S&P500 Stock Value Prediction of Vulcan Value Dataset

Renaldy Herlim, Sahil Shah, Jaye Cho

UCSD Datahacks 2021

Abstract- The stock market is something that everyone ranging from economists, researchers and the average person always try to predict. However, it has been proven many times that an index that tracks the top largest companies such as the S&P500 outperforms many who try to beat the market. Therefore there are lots of benefits for trying to predict the value of such an index. In this project, we implement different time-series algorithms that predicts the value of the S&P500 index using a dataset that consists many different indicators related to the stock market and the performance of the S&P500. We will perform analysis to extract hidden patterns and utilize historical data to create and compare models that can predict the value of S&P500 index based on data from the past week, past month, and past year of an arbitrary date. The algorithms we will compare ranges from Least Squares Regression, Gradient Boosted Trees (XGBoost), Long Short Term Memory (LSTM) Neural Network, and the Facebook Prophet model.

Key Words- Time-series Forecasting, Stock Market Prediction, S&P 500, Vulcan Value Partners

1. INTRODUCTION

Our project mainly focuses on attempting different algorithms from simple machine learning models such as the Least Squares Regression and Gradient-Boosted Trees (XGBoost) with more state-of-the-art models like the LSTM and Facebook Prophet. The dataset used for this project was provided by Vulcan Value Partners, we were given a subset of their company's data that include a series of different indexes, and other measurements of the stock market. With this data we will extract features that can describe and predict the value of the S&P500 index and create a model based on these features. We researched publicly available projects that have attempted to predict the stock market to establish our method of prediction, and we decided to attempt to focus on the LSTM model. The goal of our predictive models is for the model to be able to predict the index value given data from the past week, month and year.

2. DATA CLEANING & PREPROCESSING

In order to extract trends and patterns throughout the dataset, we initially started with an exploratory data analysis. We started with just plotting the S&P values over time. From there, we calculated and graphed the exponential moving averages to clean/smooth out the data. We then began to try and find predictors which correlated with S&P. In order to achieve this, we isolated series' which represented indexes or major billion dollar markets. This is due to S&P companies being in that realm of size. After graphing some series indexes, we realized there were null values. We replaced null values with surrounding index averages. This ensures the rigidity of the null values. After replacing these values, there were many categorical/text variables which we needed to convert into numerical data. We decided to one-hot-encode 'frequency', 'seasonal adjustment', and drop the name variable. We then decided to merge the series and observation datasets to improve readability. Lastly, we normalized the data to improve smoothness.

To prepare our data for input into the models we also calculated features that provides additional information about the S&P500 value. The features we created are the daily percent change (Fig.2) as well as the exponential moving averages (Fig. 1) of the value.

3. ANALYSIS AND MODELING

Our first goal during EDA was to find correlators/predictors to S&P as well as finding trends relating to it. We first decided to group all of the indexes to find correlation and didn't find any. We then isolated series', which represented indexes or major billion dollar markets. This is due to S&P companies being in that realm of size. Here, we found that effective federal fund volume, or the amount of loans/money banks give out to corporations, was steadily increasing. We found that this had a correlation of 0.81 in comparison to S&P rise. Next, we found that total assets and liabilities of commercial banks, 'TLAACBW027NBOG' and 'TLBACBW027NBOG' were correlated to S&P rise. The simultaneous rising of assets and liabilities of banks indicated a steady increase in stockholder's equity. This points to an increase in S&P rise as well. We found both of these correlations by graphing the trends separately and then together to compare. Lastly, the biggest correlator was GVIP, or gross value of products. Essentially, this is GDP, or the total value of goods and services produced within a country in a given year. In this case, the heavy correlation between GVIP and S&P is intuitively justified since as production within a company increases, their stocks will increase as well. Due to this, we did not use GVIP in our prediction analyses since we knew that there was intrinsic data being carried over into the training set. This can lead to some hidden multicollinearity and other issues. This concludes our main trend analysis. Based on this, we created 4 different machine learning models to analyze/predict the stock market index for S&P: linear regression, LSTM deep learning

model, Facebook's prophet, and XGBoost.

