Jestr

Journal of Engineering Science and Technology Review 13 (4) (2020) 117 - 123

JOURNAL OF
Engineering Science and Technology Review

Research Article

www.jestr.org

An ARIMA- LSTM Hybrid Model for Stock Market Prediction Using Live Data

Sakshi Kulshreshtha* and Vijayalakshmi A

School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, India

Received 31 May 2020; Accepted 25 June 2020

Abstract

The stock market is a highly volatile industry with ever changing bull (rise) and bear (fall) trends. This paper proposes a new hybrid model using Long Short- Term Memory (LSTM), a Recurrent Neural Network (RNN) technique and Auto Regressive Integrated Moving Average (ARIMA), a time series forecasting technique to capture the live stock market data of S&P 500 using preexisting Application Programming Interface (API). Rise and fall in stock values in the previous years is analyzed. A novel LSTM- ARIMA hybrid is designed for capturing the linear and non- linear portions of the time series. The Prophet forecasting library by Facebook has also been used that requires less preprocessing. Finally, both the approaches are compared and the one with better performance is accepted for the final stock market prediction system. In this case, Prophet has a high Root Mean Square Error (RMSE) of 27.59 and Mean Square Error (MSE) of 761.33 whereas the ARIMA- LSTM hybrid gives an MSE of 3.03 and RMSE of 1.74 along with a 99% fit of the model. Hence the hybrid performs much better than Prophet and is accepted as the final algorithm for implementation.

Keywords: time series forecasting; stock market prediction; S&P 500; live data; yfinance; ARIMA; auto regressive integrated moving average; LSTM; long short term memory; prophet

1. Introduction

The computerization of financial activities, connection through the internet and support of related software has drastically changed the way the stock market and financial operations are being implemented. It has experienced a drastic change in the way the operations are carried out.

Predicting the stock market is a challenging task, even for the people involved in the industry for many years due to its erraticity and unpredictability. Various factors like the current financial scenario, business environment and other physical and physiological factors along with the previous trends- all of these combine to make the share market highly volatile. This makes it difficult to predict the future market scenario with a high level of accuracy.

The solution lies in incorporating the machine learning and deep learning techniques into the stock market scenario to be able to better process and conduct analysis and visualizations of the stock value data. These algorithms tend to study the data, discover a pattern in the previous years' stock values, extrapolate the trend and predict the expected future stock market data to the user.

According to a research by Dev Shah et al. [1] people have implemented various approaches for stock market prediction. While many of them work only with a static historical data for market prediction, the other live data projects use features like price, volume, financial ratio, technical factors and Open, High, Low, Close and Volume (OHCLV) parameters, varying amongst them.

With varying datasets and algorithms like Support Vector Machine (SVM), Logistic Regression, Naïve Bayes and Random Forests, the accuracy of the project ranged from a meager 45.1% to a maximum of 86.2% only. RMSE, MSE and Mean Absolute Error (MAE) are some other suitable metrics for evaluating regression based problems. The various approaches have an RMSE of around 21 and MAE of 4.98 to 16.68 with a standard deviation of 3.7%.

In this paper, the S&P 500 stock index is used. The S&P 500 is a stock market index that quantifies the stock performance of top 500 large companies listed on the stock exchanges in the United States. It is one of the most commonly followed indices and is considered to be one of the best representations of the stock market. It includes companies like Adobe, Microsoft, PayPal, Amazon, Cisco, Ford and General Motors.

The main aim of this paper is to predict the future stock prices of the various shares of the stock market. The objectives are to capture the live stock market data from the APIs, create a novel ARIMA- LSTM hybrid for prediction, apply Prophet on the time series data, compare the performance of the two algorithms and predict expected market scenario in the future using the better algorithm.

It is found that the ARIMA- LSTM hybrid performs better than Prophet and gives an MSE of 3.03 and RMSE of 1.74 along with a 99% fit of the model whereas Prophet has a very high RMSE of 27.59 and MSE of 761.33. Finally a user interface is developed using the ARIMA- LSTM hybrid for hosting the prediction system based application.

2. Related Work

A recent work on the S&P 500 time series data by Min Wen et al. [2] applies convolution neural network on a

*E-mail address: sakshi.kulshreshtha2016@vitstudent.ac.in
ISSN: 1791-2377
doi:10.25103/jestr.134.11

reconstructed time series and uses motif extraction for macroscopic pattern discovery on the financial data. It renders an accuracy of around 56.14% with precision and recall of 55.44% and 74.75% respectively.

