Stock Forecasting with Sentiment Analysis and Deep Learning

October 15, 2023

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, changing monetary policies 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 Meta Prophet model and long short-term memory (LSTM) models.

The many drivers of stock demand and supply are inherently affected by public opinions directed by news headlines , customer satisfaction and word of mouth creating the level of brand trust and demand. Those sentiments contribute to the performance of stocks but are difficult to account for precisely even with the most experienced stock traders' estimations.

Our research proposes a solution to account for the sentiment factor by predicting stock price movement for the next day based on the integration of sentiment analysis of news headlines and a look-back of price movements for the past-respective 10 days. The data collected pertains to the stocks of five companies with financial metrics including 18 features to train a hybrid model combining both the Meta Prophet model processed time series predictions and the LSTM model processed sentiment scores and financial data. 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.

Our stock selection process was based on finding similar stocks of comparable trading volumes and presence on the stock market and financial headlines. The data was scraped from the internet over the span of 5 years (2018-2023). The final stock list was concluded to include (Microsoft - MSFT, Tesla - TSLA, Google-GOOG, Amazon- AMZN, and Netflix- NFLX) . In general, we aim to quantify the benefit of training our models on a single stock (MSFT) vs. one or multiple stocks (TSLA, GOOG, AMZN, NFLX).

Five final models were produced for evaluation: a base LSTM model, a hybrid model trained on a single stock using LSTM, a hybrid model trained on a single stock using Linear Regression, a hybrid model trained on multiple stocks using LSTM and a hybrid model trained on multiple stocks using Linear Regression . The goal is to measure the capability of the models to generalize and predict unseen stocks for future implementations. This research provides the basis for future work in developing such hybrid forecasting models. The work aims to understand the efficiency 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 were promising for the hybrid models over the base LSTM model, the linear hybrid model showed an overall better performance than its hybrid-LSTM counterpart. LSTM base model showed a comparable performance only on pre-trained MSFT data, deeming it's generalization capabilities as inadequate. However the linear hybrid model performance can be tuned to improve model accuracy to the point of profitability.

1 Introduction

A variety of macroeconomic and micro-economic factors influence the stock market, such as monetary policy, political developments and institutional investor expectations[13] However, the precise consequences of these elements are variable making stock price change difficult to anticipate when aiming to 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 accuracy to motivate further development of our predictive model. The general question guiding this work is whether an optimized model for time-series forecasting with prediction of seasonal trends accurately classifies daily stock price change when combined with market sentiment analysis.

From a technical perspective, the goal of this project is to investigate the predictive power of a hybrid model combining Prophet, a forecasting model created by the Facebook data science team[4] and a long short-term memory model, in comparison to a general LSTM model, in predicting stock price change. In this goal, we will evaluate the accuracy of the hybrid model in comparison to a single LSTM model, in addition to the change in accuracy achieved when models are trained on a single stock versus multiple stocks.

As stated previously, our work has potential benefit in that it serves as a benchmark for future research in optimizing hybrid forecasting models. There is further benefit as the model could support real-world trading strategies. Depending on the predictive success, our model could either 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 for our work is largely reliant on the precision score we achieve when evaluating the success of our model.

2 Problem Definition and Outcomes

In this research, the problem we aim to solve is 2-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 forecasting.

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 root mean square error (RMSE) 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 error 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, 12, 11, 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. The 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[15]. 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 preprocessing, model architecture, training and model evaluation. The logistics of these steps and tools used are outlined below.

Figure 1: Methodology Steps

4.1 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, 2018 to February 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. Initial features included open price ,high price,low price ,close price ,adjusted close price ,volume traded, in addition to other stock indicators like ,RSI, Middle Band, Upper Band, and Lower Band thresholds.

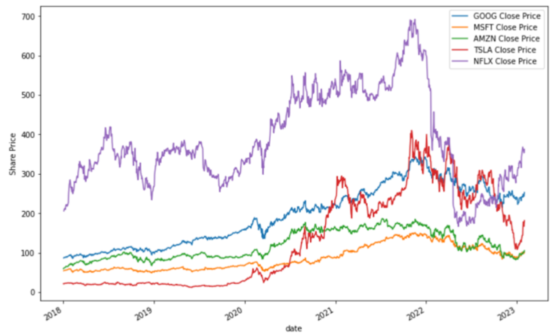

Figure 2: Change in Closing Stock Price Over Time for Target Companies

Regarding data used for sentiment analysis, historical news headlines pertaining to each of the five target stocks were retrieved from the Financial Post using the Beautiful Soup Python package[8]. Headlines were compiled over the date range of January 1st, 2018 to February 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.

