Stock Market Trend Prediction Along with Twitter Sentiment Analysis

Priyadarshan Dhabe, Ayush Chandak, Om Deshpande, Pratik Fandade, Naman Chandak, and Yash Oswal

Abstract The Stock Market Prediction and Analysis has always been one of the most challenging tasks (Polamuri and Mohan in A survey on stock market prediction using machine learning techniques, 2019; Parmar et al. in First international conference on secure cyber computing and communication (ICSCCC), pp. 574–576, 2018). The variety of influences and unpredictability beats even the heavyweights to ground when it comes to successfully analyzing Stock Price data. In the proposed System, we have designed and successfully built a Machine Learning model using Long-Short Term Memory (LSTM) algorithm which helps for prediction of stock price data. We have done experimentations for better training, accuracy and results, on used data. The proposed system is also deployed on a web application which helps eliminate/reduce the difficulty of its use for the users. The model also works on the real-time data as we are using Yahoo finance API for getting updated data for model training and prediction. Lastly, The Indian stock market prices are also heavily driven by public sentiments which have for providing a better public opinion upon a particular stock. To help our users tackle this, we have added twitter sentiment analysis as a feature which provides us results in term of percentages of positive and negative sentiments within the tweets in the public domain at present about a particular stock, achieving a better opinion on a particular stock for the users.

P. Dhabe (✉) · A. Chandak · O. Deshpande · P. Fandade · N. Chandak · Y. Oswal
Department of Information Technology, Vishwakarma Institute of Technology, Pune, Maharashtra, India
e-mail: priyadarshan.dhabe@vit.edu

A. Chandak
e-mail: ayush.chandak18@vit.edu

O. Deshpande
e-mail: om.deshpande18@vit.edu

P. Fandade
e-mail: pratik.fandade18@vit.edu

N. Chandak
e-mail: naman.chandak18@vit.edu

Y. Oswal
e-mail: yash.oswal18@vit.edu

V. E. Balas et al. (eds.), *Intelligent Computing and Networking*, Lecture Notes in Networks and Systems 632, https://doi.org/10.1007/978-981-99-0071-8_5

The resulting model successfully gives us a prediction graphs as an output when given a particular stock on the proposed web application. We obtained least error in prediction, for Asian Paints data for the split of 80:20, using 75 epochs.

Keywords Stock market prediction · LSTM · Yahoo finance · Sentiments · Twitter

1 Introduction

For business analysts and researchers, forecasting the stock marketplace rate is usually a task. Stock market costs estimation isn't most effective, a thrilling however additionally tough vicinity of studies [1, 2]. Predicting the stock market with complete accuracy may be very tough as external entities such as social, mental, political, and financial have a top notch and large impact on it. The characteristic of the data related to the stock market is commonly time version and nonlinear. Prediction of inventory marketplace performs a crucial position in stock enterprise. If traders lack enough information and knowledge, then their funding can go through the greatest loss. Traders must expect the destiny stock fee of agencies a good way to attain excessive earnings. Diverse prediction techniques were developed to do predictions on the stock market as it should be. This model considers a company's former equity share value and uses the RNN method called as LSTM. The data set was obtained directly from Yahoo Finance. The proposed approach uses a share's historical data and makes predictions on a given attribute. Attributes of a share can be day high, day low, opening price, prior day opening and close price, day of trading, total trade quantity and turnover are all characteristics of shares. The said model apply time series analysis to foretell the share price over a given time span. Long Short-Term Memory (LSTM) is one of many forms of Recurrent Neural Networks (RNN) that can capture input from previous stages and use it to predict the future. Given the effect of social media on our daily lives, understanding public sentiment of a stock market company through various social media platforms has become a need in today's world. It's crucial to track public opinion while making a decision with respect to them and deciding the next step. For doing the same, social networking websites are a good place to start. Twitter is popular platforms for candid public sentiment analysis on a variety of topics. With the use of Tweepy, TextBlob, and Data Frame by pandas this study intends to examine public mood via Tweets from Twitter of any particular Stock in recent time. Then Later indicate the positivity, negativity, and neutrality of a tweet based on the Polarity score and visualize the data to gain a clearer picture of the attitude that dominates. This research project seeks to analyze social media data, with a focus on Twitter, in order to compute sentiment scores and depict them, with the goal of explaining people's social media sentiment of any particular stock or a listed company.