A hybrid model of exponential smoothing, ARIMA and backpropagation neural network (BPNN) is used by Creighton et al. [3] on the S&P 500 and S&P 400 daily closing index to complement for linear and nonlinear predictions. It uses the KoNstanz Information MinEr (KNIME) analytics platform and works well while predicting weekly data that has less noise and more linear growth but is unsuitable for daily predictions. The model gives a directional accuracy of 45.1%, MAE of 16.68, MSE of 434.121 and an RMSE of around 20.836.

Another model by Maini et al. [4] used Natural Language Processing for deriving content from new articles. It applied Random Forest along with SVM linear and Radial Basis Function (RBF) kernels to predict the stock market using Dow Jones Industrial Average, Reddit news data and the Guardian's news API. While Random Forest gave a better accuracy of 86.2%, SVM is also a good substitute for time series forecasting which gives 84.6% for linear kernel and 85.18% for RBF kernel.

A hybrid model was created by Hossain et al. [5] based on LSTM and Gated Recurrent Unit (GRU) that worked on the S&P historical time series data and made use of the yfinance API by Yahoo for extracting the data. The hybrid model gave MAE of around 0.023 and yielded much more accurate predictions when compared to the forecast of individual LSTM or GRU layer.

Shah et al. [6] did a comparative analysis of LSTM and Deep Neural Network (DNN) for forecasting the Bombay Stock Exchange (BSE) Sensex data. Both models were suitable for daily predictions and gave RMSE of around 1% but it was seen that LSTM was more suitable for carrying out predictions on a weekly basis.

A survey on stock market prediction systems was conducted by Iyer et al. [7] and a comparative analysis of the algorithms was done subsequently. Different datasets like National Stock Exchange (NSE) stock data, Hang Seng index, BSE data and Taiwan stock index were used for different approaches involving Fuzzy Systems, Artificial Neural Network (ANN), Bayesian algorithm and Sentiment Analysis. It was found that BPNN and Markov model with Decision Tree gave better performance and an accuracy of around 88%.

A novel stock price trend prediction system by Zhang et al. [8] predicts stock prices and growth rates and classifies stocks as up, down, flat or nil. It uses TA- Lib open source library for data and applies Random Forest, Imbalanced learning and Feature selection for trend prediction. It gives an accuracy of 67.5% with a standard deviation of 3.7% and is more suitable for predicting stock prices for the long term, i.e., for 30 to 40 days.

Selvin et al. [9] used three different algorithms- LSTM, RNN and Convolutional Neural Network (CNN) - Sliding Window model on the NSE data. Sliding window size was 100 minutes- 90 for information and 10 for prediction. It was found that CNN gives a better result than LSTM and RNN. The model gave MAE of 5.13% for RNN, 5.31% for LSTM and 4.98% for CNN. Deep learning outperformed ARIMA which has MAE of 29.87% as it is a linear model.
Financial forecasting of the Google stock price data by Persio et al. [10] used LSTM, GRU and Multilayer RNN to detect short term ups and downs in the time series. It also applied Adam optimization algorithm on it. The model gave an accuracy of 72% for a 5 day- period data prediction. Sadia et al. [11] applied Random Forest and SVM on a historical dataset from Kaggle for predictions. The data was trained on OHCLV, trade and value parameters and rendered an accuracy of 78.7% for SVM and 80.8% for Random forest.

Another approach by Deepak et al. [12] involved the application of SVM with RBF kernel on the BSE Sensex dataset to predict stock market scenarios for the next week, day and minute. The model gave an accuracy of about 80 to 85% depending upon the share considered. Tab. 1 summarizes the different approaches taken on various datasets for stock market prediction along with their accuracy and mean errors.

