Forecasting S&P 500 Index Closing Price using LSTM and finBERT

Divya Susan Thomas

Department of Computer Science

The University of Texas at Austin

ABSTRACT

Predicting the movement of the stock market is of vital importance to traders, investors and portfolio managers worldwide. Due to the Covid-19 Pandemic lockdown and the lowering barrier to entry to investing, stock market participation has increased greatly. This has made the market more complex and volatile. The increasing availability of computational resources has led many to employ advanced machine-learning techniques to predict stock market movements. Long short-term memory (LSTM) models have shown promising predictive power. This study used LSTM models to predict the next day's closing price of the S&P 500 index. Fundamental, macroeconomic and technical data as well as unstructured textual data obtained from financial headlines were collected for the ten years between 2010-2020 and used to train the models. Sentiment scores from the textual data were obtained using the finBERT model. Both Single layer and multi-layer LSTM models were developed. After hyperparameter tuning was performed to optimize the models, they were trained on both the dataset including sentiment from headlines and one without. Model performance was evaluated using Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE) and Pearson correlation coefficient (R). Single-layer LSTM models performed better than multi-layer models and the best LSTM model which had a single layer of 10 neurons showed high forecasting accuracy even during the unusual COVID-19 market. Furthermore, including sentiment data from headlines did not improve the prediction accuracy of the models.

Keywords: S&P 500 index, LSTM, finBERT, Sentiment Analysis, Time series forecasting

1. INTRODUCTION AND RESEARCH BACKGROUND

Stock market prediction is a complex and challenging task but one that is important for stock market investors and traders. We are currently at the end of a decade-long bull run where the stock market experienced a relatively steady rise and lower volatility. As we enter an era with more stock price fluctuations, a good price predictive model is important, especially for participants in inverse and leveraged Exchange Traded Funds (ETFs) to reduce risks and make gains in the market. However, stock price prediction is difficult because the markets are noisy, non-parametric, non-linear and deterministic chaotic systems (Ahangar et al, 2010). Moreover, the optimal set of variables selected for price prediction varies greatly. Some studies only utilize historical data or technical indicators while others use a combination of technical indicators and historical data (Di Persio & Honchar, 2016; Wang & Kim, 2018; Qiu & Song, 2016; Wu et al., 2022). Feature selection is of great importance as the performance, interpretability and robustness of the model depend on this. The predictive power of the model may be limited if too few features are selected whereas the performance may be worse if too many are selected due to overfitting to noise.

Bhandari et al. implemented an Long Short Term Memory (LSTM) model with high predictive accuracy to predict the closing price of the S&P 500 index using only historical, macroeconomic and technical data (Bhandari et al., 2022). Other studies which used only unstructured textual data to predict stock prices also found promising results as discussed in more detail in the following paragraphs.

Since there was no study which used LSTM trained on both structured and unstructured data to predict the S&P 500 index, this study focused on training single and multi-layer LSTM models to predict the closing price of the S&P 500 index utilizing a combination of historical, macroeconomic and technical data as well as using sentiment analysis of financial news obtained using FinBERT. The goal of this study was to find the architecture of the LSTM model with the highest accuracy in forecasting the next day's closing price of the S&P 500 index and also investigate if the performance of the LSTM model was improved by incorporating unstructured textual information such as sentiment extracted from financial news using FinBERT, which is a pre-trained NLP model that has been shown to outperform all current state-of-the-art models in analyzing sentiment of financial text(Liu et al.,2021).

Prior studies have found LSTM models to be quite effective for stock market prediction. Chen et al. used an LSTM model to predict China stock returns using historical data and found the model

to be promising. Similarly, in 2017, Roondiwala et al. used the LSTM model to predict the closing prices of NIFTY 50 using only the open, close, low and high prices of the index from 2011 to 2016 with RMSE value of 0.0086 (Roondiwala, Patel, & Varma, 2017). LSTM networks were also found to outperform memory-free classification methods, i.e., a random forest (RAF), a deep neural net (DNN), and a logistic regression classifier (LOG) in predicting movements of S&P 500 stocks from 1992-2015(Fischer & Krauss, 2018). The use of the LSTM model trained on price history along with technical indicators to forecast S&P 500 prices one minute, five minutes and ten minutes ahead for high-frequency trading also has shown promising results (Lanbouri & Achchab, 2020). Furthermore, on comparing the performance of LSTM, Backpropagation, SVM, and Kalman filter in predicting stock price, LSTM was the best choice in terms of prediction accuracy with low variance (Karmiani, Kazi, Nambisan, Shah, & Kamble, 2019). Similarly, a study investigating the performance of the LSTM, Support Vector Regressor, and ARIMA in predicting the stock price of the S&P 500, DJIA, Nikkei 225, Hang Seng Index, China Securities Index 300, and ChiNext index found that LSTM outperformed other models for S&P 500 data (Yu & Yan, 2019). Finally, previous studies that have implemented single-layer and multi-layer LSTM models to predict stock market index closing prices have found the performance of the single-layer LSTM to be superior (Bhandari et al., 2022; Yadav, Jha, & Sharan, 2020).

Researchers also found encouraging results from only taking non-quantifiable data such as financial news articles about Apple Inc. and predicting its future stock trend with news sentiment classification (Kalyani et al., 2016). Gite et al. (2021) implemented an LSTM model using historical stock data along with sentiments from news items to predict the closing price of the NSE (Indian stock market index) and found it to be a better predictive model than LSTM based on just historical price data. Similarly, predictive models built on both stock historical stock prices and financial news showed high accuracy in predicting the index prices in both Vietnam and China(Duong et. al, 2016; Ko & Chang, 2021). The sentiment analysis model based on BERT was found to perform better than TextCNN, TextRNN, Att-BLSTM and TextCRNN(Li et al.,2021). Macroeconomic factors such as the gold index, interest rate, and exchange rate were also found to be highly significant in projecting stock returns (Jabeen et al., 2022).

2. DATA SOURCES AND VARIABLES

This study utilized both structured and unstructured data to predict the closing price of the S&P 500 index. Structured data collected included fundamental, macroeconomic and technical data that were identified to have an impact on the stock price after reviewing the literature. Unstructured data consisted of financial news headlines from Benzinga. Data was collected for a period of 10 years from January 2010 to June 2020 which included the Covid-19 pandemic. In total, 12 variables were collected. These are summarized in Table 1.

Fundamental data collected contained the close price which is the dependent variable in this study. The close price is the price at which the security last trades when the exchange closes on a trading day. Daily closing prices of the S&P 500 were collected from Yahoo Finance for 10 years except at weekends and holidays when the market was closed.

Macroeconomic data used in this study include the US dollar index (USDX), Cboe Volatility Index(VIX), Civilian Unemployment Rate(UNRATE), Effective Federal Funds Rate (EFFR) and the University of Michigan: Consumer Sentiment (UMC-SENT). The US dollar index (USDX) is a metric that measures the value of the US dollar relative to six major foreign currencies, namely, the Euro, Swiss franc, Japanese yen, Canadian dollar, British pound, and Swedish krona. The index indicates the strength of the US dollar in the global markets and generally, there exists a positive correlation between the index and stock prices (Novianti, 2016). Index data used in this study was the daily Adjusted Close price of the index (ticker symbol DX-Y.NYB) collected from Yahoo finance for 10 years.