2.1 Linear Regression

The linear regression model was mainly created as a baseline for analysis. We knew that for the scope of this project, linear regression would perform the worst, but kept it just as a baseline. We used 'EFFRVOL', 'date', and the two asset and liability series accounts as predictors for the linear regression model and used the Sci-kit learn library for machine learning analysis. The best predictor out of these was 'EFFRVOL', or the federal fund volume. We got a minimum MSE of 955.57.

2.2 LSTM Model

We found that the LSTM model produced the most consistent predictions results, and had the lowest RMSE score. The process of modeling starts with first processing the S&P 500 value we are predicting by creating a lagged version of the value. This lagged value will be the feature for our model. The lagged value is specified by a hyperparameter that best suits the model. We tuned these hyperparameters and found that the best performing model uses a timestep of 2 days, and a batch size of 5 (Fig. 6). After repeating the tests multiple times this model resulted in an average RMSE score of 166, which performs significantly better than the other models. We plotted the prediction against the validation set to visualize the results of our prediction (Fig. 5)

2.3 Other Models

The other models such as the XGBoost (Fig. 4) and the Facebook Prophet (Fig. 3) model performed poorly in our prediction. However this does not mean that these models are not good predictors of the S&P 500 value, we simply did not have enough time to perform further experiments on improving these models. Given more time we will be able to perform further tests that can improve the model's accuracies.

4. EXPERIMENTAL TESTING

We hypothesize that the LSTM deep learning model for predicting stock prices after 7 days would perform the best out of all the models and parameters. We were not able to perform extensive testing among the models due to our lack of time, however we were able to perform a hypothesis test of the model's performance among the different hyperparameters. We will utilize a one-tail t test to compare the average model scores (RMSE score) for the LSTM model's parameter combinations and check for a p-value of < 0.05 to check for significance. After hypothesis testing, we found that batch size was insignificant in its variation. There were no significant p values when comparing batch sizes. However, when comparing the timesteps, we found significant differences when comparing the smaller timesteps(2,3) to the larger ones(15,30,90). Due to the smaller p-values, this implies significance and shows that in this case of the LSTM model, smaller timesteps indicate lower RMSE and thus higher accuracies. We can justify this in the tables below.

5. CONCLUSION

The project explored different methodologies of predicting the value of the S&P 500, using the dataset provided by Vulcan Value Partners. We concluded that an LSTM model with the lowest timesteps performed best in predicting the values and reducing RMSE. However, we feel that the other models have a lot more potential in being more accurate if we had more time to improve them. We also believe that with more preprocessing of the data we can extract more meaningful features from this rich dataset. We are aware that we did not use the data to its full potential because of the lack of time to extract the features from the messy data. Due to our lack of knowledge with the stock market in general and lack of knowledge with processing such a complicated dataset we were unable to incorporate the other stocks and other measurements of the stock market that was provided in the dataset into our model.

Considering we had zero prior knowledge about time series forecasting and how to predict the stock market, we were able to apply the data analysis and machine learning skills that we have from the classroom into a real-world scenario project. We learned many things from time-series neural networks, gradient boosting, as well as how to handle time-series data. In the future, we will apply our learned knowledge into more time-series based projects and attempt to improve on our models to predict the stock market.

CITATION

[1] https://github.com/sathishmanthani/predict-sp500-index We referenced this project and got alot of our ideas from the methodologies mentioned and practiced in the project.
[2]https://towardsdatascience.com/lstm-time-series-forecasting-predicting-stock-prices-using-an-lstm-model-6223e9644a2f This article was referenced for us to learn about LSTM models and how they can be used in stock market prediction.
[3]https://towardsdatascience.com/time-series-forecasting-predicting-stock-prices-using-facebooks-prophet-model-9ee1657132b5 Guidelines on using Facebook Prophet that we referenced for our modeling.
[4]https://towardsdatascience.com/forecasting-stock-prices-using-xgboost-a-detailed-walk-through-7817c1ff536a We referenced this article to learn about XGBoost algorithms.