Table 1. Comparative Analysis of Various Prediction Techniques

Authors	Dataset	Technique	Evaluation Metric
Wen et al.	S&P 500	CNN using Motif extraction	Accuracy- 56.14% Precision- 55.44% Recall- 74.75%
Hossain et al.	S&P 500	LSTM- GRU hybrid	MSE- 0.00098 MAE- 0.023
Shah et al.	BSE Sensex	LSTM, DNN	RMSE- 1%
Iyer et al.	NSE, Hang Seng, etc	Fuzzy System, ANN, etc	Accuracy- 88%
Zhang et al.	TA- Lib	Random Forest	Accuracy- 67.5% Std deviation- 3.7%
Creighton et al.	S&P 500 and S&P 400	ARIMA- BPNN hybrid	MAE- 16.68 MSE- 434.121 RMSE- 20.836 Accuracy- 45.1%
Selvin et al.	NSE	RNN, LSTM, CNN	MAE- 5.13% (RNN) 5.31% (LSTM) 4.98% (CNN)
Sadia et al.	Kaggle Dataset	SVM, Random Forest	Accuracy- 78.7% (SVM) 80.8% (Random forest)
Maini et al.	Dow Jones Industrial Average	SVM, Random Forest	Accuracy- 84.6% (SVM- linear) 85.18% (SVM- RBF) 86.2% (Random forest)
Deepak et al.	BSE Sensex	SVM- RBF kernel	Accuracy- 80 to 85%
Persio et al.	Google Assets	RNN, LSTM, GRU	Accuracy- 72%

3. Proposed Methodology

This paper aims to capture the live stock market data from the source using preexisting APIs and analyze the rise and fall in stock values in the previous years. It will then predict the expected market scenario in the future using relevant machine learning algorithms for better accuracy. The project will focus on the technical analysis segment that includes doing a statistical analysis of the data, understanding the charts and identifying the trends in the stock market.

Two approaches have been used in the project for stock market prediction: a novel ARIMA- LSTM hybrid has been

designed which combines neural networks with time forecasting series for capturing the linear and non- linear portion of the time series. Another forecasting library called Prophet designed by Facebook has also been used that handles the missing data and outliers uses intuitive parameters for optimal predictions.

Finally, the algorithms are compared and the more accurate and less error prone algorithm is selected for future stock market prediction. The user can input the company's stock name whose predictions he wants to get. It will retrieve the stock's live data, apply the chosen machine learning algorithm, train the previous data and finally predict the expected future trend and stock values along with visualizations to the user.

The proposed methodology involves capturing the live stock market data using the yfinance API, developing the novel ARIMA- LSTM hybrid, applying Prophet on the time series data, comparing the two algorithms to find an optimal solution and finally deploy it for the stock market prediction system. The steps have been described in detail as given below:

3.1 Capture Live Stock Market Data

One of the major challenges of this work is to capture live data while maintaining the accuracy to be high as well. There are various stock market data APIs that offer real-time data on financial assets that are currently being traded in the market. It is possible to retrieve the current prices and historical data of the public stocks with the help of these APIs. They can help generate some indicators which are crucial for monitoring the market and building trading strategies. Some of the most recent APIs which are active in 2019 are Yahoo's yfinance API, Googlefinance, iexfinance and worldtradingdata.

This work uses the yfinance API of Yahoo to capture the live stock market data in OHCLV format. It gives the day's opening, closing, highest and lowest stock values along with the day's traded volume which is ultimately responsible for the volatility in the stock market. It is a standard API used by both individuals and enterprise level users as it provides reliable data of around the past 35 years and is easy and free to use.

3.2 Develop ARIMA- LSTM Hybrid Model

A statistic- based time series forecasting technique called ARIMA is used for predicting the linear part of the data and the nonlinear residuals will be tuned using LSTM. A novel hybrid model of the two algorithms will be designed for optimal predictions. It is expected to reach a decent accuracy capturing all the trends in the stock prices with an RMSE much better than that of existing systems that aim at forecasting the stock market.

ARIMA refers to Auto Regressive Integrated Moving Average. Here, Auto Regressive means that there is a changing variable that regresses on its own lagged or prior values. Integrated refers to differencing raw observations to allow the time series to become stationary. Moving Average signifies the dependency between an observation and a residual error from a moving average mode. The three models together help to capture the inherent trend in the market data.

To apply ARIMA on a given time series, it should be checked if it is stationary. The time series should be visualized and the necessary statistical analysis must be implemented on the data to ensure the following:

1) Mean of the time series should be constant, not a function of time.
2) Variance should not be a function of time.
3) Covariance of the i-th term and the (i+m)-th term should not be a function of time.

The above conditions can be ensured using the Dickey-Fuller test. If the time series is not stationary, trend and seasonality should be removed from it to make it stationary.