Cboe Volatility Index (VIX) measures the 30-day forward projection of volatility of the S&P 500 index (SPX). It provides a measure of the market sentiment, especially fears among investors. Studies have generally shown a strong inverse relationship between VIX and the price of the S&P 500 index (Ruan, 2018). In this study, the historical daily adjusted Close price of VIX for the 10 years from 2010-2020 obtained from Yahoo Finance was used. Civilian Unemployment Rate (UNRATE) is a metric that represents the number of unemployed individuals who are 16 years or older, residing in one of the 50 US states, as a percentage of the US labour force. Studies have shown it to be a strong predictor of stock prices (Pan, 2018; Farsio & Fazel,2013). Therefore, the monthly UNRATE date for the 10 years collected from FRED was used in this study. The monthly data was then converted to daily data using the forward-filling method. Effective Federal Funds Rate (EFFR) is a volume-weighted median of overnight federal funds transactions reported in the

FR 2420 Report of Selected Money Market Rates. It has been shown to affect inflation, growth and unemployment and generally, has an inverse relationship with stock prices (Bernanke & Kuttner, 2005). Daily EFFR data for the 10 years was collected from the FRED website. The final macroeconomic indicator utilized was the monthly University of Michigan: Consumer Sentiment (UMCSENT) data which was also collected from FRED. UMCSENT is a monthly survey conducted by the University of Michigan that measures consumer sentiment and expectations of the economy and future spending. It has been shown to influence stock prices (Lansing & Tubbs, 2018).

Technical indicators used in this study are the three most popular indicators used by traders, namely, Moving Average Convergence Divergence (MACD), Relative Strength(RSI) and Average True Range(ATR). Daily values for these over the 10 years were computed by the historical stock market data gathered from Yahoo Finance. MACD, calculated by subtracting the 26-day exponential moving average from the 12-day exponential moving average is used by traders to determine the direction and momentum of a stock price. ATR measures the volatility of a stock over a period. Commonly used periods are 14,20 and 22 days. In this study, 14 days was chosen as the period since it is the most commonly used. RSI measures the momentum of stock and is used to identify bearish and bullish price signals. All three of these indicators have been shown to have the ability to predict stock prices over shorter periods (Rodriguez-Gonzalez et al.,2011; Anghel,2015).

Unstructured data used in this study was the 'Daily Financial news for 6000+ Stocks' dataset found on Kaggle which contains around 4 million financial article headlines from the 2009-2020 period scrapped from Benzinga. This massive data set contained headlines pertaining to around 6000 stocks, many of which did not belong to the S&P 500 index. Nearly 50% of the S&P 500 index's price is determined by the largest 35 companies. Therefore, in this study, only headlines related to these 35 companies were extracted to make the dataset more manageable. Sentiment analysis was conducted on all the daily headlines between 2010 and 2020 pertaining to these companies from the dataset to obtain positive, negative and neutral S&P 500 news sentiment scores for every day for ten years.

Two datasets were prepared for the study. Dataset 2 consisted only of the fundamental, macroeconomic and technical indicators. Dataset 1 contained in addition to these, the sentiment scores of headlines. This has been done so we can investigate if incorporating unstructured textual

information such as sentiment extracted from financial news and adding the textual data can improve the predictive power of the LSTM model. This information is summarized in Table 1.

Table 1: Datasets and Variables used in study

Datasets	Content of Dataset	Variables
Dataset 1	Fundamental data, macroeconomic data, technical indicators, Sentiment scores from headlines	Close price, VIX, EFFR, UNRATE, UMCSENT, USDX, MACD, ATR, RSI, Positive sentiment score, Neutral score, Negative score
Dataset 2	Fundamental data, macroeconomic data, technical indicators	Close price, VIX, EFFR, UNRATE, UMCSENT, USDX, MACD, ATR, RSI

3. METHODS

3.1 S&P 500 Close Price Denoising

As can be seen in Figure 1, the closing price of the S&P 500 shows complex, noisy behaviour. Studies have shown discrete wavelength transformations and in particular, Haar wavelengths to be highly effective in denoising stock price data (Ortega & Khashanah,2014). Therefore, the closing price was denoised by using the haar wavelength with soft thresholding and the VisuShrink approach.

Figure 1:Movement of S&P 500 close price from 2010-2020

3.2 Preprocessing and Sentiment Analysis on financial news headlines

As mentioned in Section 2, using the Kaggle financial news headlines dataset, daily positive, neutral and negative sentiment scores were obtained for the ten years from 2010-2020. First, only

headlines pertaining to the largest 35 companies in the S&P 500 index were extracted into a dataframe. Then AutoTokenizer was called with the *from_pretrained*() method to return the correct tokenizer class instance for the finBERT model and then the pre-trained finBERT model was fetched from Hugging Face. Each headline in the dataframe was then tokenized and fed to the finBERT model. The output logits were passed through a softmax layer to get a tensor with 3 values representing the positivity, neutrality and negativity of each headline. The scores for all the headlines are then aggregated by date and the average is found to give a positive, negative and a

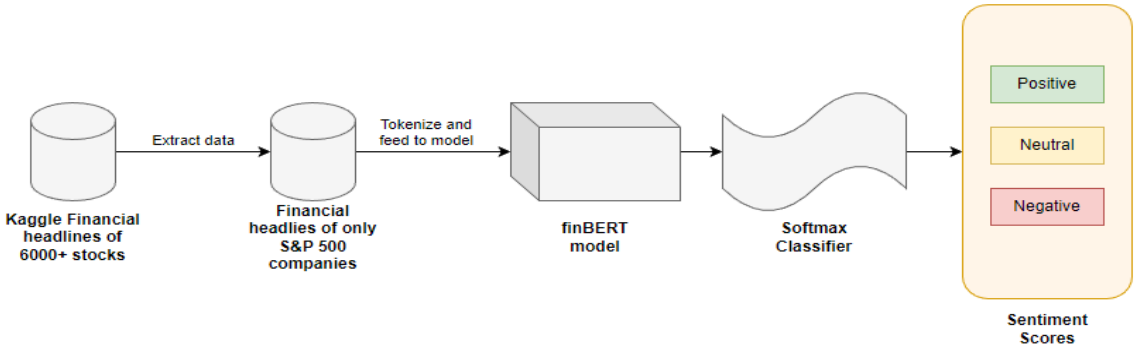

Figure 2:Workflow of extracting sentiment scores

neutral average score for each day. These average sentiments are taken to be the daily S&P 500 positive, neutral and negative sentiment scores. These were then joined to the structured data using the date as a key. This process is depicted in Figure 2.

3.3 Data Normalization

The range and standard deviation of the input variables also varied greatly from each other. For example, as shown in Figure 3, the Close price ranges from 250 to 3000 while sentiment score variables range from 0 to 1. Therefore, there was a need to perform feature scaling to ensure that the machine learning algorithms performed well. Min-max normalization technique using MinMaxScaler from sklearn was used to scale all features to values between 0 and 1.