4.2 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. The formulas for this technical indicators are reflected in table 2.

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

Features	Formulas
10-, 20-, 50-, 100-Moving Day Average	(A1 + A2 + ...+An) / n
Exponential Moving Average	Price Today * (Smoothing / (1 + Days)) + EMA Yesterday * (1 – (Smoothing / (1 + Days))
Relative Strength Index	100 - (100 / (1 +RS))
Bollinger Bands	20-period SMA (Simple Moving Average)

Table 2: Formulas of Technical Indicators

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. Figure 3 is a diagrammatic representation of this. After the processing of sentiment was complete, the financial metrics were merged with sentiment scores on the date index to form the final data sets used in model production.

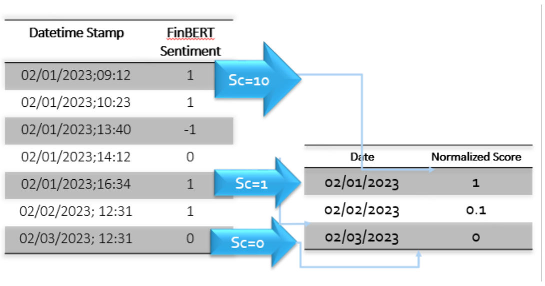

Figure 3: Sentiment Pre-processing

4.3 Model Building

This section describes the modelling process throughout the data gathering and pre-processing then passing the data through the models in parallel, then creating a combined view of the results from both to enhance the final price predictions of the stocks. Figure 6 depicts a high level architecture of our workflow.

4.3.1 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, 12, 11, 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 4* to represent updating the model at a single time point[14]. *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[14]. 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[14].

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.

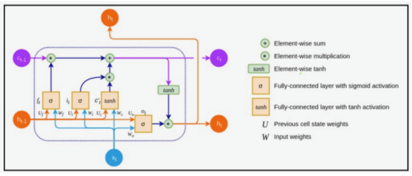

Figure 4: LSTM Framework

Base LSTM Single Model Building

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 hyperparameters 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.

4.3.2 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.

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

$g(t)$ 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.

Prophet model Building
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.

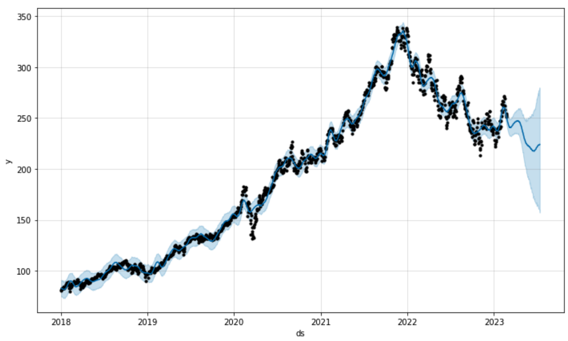

Figure 5: Prophet predicted stock price and corresponding CI based on the stock price of Microsoft

4.3.3 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 data-set. While we are

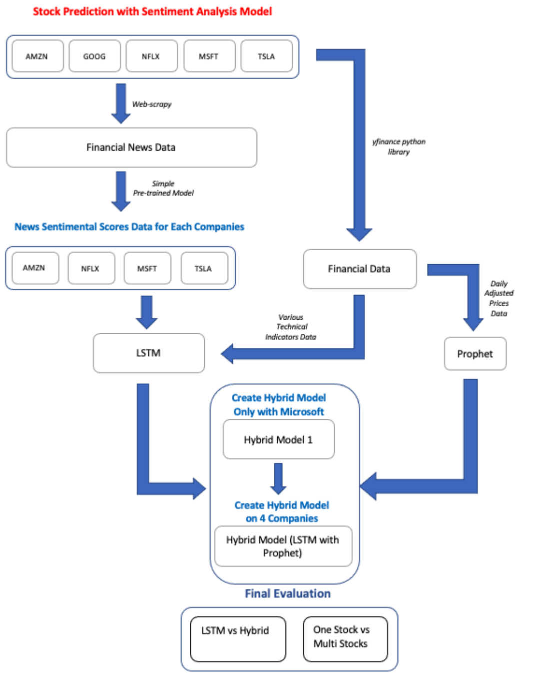

Figure 6: Data workflow through different models

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 generalization capability achieved by training the hybrid model on price predictions from alternative stocks.