Trend can be removed from the time series in many ways like applying transformation functions like log functions, aggregation by taking weekly or monthly averages, smoothing by taking rolling average or by applying polynomial fitting on a give regression model.

For removing seasonality, two methods can be followed- either differencing with a particular time lag or decomposing the series by modeling trend and seasonality and removing them from the model. Differencing refers to the technique of taking the difference of the observation at a given instance of time with that at the previous instance.

ARIMA uses three ordered parameters p, d, q which take integer values to describe the model. p is the lag order or number of lag observations in the model. It can also be represented as the number of AR terms in the model. d is the degree of differencing or the number of times the raw observations are differenced. It is taken as 1 for ARIMA. Similarly, q is the number of MA terms in the model. It can alternatively be called as the order or size of the moving average window.

These three parameters together determine the performance of the ARIMA model and hence, must be chosen optimally. Once the three parameters are chosen, the ARIMA model can be built using them and final predictions can be made.

The residuals from the ARIMA model are trained using LSTM, which is an artificial recurrent neural network technique. It is used because it has comparatively more long term memory than RNN. Also, it helps mitigate the vanishing gradient problem which is commonly seen in neural networks.

It uses a series of gates contained in memory blocks which are connected through layers. The 3 types of gates are input gate which writes input to the cell; forget gate which reads output from the cell and output gate which resets the old cell value.

A number of hyperparameters need to be specified for the LSTM model to predict optimally. The number of layers should be 1 for simple problems, 2 for complex features and more than 2 layers make it harder to train the dataset. Dropout regularization can vary from 0.2 to 0.5 where higher dropout may lead to overfitting. Optimal batch size for training is same as the train dataset size and should be 1 for testing. Output activation should be linear for regression and softmax for classification problems. Also there are a variety of optimizers to choose from such as adaptive moment estimation based Adam optimizer, RmsProp and Stochastic Gradient Descent.

Time series data is composed of linear and nonlinear portion. Essentially, ARIMA model when used alone, only predicts the conditional mean. The reason being that ARIMA model is known to perform well on linear problems. The nonlinear portion or the residuals of the ARIMA model can be modeled using LSTM as it outperforms other models that can be used to model non linearity. So, the two models are consecutively combined to encompass both linear and nonlinear tendencies of the time series in the hybrid

ARIMA- LSTM model. Fig. 1 depicts the design of the proposed ARIMA- LSTM hybrid model.

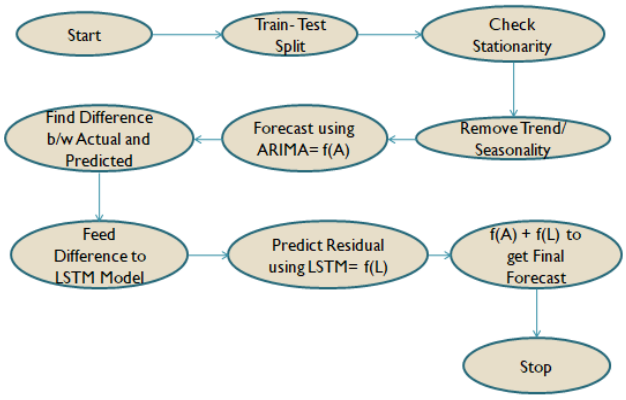

Fig. 1 Design of ARIMA- LSTM Hybrid

3.3 Develop Prediction Model using Prophet

Another algorithm that is used is the time series forecasting library designed by Facebook called Prophet. It is the only technique that covers all three aspects of time series, i.e. Trend, Season and Holiday aspects. It is one of the most recent technologies developed exclusively for predicting time series data and fits the future values quite optimally.

Prophet captures the trend by fitting a piece wise linear or logistic curve over the trend or non-periodic part of the time series. It uses Fourier series to provide a flexible model for tapping the yearly, weekly and daily seasonality. It also provides a custom list of holidays and events in the form of a built-in dataframe for each country with important holidays and their dates. There are other additional parameters too that model its effects on the data.

Prophet is based on simple intuitive parameters and is shown to give high accuracy as it also gives support to the impact of custom seasonality and holidays. It performs data preprocessing implicitly and handles the outliers well. Since it is easy to tune its parameters and also does not require expert level understanding to use it, it is emerging as a popular choice for time series forecasting.