2 Literature Survey

In the work [3], researchers used KNN and nonlinear regression for stock price prediction. They used data of 6 companies from Jordanian stock exchange to help all the stakeholders. Their experience says that KNN is robust, reasonable and coherent method in this case. It also gives small error ratio. Mehtab and Sen [4] have shown that how LSTM can be employed in practice for foretelling stock prices of NIFTY 50 changes on the National Stock Exchange (NSE) of India. It is among the most recent ideas in this work domain. The authors created three forecasting models using daily stock prices. The models' prediction accuracies were then assessed due to their ability to predict the perturbation patterns of NIFTY index's nearer value over a one-week time horizon. Authors used NIFTY, fifty index values in the span, Jan 2018 to Jun 2019, for testing [4]. The work proposed in [5], uses Convolutional Neural Networks (CNN) and multivariate time series data for stock prediction. The authors' suggested prediction model combines a CNN's learning capacity with prediction validation to achieve a high degree of accuracy in expecting future index values of NIFTY and the trend in movement. The authors offer three distinct CNN designs, each with a different number of variables utilized in predicting, count of systems sub-models and input data size for model training. The CNN-based multivariate foretelling prototype was considerably worthy in predicting the weekly movement of NIFTY index values, according to the experimental data. LSTM networks have also been proposed for stock price prediction. Use of LSTM [9] in stock price foretelling is described in [5]. In the study [6], the comprehensive procedure of evolving a stock price foretelling model utilizing the ARIMA model is given. A stock price prediction algorithm is integrated with available data from New York Stock Exchange (NYSE) and the Nigerian Stock Exchange (NSE). The results showed that the ARIMA model has a lot of potential for largely short-term prediction and might compete well with conventional stock price prediction strategies. This can guide stock market investors for taking profitable investment decisions. ARIMA models may compete quite well with developing forecasting approaches in short-span foretelling based on the findings obtained [6]. This paper [7] has proposed a hybrid model that combines the benefits of a CNN and a LSTM approach they talk about the different works related to pattern reading and prediction, providing us the comprehensive view about the prediction techniques. We also studied the way they combined the models which is interesting and inspirational for our approach towards our model.

The main outcome of this study is, suggested CNN-LSTM model beat 17 baseline time series forecasting algorithms for test as well as foretelling data, along with least average values of RMSE, MAPE and RRMSE. Finally, while individual CNN and LSTM models predict verified COVID-19 occurrences time series well and efficiently, combining these two models in projected CNN-LSTM encoder decoder structure greatly increases performance of forecasting. In addition, the suggested model exhibited acceptable predication demonstrated that the suggested model produced acceptable predicting results with limited of Date 2022-01-14 Words 744 Characters 5121 Page 1 of 2 data was available. This proposed technique has helped achieve

improved accuracy for the COVID-19 cases prediction, and we may apply it to our stock market prediction model as well, but we will have to careful at the same time honoring the exclusivity of both the applications [7].

3 Dataset

3.1 Yahoo Finance

Collection of data is most crucial task in the research. The dataset is collected from Yahoo finance which affords monetary news, statistics and statement together with stock charges, press releases, monetary reports, and unique content. This dataset is perfect as you can view historic price, dividend, and cut up information for most quotes in Yahoo Finance to forecast the destiny of an organization or advantage marketplace perception (Fig. 1).

3.2 Twitter Sentiment Analysis

Twitter is a large dataset, for working with sentiment evaluation using twitter statistics [8, 10]. The statistics extraction is critical. Twitter gives access to tweets using their APIs. The data set accumulated from Twitter API to apply tweets sentiment evaluation for the statistics is collected as positive, negative or neutral tweets. This is done with the help of polarity analyzer this is a technique of identifying attitudes in textual content statistics about a subject of interest. Its miles scored the use of polarity values that variety from 1 to −1. Values toward 1 suggest more positivity, while values closer to −1 indicate extra negativity [8] (Fig. 2).