3.4 Data Split

After this, the dataset was a two-dimensional array with several observations and features, but LSTM expects three-dimensional input data (number of observations, number of input features and size of time step). Therefore, the data was reshaped to meet this. The size of the time step used in this study is 5. The data was split into training and testing sets using an 80/20 split ratio. The order of the time series data was preserved during the split. The training set was used to train the

LSTM models while the testing set was used to evaluate the performance of the models. The last 20% of the 80% of the training data (16% of total data) was used for validating during

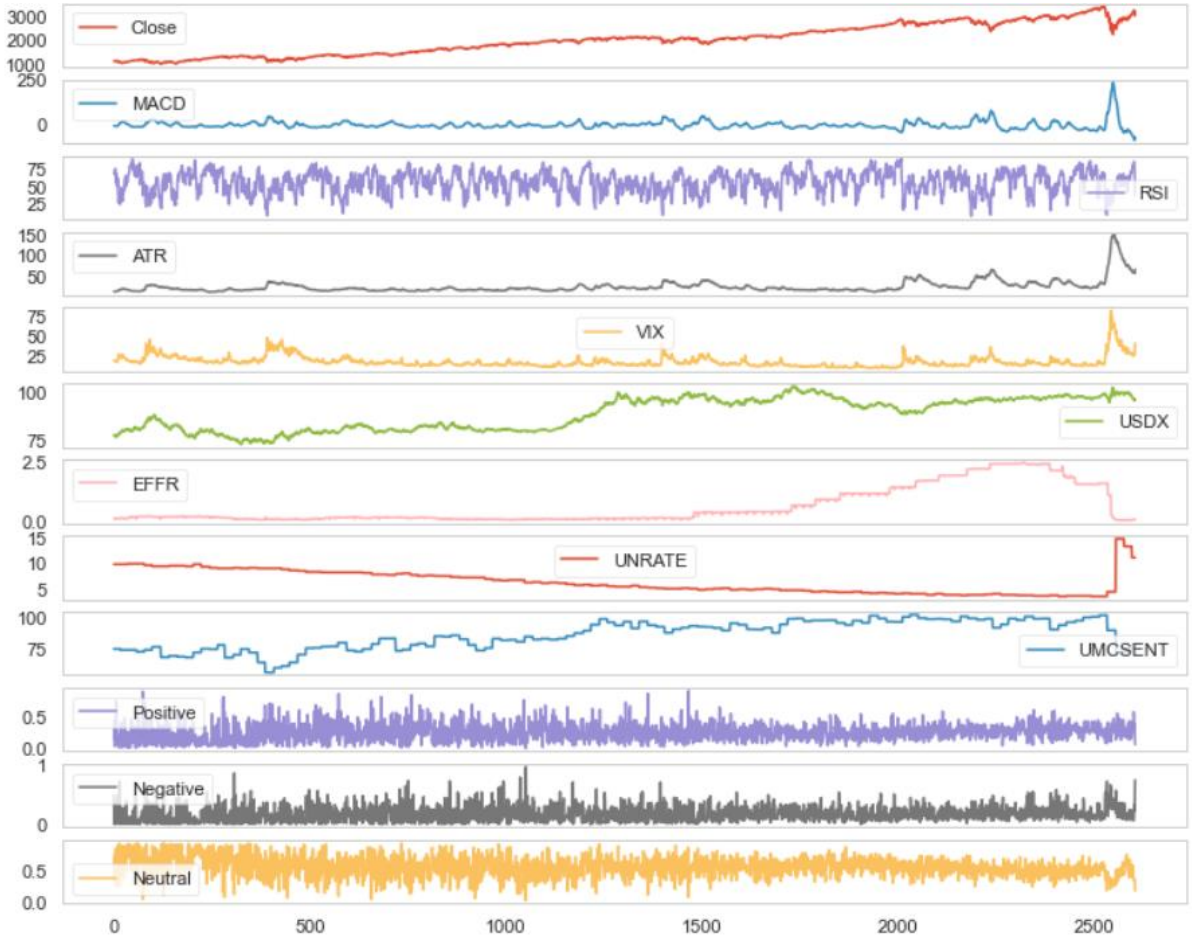

Figure 3: Snapshot of the input variables showing their varying ranges

hyperparameter tuning and once tuning was completed, this data was included back into the training set.

3.5 Hyperparameter tuning

Table 2: Summary of Hyperparameter settings used for tuning

Hyperparameter tuned	**Settings tested**
Batch size	8, 16, 32
Learning rate	0.001, 0.01, 0.1
Type of Optimizer	Adam, Adagrad, Nadam

To find the best model architecture, several variations were tested. Both single-layer LSTM and multi-layer LSTM models were fitted to the data. For single-layer LSTM architecture, 7 different

models with 2,10,30,50,100,150 and 200 neurons respectively were tested to find the best model. For the multi-layer LSTM architecture, 6 different model variations were tested. These multi-layer configurations were (8,4), (10,5), (10,5,2), (50,20), (150,100) and (100,50,20) where L1, L2 and L3 in (L1, L2, [L3]) represents the number of neurons in the first, second and third hidden layer respectively. For each model, hyperparameter tuning was performed to find the optimal values for each hyperparameter. The hyperparameters tuned in this study were batch size, learning rate and type of optimizer. The optimizers tested were Adam, Adagrad and Nadam. Learning rates tested were 0.1,0.01 and 0.001 and the batch sizes tested were 8,16 and 32.

Hyperparameter tuning was performed for each combination of hyperparameters and each model, with 50 epochs, a time step of size 5 and repeated 10 times to account for the stochastic nature of LSTM models. *EarlyStopping* callback with validation loss as the performance metric and patience (number of epochs with no improvement after which training will be stopped) of 5 was also used. Average Root Mean Square Error (RMSE) values were calculated on the validation data and the combination that resulted in the lowest RMSE value was the optimal hyperparameter setting for that model. Table 2 lists the hyperparameter settings used for tuning and Tables 3-5 list the optimized hyperparameter setting found after tuning for every LSTM model in this study.

Table 3:Values of the optimized hyperparameters for single layer LSTM models on Dataset 1

No of Neurons	Batch size	Learning Rate	Type of Optimizer
2	8	0.1	Adam
10	8	0.001	Adam
30	16	0.1	Adagrad
50	16	0.1	Adagrad
100	32	0.1	Adagrad
150	32	0.01	Adagrad
200	32	0.01	Adagrad

Table 4::Values of the optimized hyperparameters for multi-layer LSTM models on Dataset 1

No of Neurons and multi-layer model	Batch size	Learning Rate	Type of Optimizer
[8,4]	16	0.01	Adam
[10,5]	32	0.01	Adam
[10, 5, 2]	4	0.01	Adam
[50, 20]	4	0.01	Adagrad
[150, 100]	16	0.01	Adagrad
[100, 50, 20]	8	0.001	Adagrad

These were the hyperparameter settings on which the optimized models were trained and evaluated on the test data.