3.4 Comparing the Models and Selecting the More Optimal One

The predictions from the two models, viz. ARIMA- LSTM hybrid and Prophet is finally tested on the stocks of the S&P 500 data through evaluation metrics like RMSE and MAE. The one with a better performance is finalized for further development of the system. The prediction system would be designed using web development techniques and the python code would be embedded using Flask framework in the prototype model. The detailed system architecture diagram of the proposed system is given in Fig. 2

4. Implementation

First the live stock market data needs to be retrieved from the yfinance API. The stock's ticker name (in this case, MSFT for Microsoft) and the period for which the data is needed have to be mentioned. The data is displayed in the form of past years' open, high, low, close, volume, dividends and stock splits as shown in Fig. 3. The retrieved stock data is adjusted using pandas_datareader library so that there is uniformity throughout the data, irrespective of any change due to stock splits or dispatching dividends.

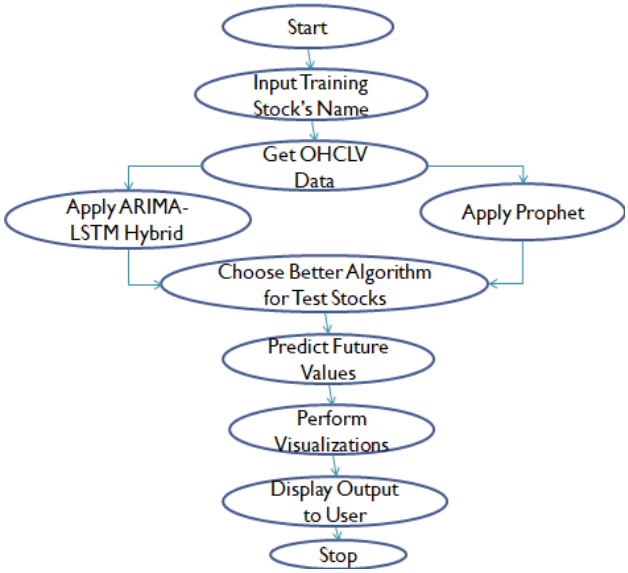

Fig. 2 System Design Flowchart

```
In [5]:  1 # get historical market data
         2 msft.history(period="max")
```

Out[5]:

Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
1986-03-13	0.06	0.07	0.06	0.06	1031788800	0.0	0.0
1986-03-14	0.06	0.07	0.06	0.06	308160000	0.0	0.0
1986-03-17	0.06	0.07	0.06	0.07	133171200	0.0	0.0
1986-03-18	0.07	0.07	0.06	0.06	67766400	0.0	0.0
1986-03-19	0.06	0.06	0.06	0.06	47894400	0.0	0.0
...	...	...	...	...	...	...	...
2020-01-13	161.76	163.31	161.26	163.28	21626500	0.0	0.0
2020-01-14	163.39	163.60	161.72	162.13	23477400	0.0	0.0
2020-01-15	162.62	163.94	162.57	163.18	21417900	0.0	0.0
2020-01-16	164.35	166.24	164.03	166.17	23865400	0.0	0.0
2020-01-17	167.42	167.47	165.43	167.10	34371700	0.0	0.0

8533 rows × 7 columns

Fig. 3 Retrieving Microsoft's live OHCLV data using yfinance API

Once the live stock market data is captured using the yfinance API, the further steps involve applying ARIMA on the time series to get residuals and using LSTM to predict the residuals. Consequently, the future stock prices are predicted using the hybrid. Then Prophet is applied on the time series data and the two algorithms are compared with respect to their performance. The better algorithm is finally deployed for the stock market prediction system. The steps have been described in detail as given below:

4.1 Applying ARIMA on the time series

Once the data is retrieved, the ARIMA- LSTM hybrid is created to capture the linear and non linear aspects of the time series data. For applying the ARIMA model to predict future values, the time series has to be visualized. It should be checked if it is stationary using the Dickey- Fuller test and made stationary if not already. Then the optimal parameters are found for building the model and finally start making predictions using the built model. The required steps have been described in detail in the following subsections:

4.1.1 Visualize the Time Series and Check if the Time Series is Stationary

To check if the time series is stationary, the Dickey- Fuller test should be applied. For that, it should be assumed that the null hypothesis represents that time series is not stationary and the alternative hypothesis is that time series is stationary.