Fig. 1 Price of stock (dataset) snap

	High	Low	Open	Close	Volume
0	7.619643	7.520000	7.611786	7.526071	352410800.0
1	7.660714	7.585000	7.622500	7.643214	493729600.0
2	7.699643	7.616071	7.664286	7.656429	601904800.0
3	7.686786	7.526786	7.656429	7.534643	552160000.0
4	7.571429	7.466071	7.562500	7.520714	477131200.0
...	...	...	...	...	...
2009	43.855000	43.625000	43.669998	43.752499	65397600.0
2010	42.867500	42.419998	42.700001	42.642502	132742000.0
2011	42.695000	42.427502	42.525002	42.650002	85992800.0
2012	42.962502	42.619999	42.750000	42.770000	65920800.0
2013	42.647499	42.305000	42.630001	42.307499	103999600.0

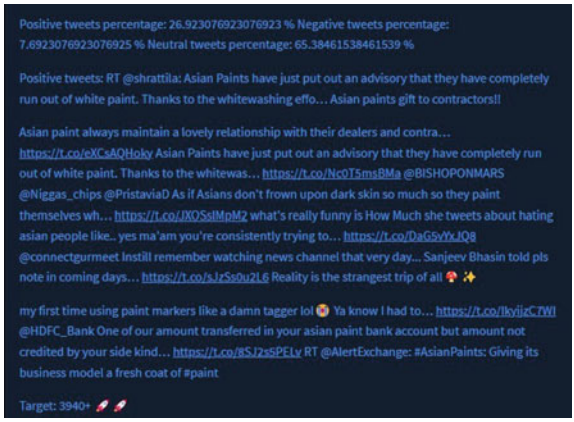

Fig. 2 A sample tweet

3.3 Standardization

Standardization is the data transformation with respect to mean value and scaling it dividing their variance. Post standardization mean and the variance becomes 0 and 1, respectively. Standardization helps to improve model performance too.

The standardization (Z) formula is as given in (1) for N samples.

$$Z = \frac{x - \mu}{\sigma} \tag{1}$$

where mean and variance are computed as in (2) and (3), respectively.

$$\mu = \frac{1}{N} \sum_{i=1}^{N} (x_i) \tag{2}$$

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_i - \mu)^2} \tag{3}$$

4 Work Flow Diagram

Following Fig. 3 indicates the flow diagram of trend prediction and Fig. 4 shows planned work flow of twitter analysis.

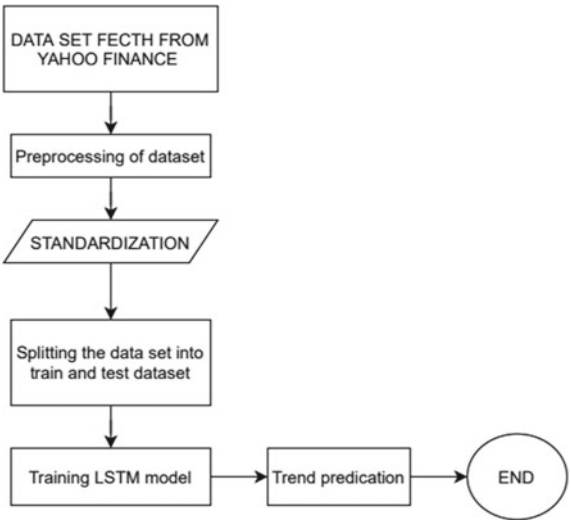

Fig. 3 Trend prediction work flow diagram

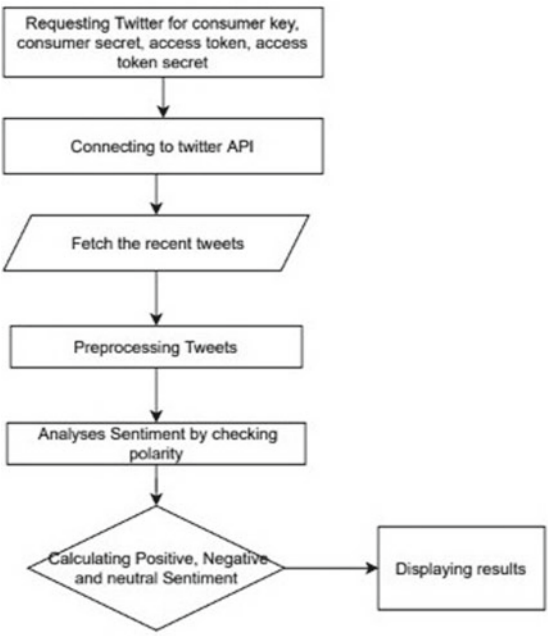

Fig. 4 Twitter analysis workflow diagram

5 Methodology

5.1 Stock Trend Predication

Data Splitting

The dataset is divided in to disjoint training and testing sets. The model learns with supervised training set and reason for the test set. Test dataset is used to evaluate the accuracy of our model's predictions. We used, both, 70–30% and 80–20% data split for the experimentation.