Table 5:Values of the optimized hyperparameters for single layer LSTM models on Dataset 2

No of Neurons	Batch size	Learning Rate	Type of Optimizer
2	8	0.1	Adam
10	16	0.01	Adam
30	16	0.1	Adagrad
50	8	0.1	Adagrad
100	32	0.1	Adagrad
150	32	0.01	Adagrad
200	32	0.01	Adagrad

3.6 Training and testing models with optimized hyperparameters

Once all the hyperparameters were optimized, these models with the optimized hyperparameter setting shown in Tables 3-5 were fitted on the complete training data and their performance were evaluated on the test data. Performance metrics used in this study were Average Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE) and Pearson correlation coefficient (R) which were computed as shown in Equation 1 below where y_i is the actual value at time i, $\hat{y}_i$ is the forecasted value at time t, $\bar{y}_i$ is the mean of the actual time series, $\hat{y}_i$ bar is the mean of the forecasted time series and N is the number of observations. Smaller values of RMSE and MAPE indicate smaller errors and hence better model performance. In contrast, the larger value of R is better since that indicates better linear correlation between the actual and predicted data points.

Equation 1:Formulas for the performance metrics

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2},$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right|,$$

$$R = \frac{\sum_{i=1}^{N}(y_i - \bar{y}_i)(\hat{y}_i - \bar{\hat{y}}_i)}{\sqrt{\sum_{i=1}^{N}(y_i - \bar{y}_i)^2(\hat{y}_i - \bar{\hat{y}}_i)^2}}$$

Each of the models with the optimized hyperparameters was trained using 100 epochs, a time step of size 5 and repeated 30 times to account for the stochastic nature of LSTM models. *EarlyStopping* with loss performance metric and patience of 5 is also employed. Both single-layer and multi-layer models were trained using the same values for these parameters. The

average of RMSE, MAPE and R values from the 30 replications were computed. The average RMSE value was chosen as the main selection criteria. So, the model with the smallest average RMSE, followed by the smallest MAPE value and largest R was considered to be the best model.

In order to decide if adding sentiment data improves the LSTM model, the models were trained on 2 datasets – As described in the previous section, Dataset 1 included the sentiment score data while Dataset 2 did not include this. Average RMSE scores of the models trained on both the datasets were used to decide if sentiment scores improved model performance.

A flowchart of the methodology proposed in this study is depicted in Figure 4.