If test statistic is less than critical value, the null hypothesis can be rejected, which means that the time series is stationary. Else, the null hypothesis has to be accepted.

Similarly if the p- value is less than 5%, the null hypothesis can be rejected. Else, the null hypothesis has to be accepted.

4.1.2 Make the Time Series Stationary

Every time series is composed of three components- trend, seasonality and residuals. Trend depicts the increasing or decreasing value in the series while seasonality refers to the repeating short-term cycle in the series. Noise or residual signifies the random variation in the time series.

To make the time series stationary, trend and seasonality should be removed from it. In this work, log transformation has been applied to remove trend from the time series. Seasonal decomposition has been used to remove seasonality from the data and hence, make the time series stationary.

4.1.3 Find optimal parameters p, d, q

ARIMA uses three ordered parameters p, d, q which together determine the performance of the ARIMA model and hence, must be chosen optimally. Auto-Correlation Function (ACF) is the measure of correlation between the time series with a lagged version of itself. Value where the ACF chart crosses upper confidence interval for the first time gives q.

Partial Auto-Correlation Function (PACF) gives the correlation between time series with a lagged version of itself but after eliminating variations which have already been explained. Value where the PACF chart crosses upper confidence interval for the first time gives p.

4.1.4 Build the ARIMA Model and Make Predictions

Finally after making the time series stationary and getting the optimal values for p, d and q, the ARIMA model is built. The model considers the stock values from the second last to the fifth last year as the training model and predict future stock values for the previous one year. Fig. 4 represents the actual vs predicted stock values using ARIMA model. The difference in the actual and predicted stock values, i.e. the residuals act as the training data for the LSTM part of the hybrid model.

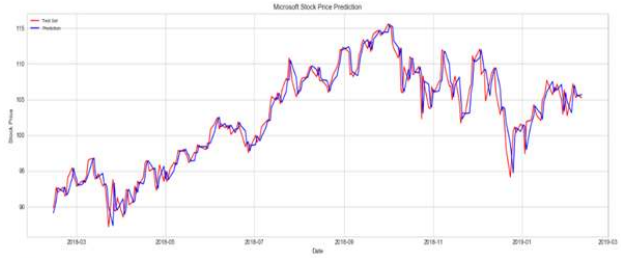

Fig. 4 Prediction using ARIMA to get Residuals

4.2 Applying LSTM Hybrid to the ARIMA Model

LSTM works well for the non- stationary portion of the data and also has a relatively longer memory. The residuals obtained from ARIMA are fed into the LSTM model and trained to tap a pattern and predict residuals for the upcoming year.

Various hyperparameters were tried to find the best fit for the prediction model. Two LSTM layers were used with 50 units each and a dropout of 0.2 at the end of each layer. Dropout regularization is a computational method to regularize a deep neural network. It probabilistically removes or drops out the inputs to a layer, which may be input variables in the data sample or activations from a previous layer. Adam optimizer was found to give the best prediction and linear activation was applied since it forms a regression problem. The model was initially run for 40 epochs but it stopped early at 11 epochs only as there was no further change in the mean squared error.

Once the data is trained, the residual values need to be predicted for the upcoming year using the designed LSTM model. Simultaneously, the ARIMA model is used to predict stock values for the upcoming year. Finally, the residuals retrieved from the LSTM model needs to be added to the predicted stock values retrieved from the ARIMA model. The sum obtained from ARIMA- LSTM hybrid gives the final predicted price of the stock for the upcoming year. Fig. 5 depicts the actual vs predicted stock price with the red line symbolizing the actual test dataset and the blue line representing the predicted values using the hybrid ARIMA-LSTM model.

Fig. 5 Prediction using ARIMA- LSTM Hybrid

4.3 Applying Prophet for Future Stock Prices Prediction

Prophet uses a decomposable time series model and can be used to check for the trend, seasonality and holiday effects in the time series data. It does not require extensive preprocessing and tuning of parameters and can also handle the outliers implicitly. This makes it easy to understand and a popular choice for time series prediction problems.

The stock market data of past 4 years is taken as the training set and the column names are accordingly altered to fit the Prophet model. Column name for closing price is changed to 'y' and that of Date is changed to 'ds'. Subsequently, the Prophet model is applied on the dataset to predict prices for the upcoming year.