Long Short-Term Memory (LSTM)

Given the problem statement, we understand that prediction in the stock market take place using pattern creation and iterations of those patterns historically and serving the exact purpose we have the "Real-Time Recurrent Learning" (RTRL), but when looked throughout, it has a major issue which has to be addressed.

The Conventional RTRL when dealing with errors which go backwards in time either leads to oscillations of weights or learning to bridge up a gap takes a lot of time lag, which hints us to the solution and the used model i.e., LSTM.

The LSTM is a kind of recurrent network in conjecture with the gradient-based learning. The original developed model promised to keep short-term memory as long as 1000 consecutive inputs. One of the important components which contribute majorly to the Architecture is Memory Cells and Gate Units. To eliminate the possibility of perturbation by irrelevant inputs, multiple input gate units and output gate units have been introduced expanding the constant error carrousel, result is a more complex unit know as a memory cell. Given below in Fig. 5, is the figure which represents the memory cell.

The Memory Cell has 3 doors to be generalised, entrance, door with a view and an info door. This cell collects data at appropriate set timings which serves as the long short-term memory of our model [9]. A significant thing to understand and decrypt is that overhead door has both the responsibility to loads and capacity to start up the state cell. Also, Memory from the past cell can be allowed to pass as it is, rather than expanding and decreasing exponentially at each layer of network, and loads can have their ideal quality as quick as possible. This also solves issue-as a value put in cell is not adjusted every time, the inclination will not be hampered towards our Indian trading entities i.e. NSE and BSE (Fig. 6).

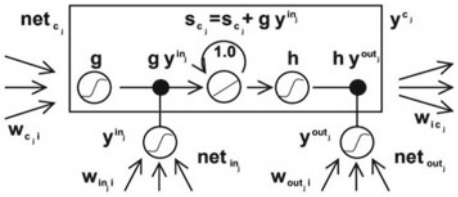

Fig. 5 A memory cell of LSTM, referred from the context of "LSTM by *Sepp Hoch Reiter & Jurgen Schmid Huber*"

Fig. 6 LSTM model

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 100, 50)           10400
_________________________________________________________________
dropout (Dropout)            (None, 100, 50)           0
_________________________________________________________________
lstm_1 (LSTM)                (None, 100, 60)           26640
_________________________________________________________________
dropout_1 (Dropout)          (None, 100, 60)           0
_________________________________________________________________
lstm_2 (LSTM)                (None, 100, 80)           45120
_________________________________________________________________
dropout_2 (Dropout)          (None, 100, 80)           0
_________________________________________________________________
lstm_3 (LSTM)                (None, 120)               96480
_________________________________________________________________
dropout_3 (Dropout)          (None, 120)               0
_________________________________________________________________
dense (Dense)                (None, 1)                 121
=================================================================
Total params: 178,761
Trainable params: 178,761
Non-trainable params: 0
_________________________________________________________________
```

5.2 *Twitter Sentiment Analysis*

Tweepy will be used to extract data in order to perform sentiment analysis. Tweepy is a Python module that allows you to access the Twitter API, which allows you to extract and access data on a range of topics. To use the Twitter API, you must first create a developer account and get it accepted by twitter through an application procedure. Once we obtain the access to it and have authorisation for a developer account, we generate consumer tokens and access tokens and define them as variables. We also utilise OAuthHandler and set_access_token to check the access tokens and authenticate the account. The same is demonstrated in Fig. 7.

The tweets search function is then used to scrape tweets, and we further search for tweets using a certain hashtag, language, and time. We must also specify the number of tweets to be extracted. When tweets are successfully extracted, they are stored in a Data Frame and labelled appropriately.

Fig. 7 LSTM model implementation

```
# keys and tokens from the Twitter Dev Console

consumer_key='Py0nJPLCN0F7Z3WNHtPCKgAxl'
consumer_secret='uMp5Ls2FzNiwPsImGMyPZARwzaCxIcsPGtClYsyUvC2xqnJdFD'
access_token='3553689916-HEuAPbMQxVOCPsQqHmyZgv8GJKKNH0MmLxH5uyh'
access_token_secret='aIrPqZVLzV1sabwRhNzHUqCnlyEkPEAbsoOSxzHjgltUm'