Visualizations

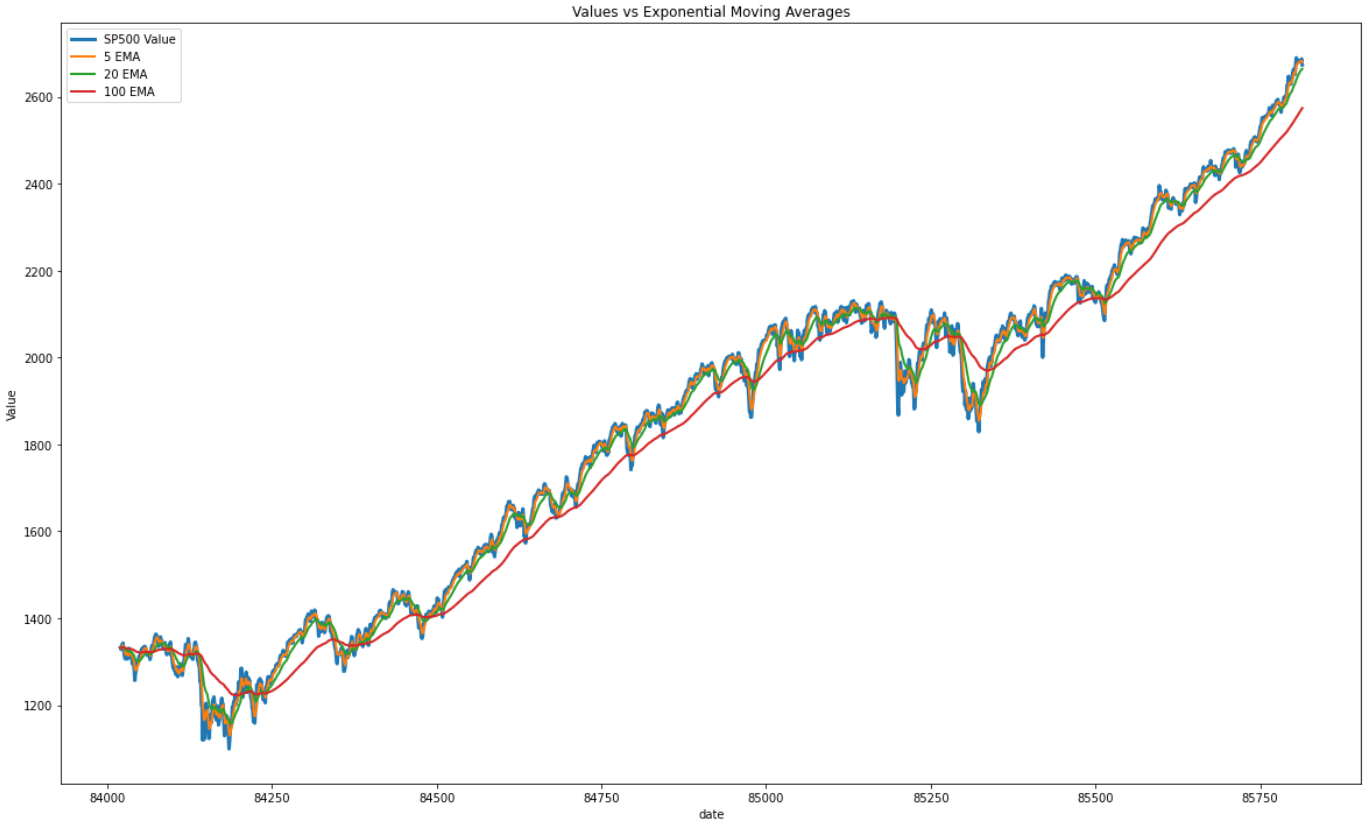

Fig 1. Exponential Moving Averages of S&P 500 Value

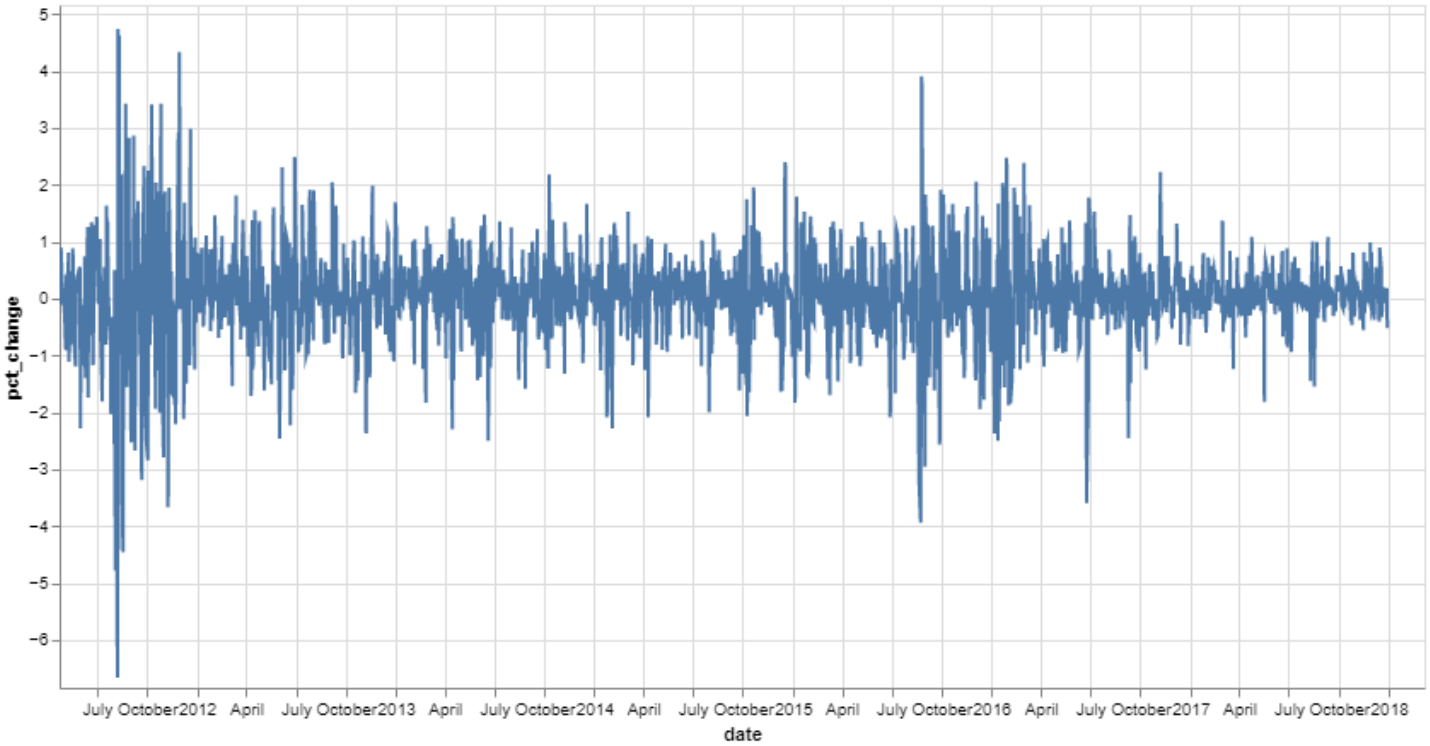

Fig 2. Daily Percent Change of S&P 500 Value Over Time

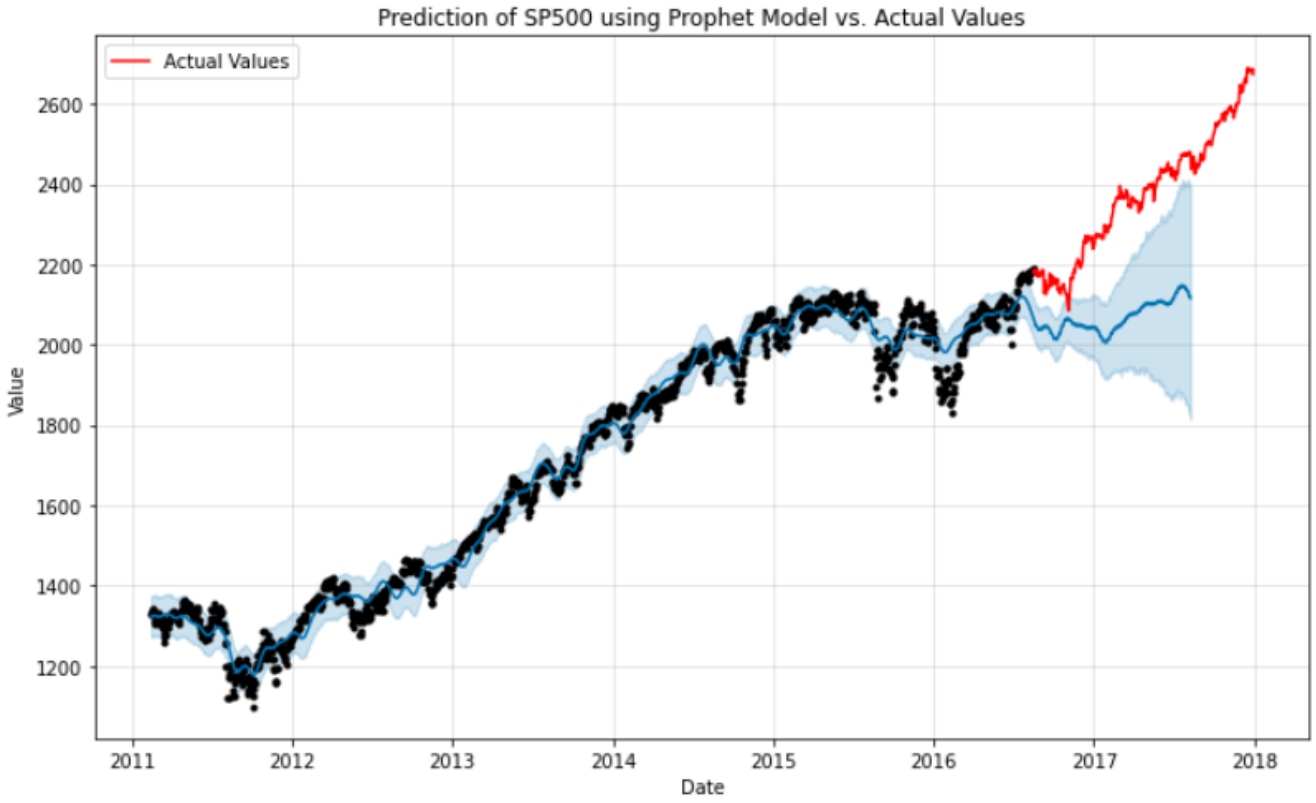

Fig 3. Prediction Line (Blue) of the Prophet Model Against Validation Set