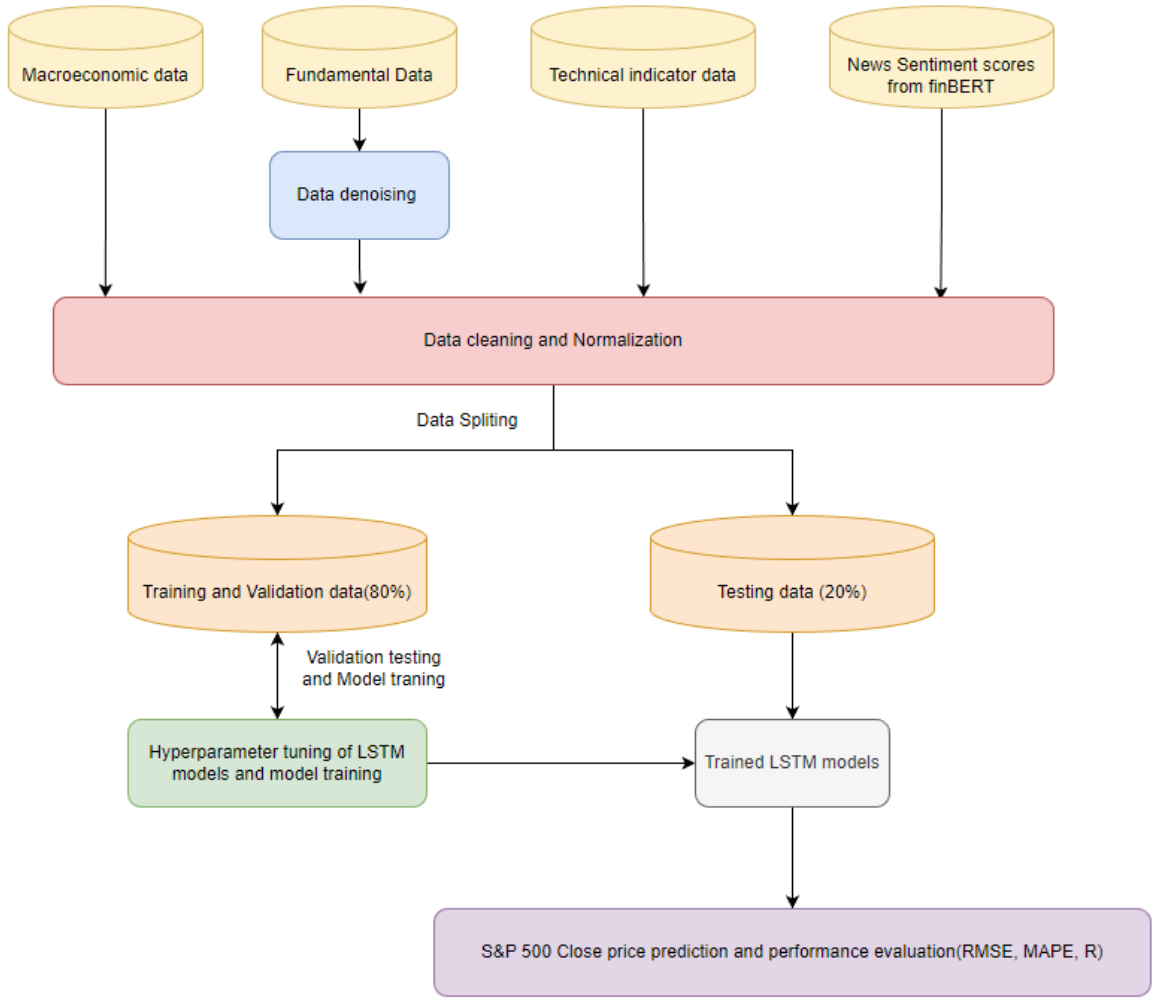

Figure 4: Schema of proposed study methodology

4. RESULTS AND DISCUSSION

4.1 Single Layer LSTM models' results on Dataset 1

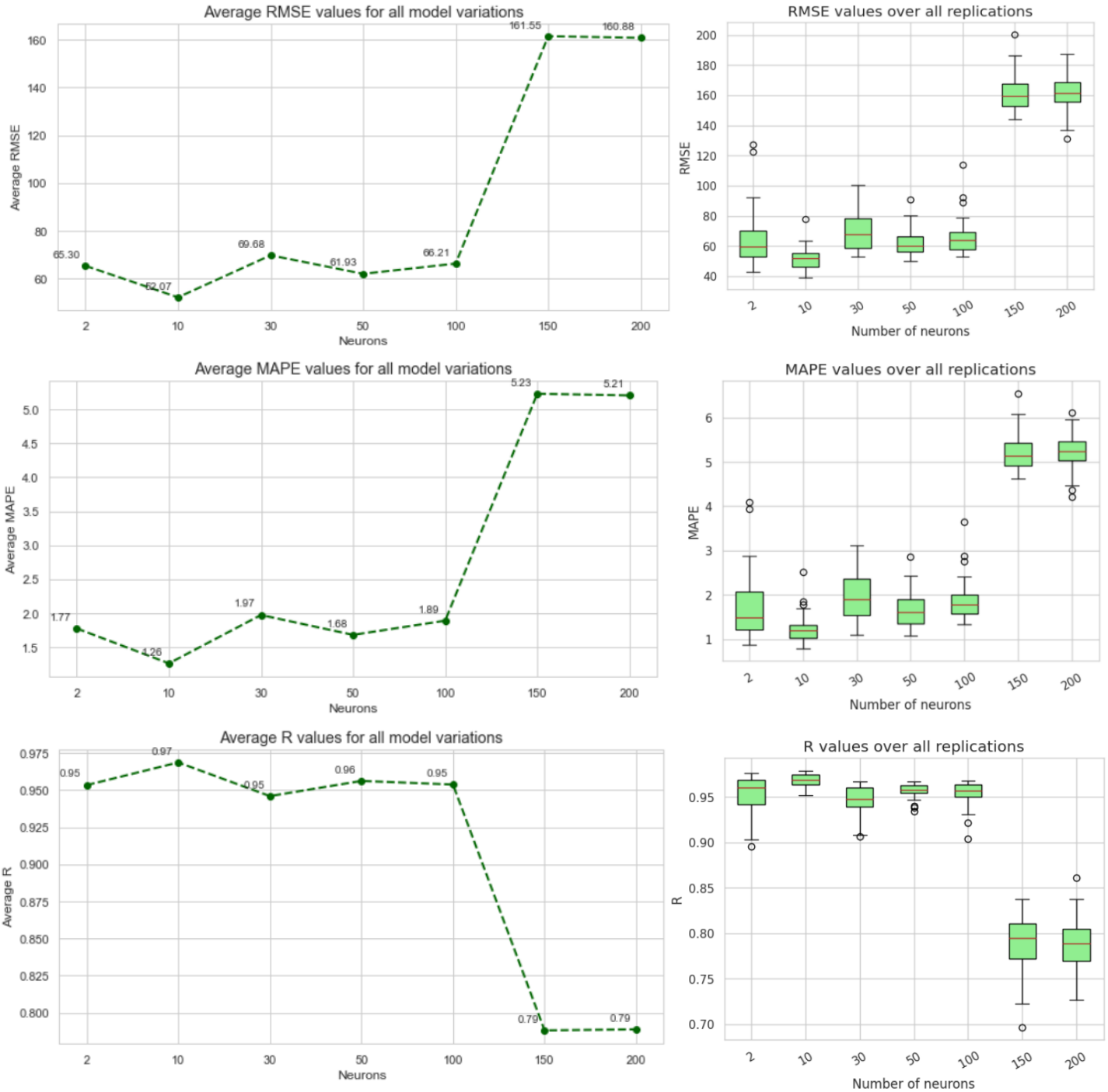

Figure 5: Results for the 7 different Single Layer LSTM models on Dataset 1 over 30 replications

Using the optimized hyperparameters, the 7 single-layer LSTM models, namely with 2,10,30,50,100,150 and 200 neurons were tested to find the best model for dataset 1 which is the complete dataset that contained both structured data (Macroeconomic, technical and fundamental data) and unstructured data(sentiment scores from headlines). The model with 10 neurons had the lowest average RMSE value of 52.07, MAPE value of 1.26 and the largest R-value of 0.97. Hence it was found to be the best model. The line plot in Figure 5 shows the variation in Average RMSE, MAPE and R for the 7 different models and the boxplots show the variation in RMSE, MAPE and R over 30 replications for each of the 7 single-layer LSTM models tested in this study. The model with 10 neurons had the seen from Figure 5, average RMSE and MAPE increase as the number of neurons drop below 10 and increase as the number of neurons increases above 10. This might be because a larger number of neurons result in overfitting while a lower number of neurons result in underfitting.

The original closing price data and the forecasted closing price values obtained from the best single-layer model with 10 neurons which had the lowest average RMSE value are depicted for the ten years of the study in Figure 6. Between 2010 to June 2018, the green curve represents predictions in the training data whereas, between June 2018- June 2020, it shows predictions in the test data. It can be seen from the figure that the forecasted values closely overlap with the actual values for the most part except for the 2020 area where more variation between the curves is visible. This deviation could be due to unusual market conditions created by the covid-19 pandemic.

Figure 6:Actual S&P 500 Close price vs the S&P 500 Close price forecasted by the best single layer LSTM model on Dataset1 (10 neurons)

However, even during the unexpected time period of 2020 in the test data, the model does capture the sudden drop and the subsequent V-shaped sharp recovery of the market. This is indicative of the strong forecasting ability of the model and suggests that it is not over-fitted.