# attempt authentication
try:
    # create OAuthHandler object
    self.auth = OAuthHandler(consumer_key, consumer_secret)
    # set access token and secret
    self.auth.set_access_token(access_token, access_token_secret)
    # create tweepy API object to fetch tweets
    self.api = tweepy.API(self.auth)
except:
    print("Error: Authentication Failed")
```

We begin cleaning the tweets after the data has been scraped and placed in a data frame. Manipulation of any type of textual data should be approached with caution, as changing incorrect data might lead to biased analysis and, ultimately, false results. We then do data purification to remove various symbols and usernames from the tweets in order to ensure more accurate sentiment analysis computations.

Following data cleansing, we employ Text Blob's sentiment function to compute Subjectivity and Polarity scores, which are critical for classifying the extracted tweets.

- Polarity is a float value between −1 and 1 that indicates whether a text is positive or negative. In a nutshell, the Polarity score aids in the analysis of a text's emotion or attitude.
- Subjectivity is a float value between 0 and 1 that determines whether a text is more subjective or objective. A subjective sentence is a piece of literature with a tone that leans more towards an opinionated expression. An objective sentence is a type of textual material with a tone that favours factual expressions. The Polarity score is also used to categorise tweets into positive (> = 0), neutral (=0), and negative (=0) categories. By categorising tweets into the supplied tags, we visualise our final results in order to evaluate trends from our textual data [10].

6 Experimental Results

After training the model, result of our testing has shown different results with number of epochs and train-test split. The data is of stock ASIAN PAINTS from "1/1/2010" to "1/12/2021" with total of 2939. The model is trained and test with 2 combinations of spilt i.e., 70:30 (train: test) and 80:20 (train: test).