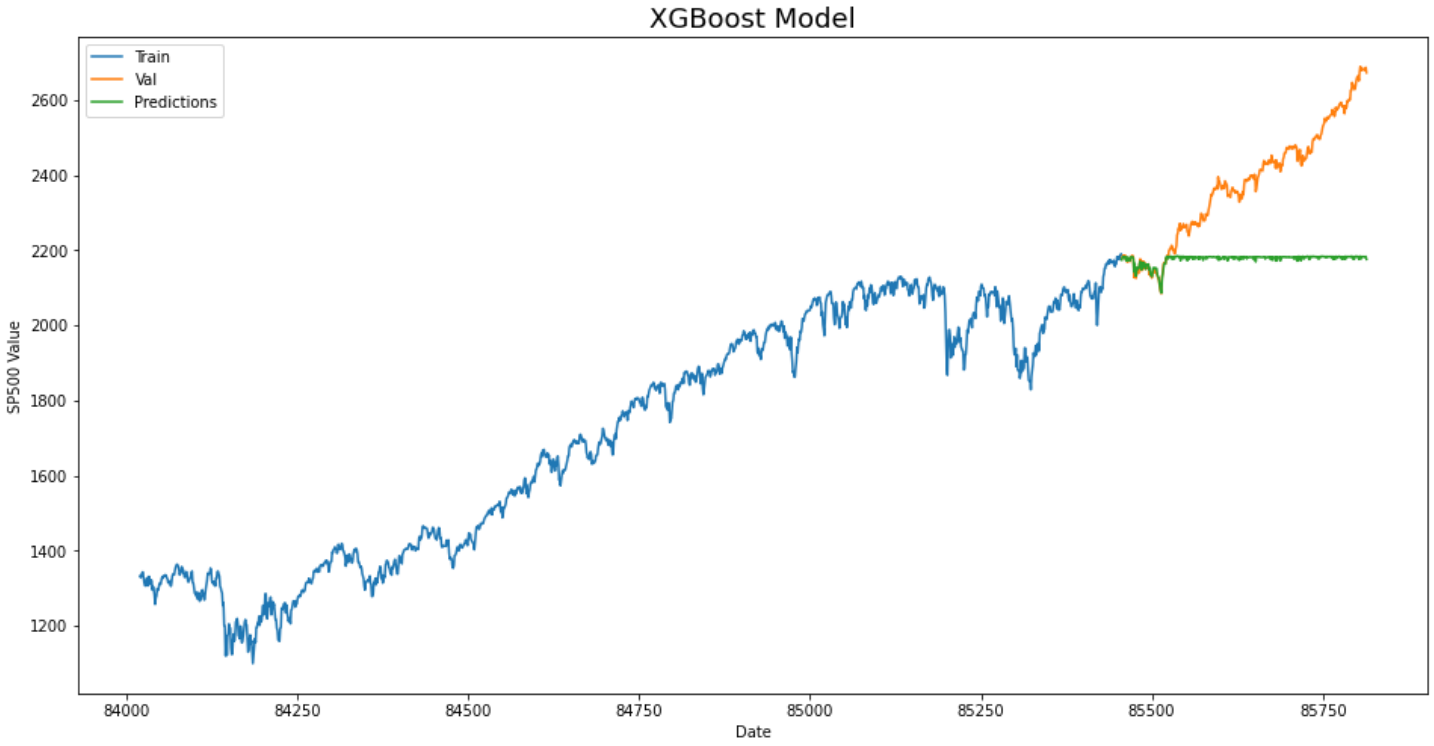

Fig 4. XGBoost Model Performance

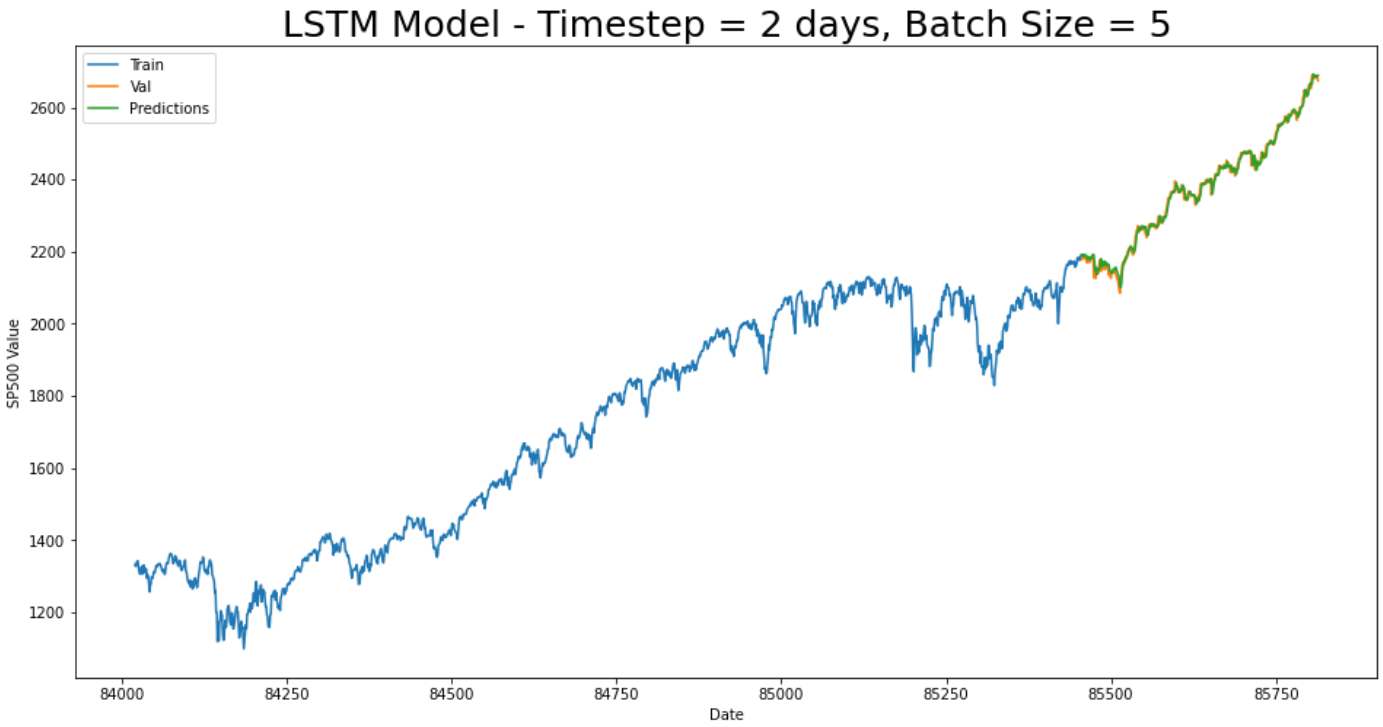

Fig 5. Best Performing Model against Validation Set

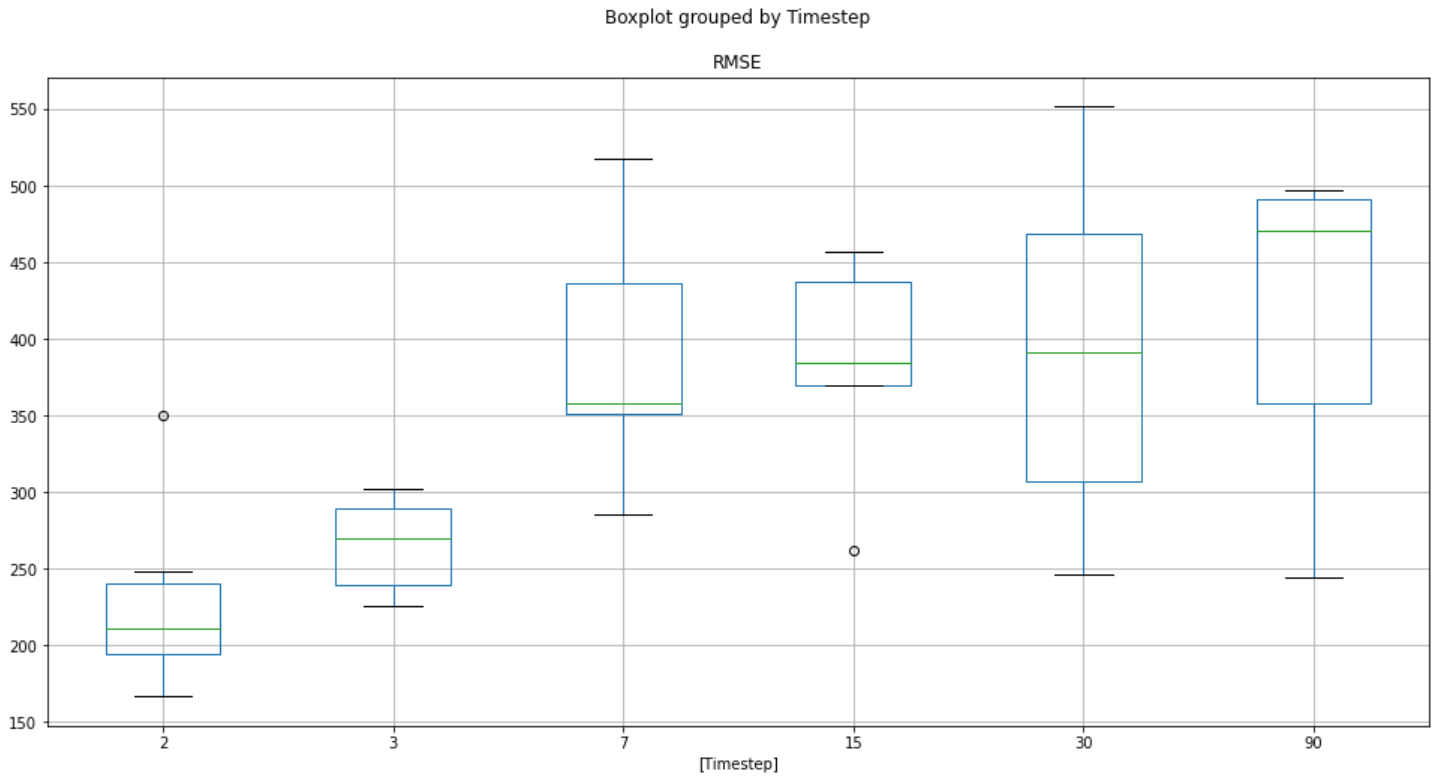

Fig 6. Boxplot of the LSTM Model Hyperparameter tuning RMSE results