4.2 Multi- Layer LSTM models' results on Dataset 1

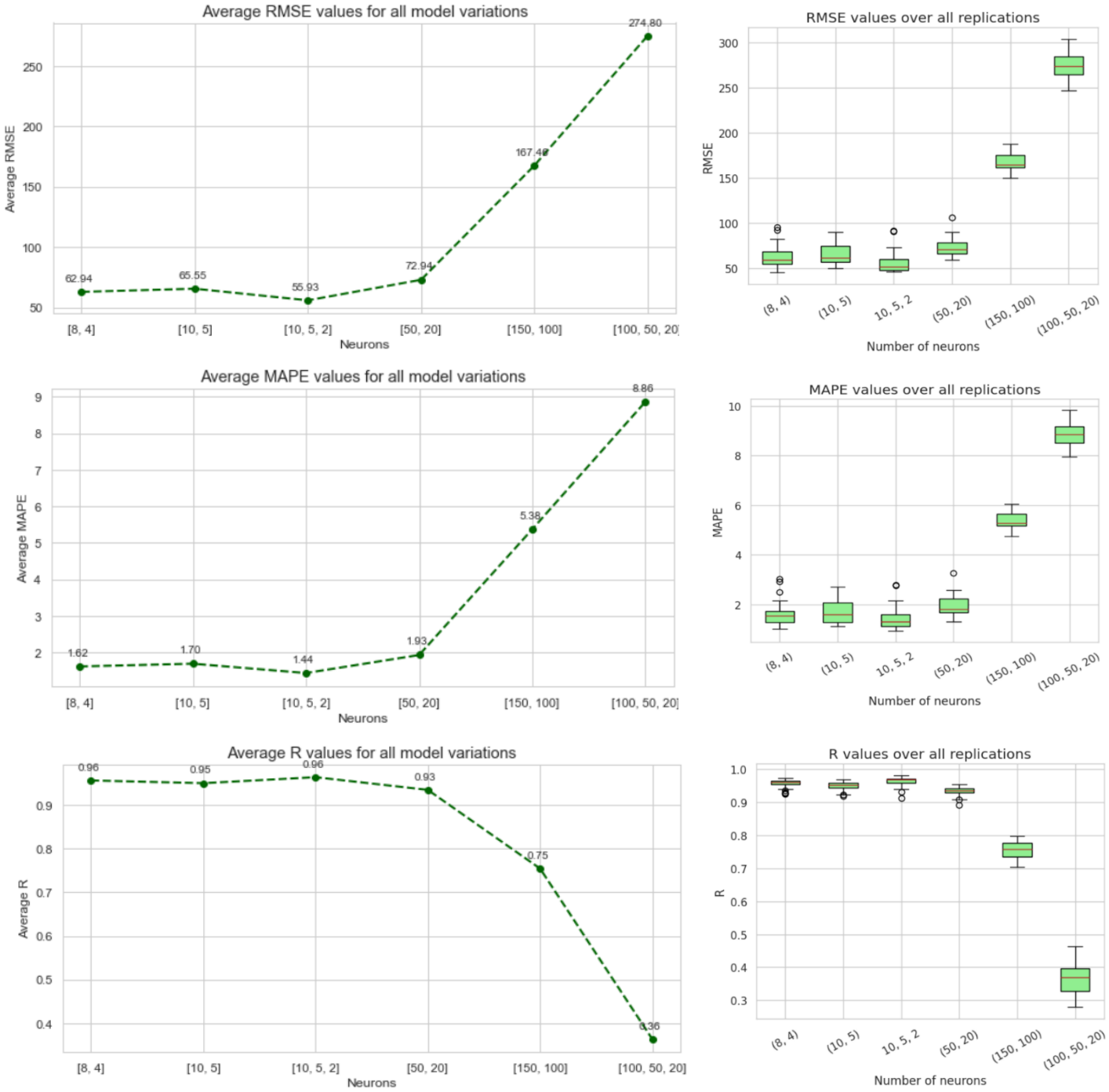

Figure 7:Results for the 6 different Multi-Layer LSTM models on Dataset 1 over 30 replications

Even though single-layer LSTM gave pretty good results, several multi-layer LSTM models were also tested to see if any of them would result in improving the accuracy of predicting the S&P 500 index close price. Using the optimized hyperparameters, the 6 multi-layer LSTM models were tested to find the best model for dataset 1. These were models with hidden layers (8,4), (10,5), (10,5,2), (50,20), (150,100) and (100,50,20) where the numbers represent the number of neurons in each hidden layer. Multi-layer LSTM model with 3 hidden layers having 10, 5 and 2 neurons respectively had the lowest average RMSE value of 55.93, MAPE value of 1.44 and R-value of 0.96. Hence it was found to be the best model. The line plot in Figure 7 shows the variation in average RMSE, MAPE and R for the 6 different models and the boxplots show the variation in RMSE, MAPE and R over 30 replications for each of the 6 multi-layer LSTM models tested in this study.

The original closing price data and the forecasted closing price values obtained from the best multi-layer model with 3 hidden layers having 10, 5 and 2 neurons respectively and which had the lowest average RMSE value is depicted for the ten years period of the study in Figure 8. Between 2010 to June 2018, the green curve represents predictions in the training data whereas, between June 2018- June 2020, it shows predictions in the test data. It can be seen from the figure that the forecasted values closely overlap with the actual values for the most part except for the

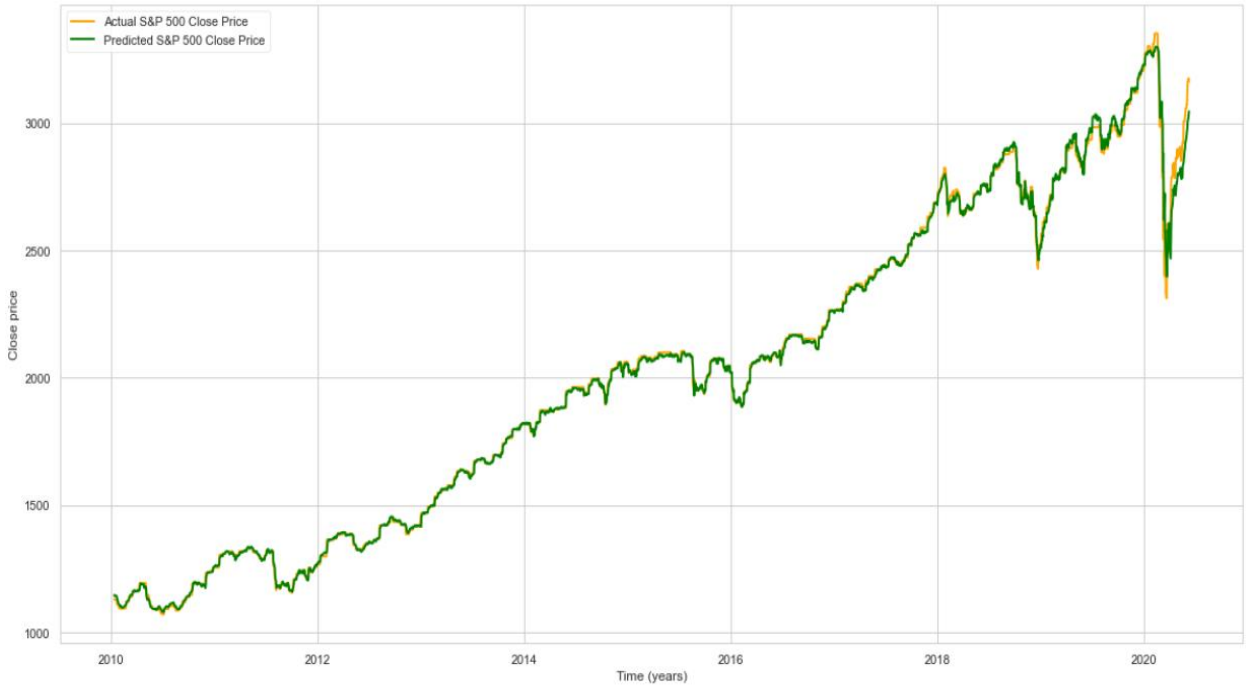

Figure 8:Actual S&P 500 Close price vs the S&P 500 Close price forecasted by the best multi- layer LSTM model on Dataset1 (10-5-2 model)

2020 area where more variation between the curves is visible. However, the overlap is not as close as in the case of the best single-layer model, especially around 2020 when unusual market conditions created by the covid-19 pandemic arose. This is to be expected since the best single-layer model has a lower RMSE value compared to the multi-layer model. However, even during the unexpected time period of 2020 in the test data, the model does capture the sudden drop and the subsequent V-shaped sharp recovery of the market. This is indicative of the strong forecasting ability of the model and suggests that it is not over-fitted. Therefore, even the best multi-layer model appears to have good predictive power. However, since the single-layer LSTM model is simpler and has a lower average RMSE value, it is a better choice than the multi-layer LSTM model since multiple layers do not seem to improve the predictive power of the model but rather seem to reduce it.

4.3 Comparing Single Layer LSTM's results on Dataset 1 and Dataset 2

Though the results showed that the single-layer LSTM model was better than the multi-layer model in this scenario, further exploration was needed to find out if adding the sentiment scores extracted from financial headlines improves the LSTM model. To do this, the single-layer LSTM models were trained on Dataset 2 which did not contain the sentiment data. Using the optimized hyperparameters for Dataset 2, the 7 single-layer LSTM models, namely with 2, 10, 30, 50, 100, 150 and 200 neurons were tested to find the best model. The model with 50 neurons had the lowest average RMSE value of 49.75, MAPE value of 1.21 and the largest R-value of 0.97. Hence it was found to be the best model. The line plot in Figure 5 shows the variation in Average RMSE, MAPE and R for the 7 different models and the boxplots show the variation in RMSE, MAPE and R over 30 replications for each of the 7 single-layer LSTM models tested in this study.