To evaluate model, we use Root Mean Square Error (RMSE), which is a de-facto way to measure error in predicting data. It is defined in (4) for n values.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}} \tag{4}$$

where $\hat{y}_i$ represents the predicted values and y_i are test values for $i = 1, 2, ..., n$ (Fig. 8).

In Table 1 the RSME of predicate value is given with respect to epochs. First with split of 70% (2057) train and 30% (882) test we can observe that with increasing no. of epochs, the RSME first decreases than increases. This is because of overfitting model.

When a model is trained with a large amount of data, it begins to learn from the noise and inaccuracies in the data set. The model then fails to predicate the input due to too many details and noise. This is known as Overfitting of model (Fig. 9).

In Table 2 the RSME of predicate value is given with respect to epochs. First with split of 80% (2352 samples) train and 20% (587 samples) test we can observe

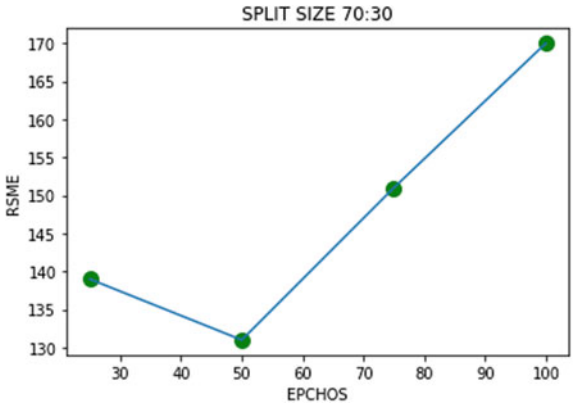

Fig. 8 Epochs versus RSME

Table 1 Epochs versus RSME versus time

Split 70:30 (samples 2057:882)		
Epochs	RSME	Time taken to run epochs
100	170.10	2030
75	151.5	2325
50	131.43	1120
25	139.02	375

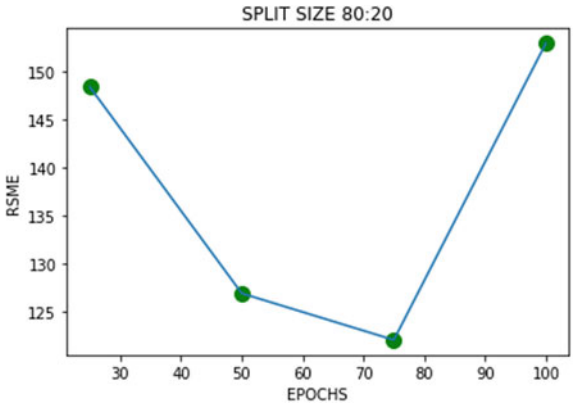

Fig. 9 Epochs versus RSME

that with increasing no. of epochs, the RSME decreases drastically. In Epoch-75 we achieved lowest RSME value (122.42).

With the use of numerous charts imported from Matplotlib, various trends and conclusions are presented based on the sentiment computations that have been calculated. The retrieved texts are tokenized, and the words with the highest usage frequency are presented in the form of a word cloud.

Table 2 Epochs versus RSME versus time

Split 80:20 (samples 2352:587)		
Epochs	RSME	Time taken to run epochs
100	152.89	2360
75	122.42	2550
50	126.85	1050
25	148.35	540

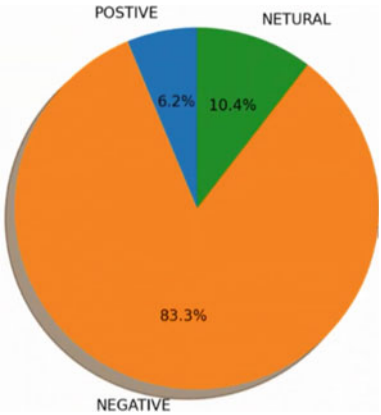

Fig. 10 Visual representation of classification of tweets

As previously stated, the Subjectivity score is a number between 0 and 1 that defines if a tweet is more opinionated or factual, and the Polarity score is a number between −1 and 1 that classifies a tag as positive, negative, or zero.

