread database access and increased processing power through creation of Spark clusters with more worker nodes. The technological challenges experienced led to the need for intermediary steps in data pre-processing that could be avoided through use of cloud-based services.

9 Next Steps

Based on the findings of this work, a number of next steps have been identified to guide future development of the proposed models. First, regarding model architecture and tuning, future work should evaluate model success when using fewer financial metrics in the training set. Currently, a wide range of features were engineered to reflect human-trading strategies, however, many of these features are highly correlated. Improved model accuracy may be achieved by reducing the feature set to include only moving price averages and basic daily statistics provided by the yfinance package.

Additionally, research should target extending the size of the training data-sets and extending the period over which predictions are made. Increased training data (longer historical look-back per company) is likely to result in improved model accuracy as the current training data-set is relatively small. Further, extending the prediction period beyond just the following day could provide new application directions to translate model predictions into a long-term investing indicator as opposed to a daily trading indicator. Coupled with extending the the prediction period is extending the look-back period of the LSTM model. The current model uses a 10 day look-back period to form predictions, however, extending this look-back period similar to research by Thormann et al. could impact model accuracy[15]. Although the exact outcomes of such changes are unclear with respect to the current model, future work could work to quantify the change in predictive power associated with look-back extension.

The final model alteration that should be explored in further work is the integration of Prophet data into the model. In the current implementation, Prophet data is appended to LSTM predictions to be passed to the hybrid model. Alternatively, Prophet predictions could be included in the dataset used to train the base LSTM model. Results indicate the proposed base LSTM models often provide lagging predictions, missing large swings in stock price. Prophet data included in the base model could rectify this issue to provide stronger leading predictions. Overall, the identified model alterations represent possible directions for further model development. While these possibilities do not guarantee improvements in predictive accuracy, there is potential for the changes to positively impact model applicability.

Beyond changes to the current model formats, the results of this research indicate that the Prophet model may not be ideal in this setting. future work could explore the impact of combining an alternative probabilistic model with LSTM such as the auto regressive integrated moving average model. Based on existing literature, it is possible that such a model could perform better than Prophet[7]. It may also be of value to explore an LSTM model trained on a wide range of target stocks without the inclusion of a secondary probabilistic model to better understand the faults of Prophet in this work.

Finally, future development of this research following appropriate model tuning should focus on further development of the proposed application. The current application file remains in a "bare-bones" state, providing only functionality to predict the next day's closing stock price. While this is successful in proving practicality of the forecasting solution, it is very limiting to users. Future development should focus on allowing customized training of models. Currently, only pre-trained models can be used in the application setting. Allowing users to train models on specific companies of interest could improve model accuracy and applicability. Additionally, such development could provide functionality for selecting varied prediction and look-back periods to meet user needs.

10 Conclusion

Overall, this work provides a basis for future research in the use of hybrid modelling and sentiment analysis in stock forecasting. While overall results of prediction accuracy were not adequate to conclude viability of our model in practice, future model optimization is possible and could help produce a solution with more accurate predictions. We were successfully able to answer our questions of interest, with findings indicating that the hybrid LSTM-Prophet model was unable to improve upon predictions made by the base LSTM model. Further, contrary to our hypothesized outcomes, the expanded model was unable to improve upon the single-company hybrid model in predicting price changes of an unseen test stock. Overall, the base LSTM model achieved the best accuracy scores on both the Microsoft validation data and Google test data. Despite these findings, our big data management framework remains a strength in our work and can help simplify the process of further model exploration based on the suggested next steps outlined above. Tuning and changes to the architecture of the existing model, extended analysis of market sentiment, and significant changes in the supplemental forecasting model used are likely to achieve improve results with the support of our data managment framework.

References

[1] Mohammed A. Alshara. Stock forecasting using prophet vs. lstm model applying time-series prediction. In *International Journal of Computer Science and Network Security*, volume 22, pages 185–192, 2022.

[2] R Aroussi. yfinance 0.2.12, 2023.

[3] Chen K. Zhou Y. Dai F. A lstm-based method for stock price returns prediction: A case study of china stock market. In *2015 IEEE International Conference on Big Data, CA, USA*, pages 2823–2824, 2015.

[4] R Kumar. Predicting stock prices using facebook's prophet model, 2021.

[5] Xinyi Li, Yinchuan Li, Hongyang Yang, Liuqing Yang, and Xiao-Yang Liu. Dp-lstm: Differential privacy-inspired lstm for stock prediction using financial news. *arXiv preprint arXiv:1912.10806*, 2019.

[6] Wenjie Lu, Jiazheng Li, Yifan Li, Aijun Sun, and Jingyang Wang. A cnn-lstm-based model to forecast stock prices. *Complexity*, 2020:1–10, 2020.

[7] Qihang Ma. Comparison of arima, ann and lstm for stock price prediction. 218:01026, 2020.

[8] The Financial Post. News archives, 2023.

[9] Jiayu Qiu, Bin Wang, and Changjun Zhou. Forecasting stock prices with long-short term memory neural network based on attention mechanism. *PloS one*, 15(1):e0227222, 2020.

[10] Tej Bahadur Shahi, Ashish Shrestha, Arjun Neupane, and William Guo. Stock price forecasting with deep learning: A comparative study. *Mathematics*, 8(9):1441, 2020.

[11] Apache Spark. What is apache spark?, 2023.

[12] Md Arif Istiake Sunny, Mirza Mohd Shahriar Maswood, and Abdullah G Alharbi. Deep learning-based stock price prediction using lstm and bi-directional lstm model. In *2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES)*, pages 87–92. IEEE, 2020.

[13] T Swathi, N Kasiviswanath, and A Ananda Rao. An optimal deep learning-based lstm for stock price prediction using twitter sentiment analysis. *Applied Intelligence*, 52(12):13675–13688, 2022.

[14] Yajiao Tang, Zhenyu Song, Yulin Zhu, Huaiyu Yuan, Maozhang Hou, Junkai Ji, Cheng Tang, and Jianqiang Li. A survey on machine learning models for financial time series forecasting. *Neurocomputing*, 512:363–380, 2022.

[15] Marah-Lisanne Thormann, Jan Farchmin, Christoph Weisser, René-Marcel Kruse, Benjamin Säfken, and Alexander Silbersdorff. Stock price predictions with lstm neural networks and twitter sentiment. *Statistics, Optimization & Information Computing*, 9(2):268–287, 2021.

[16] Xiaochun Zhang, Chen Li, Kuan-Lin Chen, Dimitrios Chrysostomou, and Hongji Yang. Stock prediction with stacked-lstm neural networks. In *2021 IEEE 21st International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, pages 1119–1125, 2021.