As can be seen from Figure 59, average RMSE and MAPE increase as the number of neurons increase above 50 neurons but when the number of neurons decreases, the error initially increases but interestingly drop for the model with 10 neurons before increasing again. The original closing price data and the forecasted closing price values were obtained from the best model with 50 neurons which had the lowest average RMSE value is depicted for the ten years period of the study in Figure 10. Between 2010 to June 2018, the green curve represents predictions in the training data whereas, between June 2018- June 2020, it shows predictions in the test data. It can be seen from the figure that the forecasted values closely overlap with the actual values for the most part except for the 2020 area where more variation between the curves is visible. The overlap is slightly

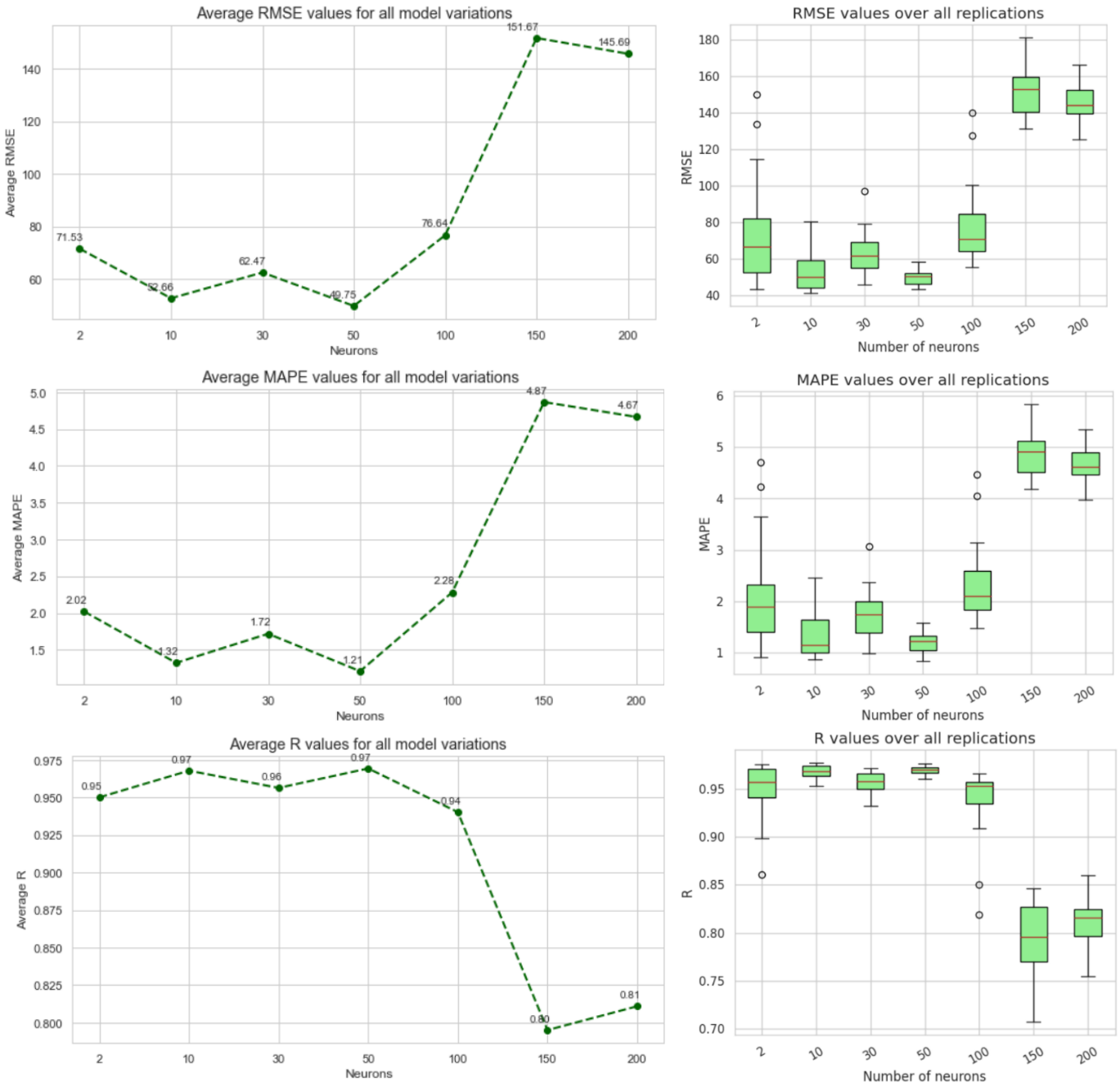

Figure 9: Results for the 7 different Single Layer LSTM models on Dataset 2 over 30 replications

even better than that found for the 10-neuron single-layer LSTM model trained on dataset 1 which included the sentiment scores. This is to be expected since this best single-layer model with 50 neurons had a lower RMSE value compared to the 10-neuron model trained on dataset 1. However, there are still deviations from the actual value even in this model though the model does capture the sudden drop and the subsequent V-shaped sharp recovery of the market just like other

previously mentioned models and is indicative of the strong forecasting ability of the model and suggests that it is not over-fitted.

Based on the results shown above, it appears that the addition of financial news headlines sentiment scores does not seem to improve the predictability of the single-layer LSTM model. When sentiment scores are included, the best model, an LSTM model with a 10-neuron single layer, has an average RMSE of 52.07 whereas when sentiment scores are not included the best model is an LSTM with a 50-neuron single layer which gives an average RMSE of 49.7 over 30 replications. This might suggest that since the RMSE values are similar, it might be better to include sentiment scores since then the LSTM model architecture needs to have only 10 neurons. However, since the 10-neuron single-layer model trained on dataset 2 (without sentiment scores) also has an average RMSE score of 52.6, it suggests that even if the goal is to have a simpler LSTM model, adding the sentiment scores from the financial headlines does not seem to help improve the model performance. This is perhaps because including the daily financial headlines might be adding more noise to the dataset than any actual extra information that is not already contained in the macroeconomic and technical data variables included in the study which includes customer sentiment (UMCSENT).