Figure 10 depicts the distribution of various tweets depending on the count of tweets based on emotion tags assigned by Polarity scores. The visualization pie-chart reveals a nearly equal distribution of neutral and negative tweets, with a significant number of positive tweets. We may make a major conclusion from this, namely that while there appeared to large negative tweets as well as positive tweets, indicating people's interest.

The trend predicated form test data set with respect to time and different epochs and split are plot below. Figures 11, 12, 13 and 14 are for the split of 70–30% and Figs. 15, 16, 17 and 18 are for splits of 80–20%.

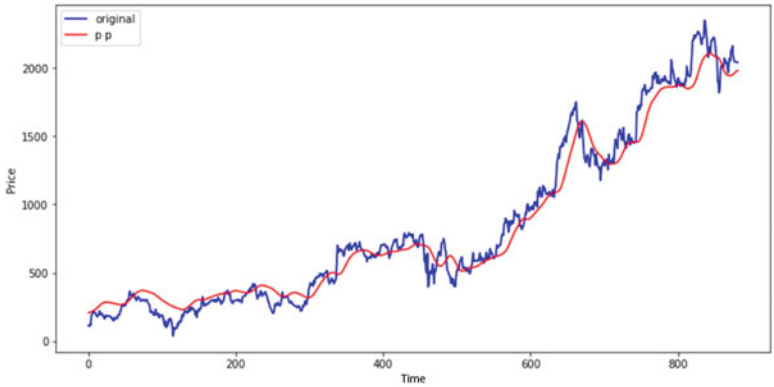

Fig. 11 Price versus time for 25 Epochs (split 70–30)

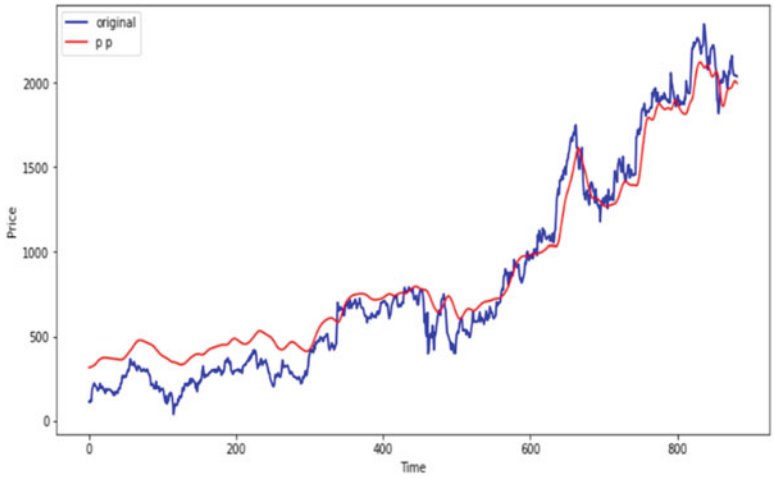

Fig. 12 Price versus time for 50 epochs (split 70–30)

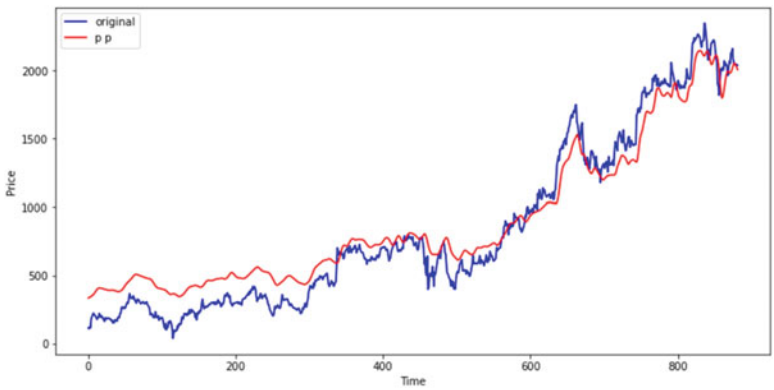

Fig. 13 Price versus time for 75 epochs (split 70–30)

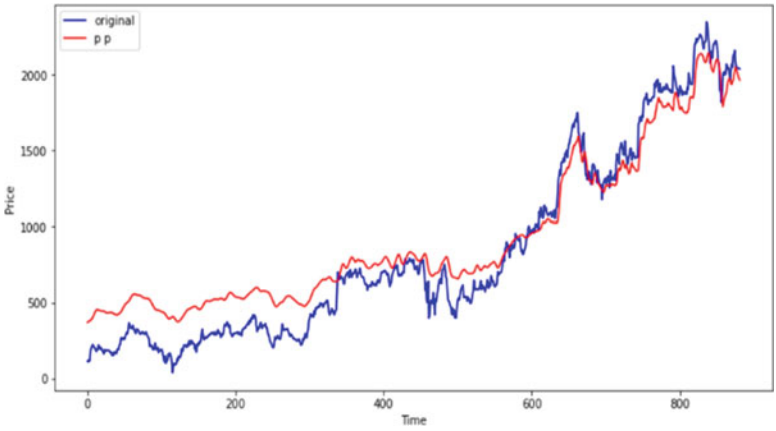

Fig. 14 Price versus time for 100 epochs (split 70–30)

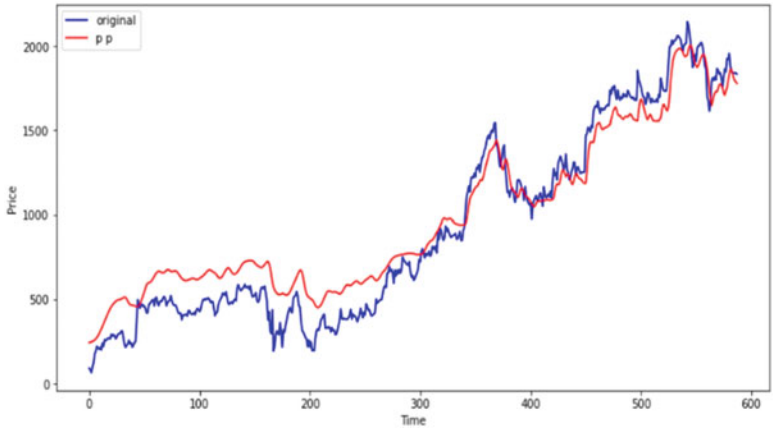

Fig. 15 Price versus time for 25 epochs (split 80–20)

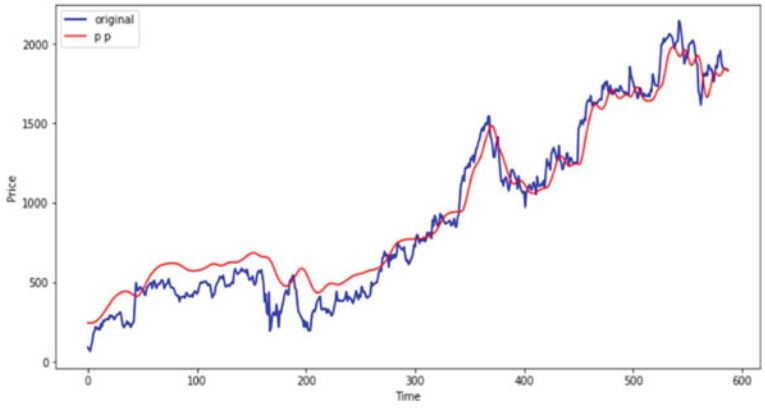

Fig. 16 Price versus time for 50 epochs (splits 80–20)

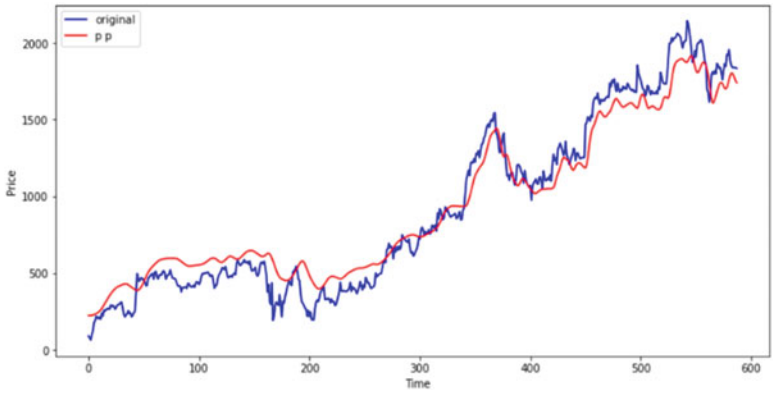

Fig. 17 Price versus time for 75 epochs (split 80–20)

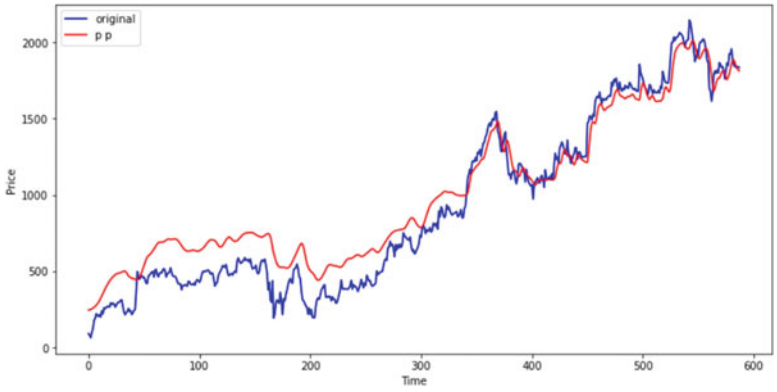

Fig. 18 Price versus time for 100 epochs (split 80–20)

7 Conclusion

This paper proposes LSTM model built to forecast future trend of STOCK and Sentiment Analysis model for determining the sentiment of asset trough twitter data feed. Model achieved the lowest RMSE in 75 Epochs with 80:20 (test: train) split of data set. Hence, we conclude that 75 epochs with 80:20 split predicate best trend predication of price with lowest error rate.

This can assist investors in gaining a significant financial benefit while maintaining a stable stock market environment. Investor can avoid huge draw down in finical market by using models in significant manner. In future work multiverse time series can be done by adding important feature in data set which will further improve the model.