Prophet model was applied on various stocks like Microsoft, Amazon, Apple, Google and Facebook and the predicted values were visualized through line graphs. Fig. 6 shows the predictions done for Microsoft's closing prices for the next year using Prophet.

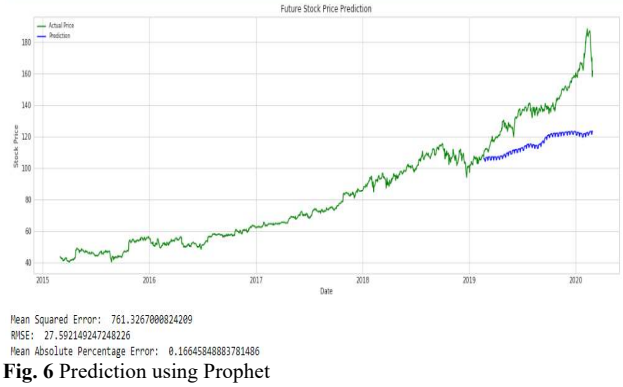

Fig. 6 Prediction using Prophet

4.4 Selecting the More Optimal Algorithm for Prediction

The two algorithms, viz. ARIMA- LSTM hybrid and Prophet are finally tested on the various stocks of the S&P 500 data. Evaluation metrics like RMSE and Mean Absolute Percentage Error (MAPE) have been used and the one with a better performance is finalized for the further development of the system.

It is found that ARIMA- LSTM hybrid performed much better than Prophet with a 99% fit of the model and an MAPE of 0.009 and a low RMSE of 1.74. Prophet has a very high RMSE of 27.59 and MSE of 761.33. So the final system will deploy the novel ARIMA- LSTM hybrid to predict the future stock market prices. The prediction system will be designed using web development techniques. The python code will be embedded using Flask framework in the prototype model.

Jupyter notebook with Python 3.6 is used for running the python scripts. Yahoo's yfinance API has been used for capturing live stock market data, pmdarima for implementing ARIMA based forecasting technique and keras for implementing LSTM. Theano has been used as backend of the LSTM model and fbprophet has been applied for implementing Prophet library by Facebook.

A web interface will be provided for the user where he can input the company's stock ticker name whose predictions he wants to check. It will then get the stock's live data, apply the chosen machine learning algorithm, train the previous years' data and predict the expected future trend and stock prices along with graph visualizations to the user.

5. Result

After developing ARIMA- LSTM hybrid and Prophet based prediction models, a comparative analysis is done between the two algorithms to select the final algorithm for predictions. Since it is a regression based problems, the evaluation metrics used here are RMSE, MSE, R- squared score and MAPE.

RMSE refers to the square root of the variance of residuals. It gives a relatively higher weight to significantly large errors. MSE measures the average of the squared differences between the actual and the predicted values. On the other hand, MAPE depicts the deviation of predicted value from the actual data in terms of percentage.

R- squared score is also referred to as the coefficient of determination. It is a statistical measure that determines how close the given data is to the fitted regression line. While RMSE, MSE and MAPE are absolute measures of fit, R-squared score is a relative measure. Lower the RMSE, MSE and MAPE, better is the prediction model. On the other hand, a higher R- squared score represents a better fit of the model.

ARIMA model alone gives an R- squared score of 93% and MAPE of 0.077 whereas the ARIMA- LSTM hybrid gives a much better fit with an R- squared score of 99% and MAPE of 0.009. The MSE and RMSE of both the models are almost similar with the hybrid performing a little better in both cases. It has an MSE of 3.03 and RMSE of 1.74. Hence it can be concluded that the hybrid performs much better than the individual ARIMA model.

On analyzing Prophet, it is found that it has a very high RMSE of 27.59 and MSE of 761.33. This is because Prophet captures only a linear trend in the data. While it may be the right choice for simple time series problems, it cannot tap patterns in volatile stock market data. Ultimately after comparing the models, it is finalized that ARIMA- LSTM hybrid wins over Prophet as it gives a good fit of around 99% and a low MAPE (0.009). The final prediction system will hence use the novel ARIMA- LSTM hybrid for the prediction of future prices of the stock market. Tab. 2 below gives a comparative analysis of the used algorithms and their evaluation metrics.