Figure 10:Actual S&P 500 Close price vs the S&P 500 Close price forecasted by the best single- layer LSTM model on Dataset 2 (50 neurons model)

5. CONCLUSION AND FUTURE WORK

The ability to predict stock prices accurately has always been coveted by many traders around the world. Prior studies have shown LSTM models, in particular single-layer LSTM models trained on macroeconomic, technical fundamental data to be very effective at predicting the close price of the S&P 500 index. Studies using sentiment score data from financial news and other unstructured data to predict stock prices have also shown promise. This study investigated if incorporating daily averaged sentiment scores of news headlines pertaining to companies in the S&P 500 index along with the macroeconomic, technical and fundamental data would help improve the predictive power of LSTM models. First, the study used finBERT, a state-of-art BERT model pretrained on financial text model to extract sentiment scores from financial headlines which were then added to the dataset. A variety of single-layer and multi-layer LSTM models were then implemented and extensively tuned to find the optimal hyperparameter settings. These models were then fitted on both the complete dataset and the dataset with the sentiment scores removed. Their performance was then evaluated using Average RMSE, MAPE and R scores. Results showed that the single-layer LSTM model performed better than multi-layer LSTM models. This validates previous studies that have also shown the same result. A single-layer LSTM model with 10 neurons was found to be the best fit and offered high prediction accuracy. The model was able to forecast rapid fluctuations of the S&P 500 index close price even in highly unusual market environments during the COVID-19 pandemic. This is a very promising outcome and suggests that traders seeking to predict the markets might find it beneficial to give LSTM models a closer look. Results also showed that adding the sentiment scores did not improve the model performance but rather decreased it slightly. This might be because the daily financial headlines sentiment scores might be adding noise but no new information to the dataset since it already contained technical and macroeconomic data including consumer sentiment.

Many tasks can be explored in future studies. The financial headlines analyzed in this study only came from Benzinga. Even though this is a popular news source among traders, headlines collected from only one news source might have biases. To avoid this, headlines can be collected from a variety of popular news sources. Similarly, sentiment from other unstructured data such as social media sentiment, analyst reports and annual reports can also be incorporated. Furthermore, hybrid models that combine LSTM with other deep learning models can also be developed and tested to compare their performance. Finally, to test the robustness and adaptability of the proposed

models, they could be used to predict market indices in other countries with economies and consumer behaviours similar to the United States.

REFERENCES

Ahangar, R. G., Yahyazadehfar, M., & Pournaghshband, H. (2010). The comparison of methods artificial neural network with linear regression using specific variables for prediction stock price in Tehran stock exchange. *arXiv preprint arXiv:1003.1457*.

Anghel, G. D. I. (2015). Stock market efficiency and the MACD. Evidence from countries around the world. *Procedia economics and finance*, *32*, 1414-1431.

Bernanke, B. S., & Kuttner, K. N. (2005). What explains the stock market's reaction to Federal Reserve policy?. *The Journal of finance*, *60*(3), 1221-1257.

Bhandari, H. N., Rimal, B., Pokhrel, N. R., Rimal, R., Dahal, K. R., & Khatri, R. K. (2022). Predicting stock market index using LSTM. *Machine Learning with Applications*, 100320.

Chen, K., Zhou, Y., & Dai, F. (2015, October). A LSTM-based method for stock returns prediction: A case study of China stock market. In *2015 IEEE international conference on big data (big data)* (pp. 2823-2824). IEEE.

Di Persio, L., & Honchar, O. (2016). Artificial neural networks architectures for stock price prediction: Comparisons and applications. *International journal of circuits, systems and signal processing*, *10*(2016), 403-413.

Duong, D., Nguyen, T., & Dang, M. (2016, January). Stock market prediction using financial news articles on ho chi minh stock exchange. In *Proceedings of the 10th International Conference on Ubiquitous Information Management and Communication* (pp. 1-6).

Farsio, F., & Fazel, S. (2013). The stock market/unemployment relationship in USA, China and Japan. *International Journal of Economics and Finance*, *5*(3), 24-29.

Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European journal of operational research*, *270*(2), 654-669.

Jabeen, A., Yasir, M., Ansari, Y., Yasmin, S., Moon, J., & Rho, S. (2022). An Empirical Study of Macroeconomic Factors and Stock Returns in the Context of Economic Uncertainty News Sentiment Using Machine Learning. *Complexity*, *2022*.

Kalyani, J., Bharathi, P., & Jyothi, P. (2016). Stock trend prediction using news sentiment analysis. *arXiv preprint arXiv:1607.01958*.

Karmiani, D., Kazi, R., Nambisan, A., Shah, A., & Kamble, V. (2019, February). Comparison of predictive algorithms: backpropagation, SVM, LSTM and Kalman Filter for stock market. In *2019 Amity International Conference on Artificial Intelligence (AICAI)* (pp. 228-234). IEEE.

Lanbouri, Z., & Achchab, S. (2020). Stock market prediction on high frequency data using long-short term memory. *Procedia Computer Science*, *175*, 603-608.

Lansing, K. J., & Tubbs, M. (2018). Using sentiment and momentum to predict stock returns. *FRBSF Economic Letter*, *2018*, 29.

Li, M., Chen, L., Zhao, J., & Li, Q. (2021). Sentiment analysis of Chinese stock reviews based on BERT model. *Applied Intelligence*, *51*(7), 5016-5024.

Liu, Z., Huang, D., Huang, K., Li, Z., & Zhao, J. (2021, January). Finbert: A pre-trained financial language representation model for financial text mining. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence* (pp. 4513-4519).

Novianti, M. (2016). *ANALYSIS ON THE INFLUENCE OF SELECTED MACROECONOMIC INDICATORS (CONSUMER PRICE INDEX, TRADE BALANCE, NON-FARM PAYROLL, HOUSING STARTS, AND S&P 500) TOWARDS US INDEX (PERIOD 2010-2015)* (Doctoral dissertation, President University).

Ortega, L., & Khashanah, K. (2014). A neuro-wavelet model for the short-term forecasting of high-frequency time series of stock returns. *Journal of Forecasting*, *33*(2), 134-146.

Pan, W. F. (2018). Does the stock market really cause unemployment? A cross-country analysis. *The North American Journal of Economics and Finance*, *44*, 34-43.

Qiu, M., & Song, Y. (2016). Predicting the direction of stock market index movement using an optimized artificial neural network model. *PloS one*, *11*(5), e0155133.

Rodríguez-González, A., García-Crespo, Á., Colomo-Palacios, R., Iglesias, F. G., & Gómez-Berbís, J. M. (2011). CAST: Using neural networks to improve trading systems based on technical analysis by means of the RSI financial indicator. *Expert systems with Applications*, *38*(9), 11489-11500.

Roondiwala, M., Patel, H., & Varma, S. (2017). Predicting stock prices using LSTM. *International Journal of Science and Research (IJSR)*, *6*(4), 1754-1756.

Ruan, L. (2018). Research on Sustainable Development of the Stock Market Based on VIX Index. *Sustainability*, *10*(11), 4113. MDPI AG. Retrieved from http://dx.doi.org/10.3390/su10114113

Wang, J., & Kim, J. (2018). Predicting stock price trend using MACD optimized by historical volatility. *Mathematical Problems in Engineering*, *2018*.

Wu, S., Liu, Y., Zou, Z., & Weng, T. H. (2022). S_I_LSTM: stock price prediction based on multiple data sources and sentiment analysis. *Connection Science*, *34*(1), 44-62.

Yadav, A., Jha, C. K., & Sharan, A. (2020). Optimizing LSTM for time series prediction in Indian stock market. *Procedia Computer Science*, *167*, 2091-2100.

Yu, P., & Yan, X. (2020). Stock price prediction based on deep neural networks. *Neural Computing and Applications*, *32*(6), 1609-1628.