To account for the previously outlined research objectives, the model training process is broken down as depicted in the figure below.

LSTM Hybrid Model
In order to achieve the hybrid model, the results from the Prophet model was combined with the results from the LSTM model to create a secondary data-set to be passed to a Hybrid-processing LSTM model for each of Microsoft, Google, and the expanded models.

LSTM Multi Model
This 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 testing both LSTM and linear regression. This model was again evaluated on Microsoft and Google test data before results were compared to those obtained from the previously noted models.

Linear Regression Hybrid Model

Based on the performance achieved by measuring the mean square error and mean absolute error before and after passing to the hybrid model, an alternative approach was decided to use linear regression as the hybrid-processing model and measure if the outcome was better.

5 Results

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

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

Root Mean Square Error: $\sqrt{\frac{1}{n}\Sigma_{i=1}^{n}\left(\frac{d_i - f_i}{\sigma_i}\right)^2}$

5.1 Single Company Hybrid Model

Regarding the single company hybrid models, the base LSTM performed best on Microsoft validation data with a RMSE of 7.86 while the linear hybrid model and LSTM hybrid models performed poorly with RMSE values of 9.10 and 12.23 respectively. Comparatively, on Google test data, the linear hybrid model performed best. The price predictions given by the linear hybrid model on Google test data are depicted in *Figure 7* in comparison to true price values.

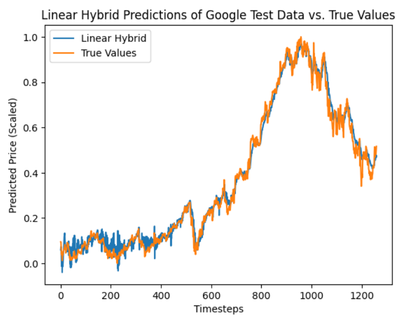

Figure 7: Linear Hybrid Model Predicted Google Test Values vs True Values

5.2 Multi-Company Hybrid Model

Regarding results of the multi-company models, both the LSTM and linear hybrid models achieved improvements in predictive accuracy over the base LSTM model. The expanded linear model was most accurate on Microsoft validation data while the expanded LSTM model achieved the lowest RMSE score on Google test data. A comparison of price predictions given by the expanded LSTM hybrid model and true values for Google test data is provided in *Figure 8*

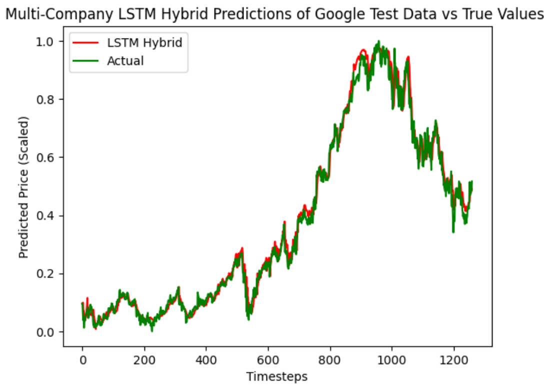

Figure 8: Google Stock Price Predictions Using the Base LSTM Model and Multi-Company LSTM Hybrid Model January 2018 to February 2023

5.3 Overall Comparison

The RMSE values of each model, evaluated on Google test data are provided in *Table 3*. The multi-company linear hybrid model performed best on Google test data while the multi-company LSTM hybrid model was the most accurate on Microsoft validation data. *Figure 9* also provides a visual comparison of results, indicating success in improving upon the base LSTM model through hybrid methods.