Table 2. Comparative Analysis of Algorithms

Algorithm	RMSE	MSE	MAPE	R- squared score
ARIMA	1.77	3.12	0.077	93%
ARIMA- LSTM Hybrid	1.74	3.03	0.009	99%
Prophet	27.59	761.33	0.166	74%

6. Conclusion and Future Work

Stock market predictions have been one of the deepest mysteries when it comes to predicting how the stock values would change in the future. In the proposed work, a novel ARIMA- LSTM hybrid is designed to perform time series forecasting on the volatile stock market data and predict future stock prices. Another library called Prophet by Facebook is also applied on the same stock market data. The primary purpose of this paper is to design a new hybrid model, compare its performance with the upcoming Prophet library and find out the more suitable machine learning algorithm for predicting the time series data. The results show that the ARIMA- LSTM hybrid outperforms Prophet in terms of MSE (3.03), RMSE (1.74) and MAPE (0.009) and has a fit of around 99%. The current work captures the trend in the data using technical and statistical aspects of the time series model using machine learning algorithms. For future work, one can also consider the effects of news and current world scenario on the stock market. An analysis can be done to show how the news module will affect the current ARIMA- LSTM model and its impact on the error rate in predictions. Also, hybrids with deep learning techniques like CNN can be applied to evaluate the performance of the prediction model.

Acknowledgement

We would like to acknowledge Yahoo for providing the yfinance API as open access to retrieve the historical as well as live OHCLV price data of the stock market.

3

References

1. Dev Shah, Haruna Isah, and Farhana H Zulkernine, "Stock Market Analysis: A Review and Taxonomy of Prediction Techniques", International Journal of Financial Studies, Vol. 7, No. 26, doi:10.3390/ijfs7020026 (2019).

2. Min Wen, Ping Li, Lingfei Zhang, and Yan Chen, "Stock Market Trend Prediction Using High-Order Information of Time Series", IEEE Access, Vol. 7, pp.28299-28308 (2019).

3. Jonathan Creighton, and Farhana H Zulkernine, “Towards Building a Hybrid Model for Predicting Stock Indexes”, IEEE International Conference on Big Data (2017).
4. Sahaj Singh Maini, and Govinda K, “Stock Market Prediction using Data Mining Techniques”, Proc. Int. Conf. Intelligent Sustainable Systems, IEEE Xplore Compliant - Part Number:CFP17M19-ART, pp.654-661 (2017).
5. Mohammad Asiful Hossain, Rezaul Karim, Ruppa Thulasiram, Neil D B Bruce, and Yang Wang, “Hybrid Deep Learning Model for Stock Price Prediction”, IEEE Symposium Series on Computational Intelligence SSCI, pp.1837-1844 (2018).
6. Dev Shah, Wesley Campbell, and Farhana H Zulkernine, “A Comparative Study of LSTM and DNN for Stock Market Forecasting”, IEEE International Conference on Big Data (2018).
7. Mohit Iyer, and Ritika Mehra, “A Survey on Stock Market Prediction”, IEEE 5th International Conference on Parallel, Distributed and Grid Computing, Solan, India, pp.663-668 (2018).
8. Jing Zhang, Shicheng Cui, Yan Xu, Qianmu Li, and Tao Li, “A Novel Data-driven Stock Price Trend Prediction System”, Elsevier Expert Systems with Applications, Vol. 97, pp.60–69 (2018).
9. Sreelekshmy Selvin, Vinayakumar R, Gopalakrishnan EA, Vijay Krishna Menon, and Soman KP, “Stock Price Prediction using LSTM, RNN and CNN- Sliding Window Model”, IEEE Conference Paper, doi: 10.1109/ICACCI.2017.8126078, pp.1643-1647 (2017).
10. Luca Di Persio, and Oleksandr Honchar, “Recurrent Neural Networks Approach to the Financial Forecast of Google Assets”, International Journal of Mathematics and Computers in Simulation, Vol. 11, pp.7-13 (2017).
11. K Hiba Sadia, Aditya Sharma, Adarrsh Paul, Sarmistha Padhi, and Saurav Sanyal, “Stock Market Prediction Using Machine Learning Algorithms”, International Journal of Engineering and Advanced Technology, Vol. 8, No. 4, pp.25-31 (2019).
12 Raut Sushrut Deepak, Shinde Isha Uday, and Dr D Malathi, “Machine Learning Approach in Stock Market Prediction”, International Journal of Pure and Applied Mathematics, Vol. 115, No. 8, pp.71-77 (2017).