Batch Size Comp.	t-stat	p-Value	Significant?
1 vs 2	-1.323	0.243	No
1 vs 3	-1.12	0.31	No
1 vs 5	-1.06	0.33	No
1 vs 7	-1.488	0.1968	No
1 vs 15	-1.69	0.15	No
2 vs 3	0.196	0.85	No
2 vs 5	-0.11	0.9138	No
2 vs 7	-0.556	0.6018	No
2 vs 15	-0.54	0.61	No
3 vs 5	-0.388	0.7137	No
3 vs 7	-1.0547	0.3398	No
3 vs 15	-1.453	0.2059	No
5 vs 7	-0.767	0.477	No
5 vs 15	-0.1	0.92	No
7 vs 15	-0.72	0.49	No

Timestep Comp	t-test	p-value	Significant?
2 vs 3	-1.437	0.21	No
2 vs 7	-3.01	0.02	Yes
2 vs 15	-3.15	0.025	Yes
2 vs 30	-2.37	0.063	No
2 vs 90	-2.84967	0.035	Yes
3 vs 7	-3.243	0.0228	Yes
3 vs 15	-3.534	0.016	Yes
3 vs 30	-2.1156	0.0879	No
3 vs 90	-2.989	0.03	Yes
7 vs 15	0.137	0.896	No
7 vs 30	-0.073	0.944	No
7 vs 90	-0.528	0.6196	No
15 vs 30	-0.167	0.873	No
15 vs 90	-0.843	0.437	No
30 vs 90	-0.726	0.5	No

Fig 7.Hypothesis T-testing Results, Batch Size and Timestep Comparison

Date	Predicted Value	Actual Value
Jan 10 - week	3242.5962	3265.35
Mar 6-week	3043.9468	2972.37
Jul 12 - 1 week	3242.5962	3013.77
Jan 10 - month	3242.7969	3265.35
Mar 6-month	3047.111	2972.37
Jul 12 - 1 month	2985.4553	3013.77
Jan 10 - year	3248.5598	3265.35
Mar 6-year	3044.2537	2972.37
Jul 12 - 1 year	2988.549	3013.77

Fig 8.Predicted Test Values from given time frames