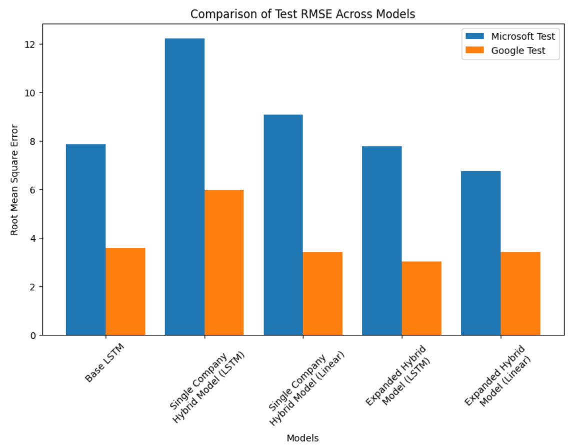

Figure 9: Comparison of RMSE Across Models on Test Data

Model	Prediction Accuracy (Google)
LSTM	Root Mean Square Error: 3.57 Mean Absolute Error: 2.80
Single Company Hybrid(LSTM)	Root Mean Square Error: 5.98 Mean Absolute Error: 5.12
Single Company Hybrid(Linear Regression)	Root Mean Square Error: 3.42 Mean Absolute Error: 2.57
Expanded Hybrid Model(LSTM)	Root Mean Square Error: 3.04 Mean Absolute Error: 2.19
Expanded Hybrid Model(Linear Regression)	Root Mean Square Error: 3.42 Mean Absolute Error: 2.82

Table 3: Summary of Model Accuracy Scores on the Google Test Dataset

6 Discussion and Evaluation

Based on the results outlined above, evaluation of our key research questions were 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 linear hybrid model successfully improved upon the base LSTM model in predicting closing stock prices for a stock unseen during training. This finding is important as it indicates that the hybrid model presented in this model is viable, motivating further testing and evaluation.

While the LSTM hybrid model had poorer RMSE scores on test data, the predictions given by this model appeared as a regularized form of the true values.

This result indicates that the model may be successful in modelling trends more effectively than sharp price changes. Exploration of the LSTM hybrid model in prediction of long term stock forecasting could be of value.

Regarding the second key question of interest, RMSE results depicted in *figure 9* indicate that expanding the hybrid model to include a wider range of stocks successfully improved predictive accuracy. The linear multi-company model performed best on Microsoft validation data while the LSTM hybrid model performed best on Google test data. This finding allows us to conclude that increased generalization was attained by expanding the set of companies used in the model training process.

Further model testing is required to understand irregularities found in the results including the spike in RMSE results for the single company LSTM hybrid model, as well as cause of differing success between models on Google data vs. Microsoft Data. One possible explanation for these results is the size of test data. The Microsoft validation data was quite small compared to Google test data meaning a few very inaccurate predictions could have a large effect on RMSE. It is possible that our model generally performs well, but misses large changes in price causing certain days with an abnormally large predictive error.

Regardless of the success of this work, future hybrid model development 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. The current framework has some unnecessary complexity that should be removed to create a simplified testing and reporting process for results. Given the positive results achieved by the 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 Limitations and Challenges

The current research provides a promising foundation for future investigations regarding the utilization of hybrid modeling and sentiment analysis in stock forecasting. By successfully answering our research questions, we have found that the hybrid model enhances the accuracy of stock price predictions compared to the base LSTM model.

These results demonstrate the potential benefits of combining multiple modeling techniques with sentiment analysis in financial forecasting, indicating that the hybrid approach can provide more reliable and accurate stock market predictions. As a result, this study's findings offer valuable insights to researchers and practitioners looking to develop more effective forecasting methods and pave the way for future research in this field.

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. We found out that 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 use financial news headlines instead 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.

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 the 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 the Prophet, our implementation requires the Prophet model to make 10-day predictions of stock price change repeatedly. 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.

Future work could expand our framework using a cloud service to allow widespread database access and increased processing power by creating 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 the use of cloud-based services.

8 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[14]. 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.

9 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 successfully improved upon the base LSTM. Further, The use of linear regression in the hybrid model achieved improvement over the base LSTM, regardless of the companies used for training.

A greater number of training stocks appeared to improve the generalization of the model.

Overall, the expanded model was able to improve upon the single-company hybrid model in predicting price changes of an unseen test stock. 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 improved results.

Further model testing on a wide range of stocks is required to confirm the findings of this work and explore the generalization of results.

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] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.