Acknowledgements Our student's team would like to thank the Director, VIT, Pune, Prof. Rajesh Jalnekar, HOD IT Prof. Ghadekar and Associate Professor Dr. Priyadarshan Dhabe, for their continuous support, guidance and inspiration for the research work.

References

1. Polamuri SR, Mohan AK (2019) A survey on stock market prediction using machine learning techniques. ICDSMLA 2019, pp 923–931
2. Parmar et al (2018) Stock market prediction using machine learning. In: 2018 first international conference on secure cyber computing and communication (ICSCCC), pp 574–576. https://doi.org/10.1109/ICSCCC.2018.8703332
3. Stock price prediction using K-nearest neighbor (kNN) algorithm—scientific figure on ResearchGate. https://www.researchgate.net/figure/1-The-results-after-applying-kNN-algorithm-for-the-AIEI_tbl3_262456253. Accessed 14 Jan 2022 (L1)
4. Mehtab S, Sen J (2019) A robust predictive model for stock price prediction using deep learning and natural language processing: presentation. https://doi.org/10.13140/RG.2.2.32046.66883 (L2)
5. Mehtab S, Sen J (2020) Stock price prediction using convolutional neural networks on a multivariate timeseries (L3)
6. Adebiyi A, Adewumi A, Ayo C (2014) Stock price prediction using the ARIMA model. In: Proceedings—UKSim-AMSS 16th international conference on computer modelling and simulation, UKSim 2014. https://doi.org/10.1109/UKSim.2014.67 (L4)
7. Lu J, Zhang Q, Yang Z, Tu M (2019) A hybrid model based on convolutional neural network and long short-term memory for short-term load forecasting 1–5. https://doi.org/10.1109/PESGM40551.2019.8973549 (L5)
8. Balachander PSJB (2020) Sentimental analysis of Twitter data using Tweepy and Textblob. IJAST 29(3):6537–6544
9. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9:1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
10. Sarlan A, Nadam C, Basri S (2014) Twitter sentiment analysis 212–216. https://doi.org/10.1109/ICIMU.2014.7066632