Trend Analysis of the Stock Market from Historical Data Using Hybrid Neural Network

1st Amit Dutta
Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
amit.dutta@g.bracu.ac.bd

2nd Aminul Islam Anik
Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
aminul.islam.anik@g.bracu.ac.bd

3rd Adnan Karim
Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
adnan.karim@g.bracu.ac.bd

4th Annajiat Alim Rasel
Senior Lecturer
Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
annajiat@bracu.ac.bd

***Abstract*—Stock market price prediction is a much-explored topic in recent years. Many factors determine changes in stock price. The stock market is also a very complex and unorganized place as there is a lot of data and information and there is confusion about which data to use and which not. The stock market can also be very unstable and volatile as it is subject to change in a very short span of time. Traditional approaches used either neural networks or time series analysis algorithms to predict the market trend. The problem with the traditional approach is that the trend generated by the neural network tends to linearize after a while and on the other hand time series analysis algorithms tend to generate a threshold or range rather than something concrete. This is why in this approach we tried to combine them to achieve better outcomes. In this paper, we have collected historical data of the stock market and trained them using a hybrid model which consists of the RNN LSTM model and FBProphet algorithm. Then we used multivariate linear regression to find the pattern between the outputs of the two models. Finally, we predict the stock market trend using multivariate linear regression which has already learned the patterns from the training data.**

***Index Terms*—RNN, FBProphet, time series, stock market, machine learning, regression.**

I. Introduction

Stock market prediction is a very popular research topic. It is because every day new methods or models are being introduced and even though the improvement they bring over their predecessor is very small sometimes but it is very important. The stock market as mentioned before is a very volatile market, The market is subject to change due to many reasons. That is why the need of creating a prediction model is very high. This way the customers who do transactions in the stock market can use this prediction or forecasting model to trade with minimized loss and maximum profit.

In previous years many attempts and approaches are taken to forecast the stock market using various methods and models. For example, a very popular method is using various financial news to predict the stock market as the market depends on them [13]. Another approach is using the various stock market indices and using regression or neural networks to predict the future trend [4]. The problem with these approaches is that the trend tends to linearize after a certain period of time. A linear line can hardly represent the trend of a stock market as we have previously mentioned that the stock market is a very volatile place.

In this work, we tried to forecast the stock market trends from the historical data of the stock market. We used approximately 2.5 years' worth of historical data on the stock market in our work from the yahoo finance website. The test result was later cross-checked with the actual value to find how the model was doing in real life. It gave a satisfying accuracy result.

II. Literature Review

Stock prices depend on various factors such as economic factors, political factors, change of leadership, and many more factors. So the stock market has a nature of high volatility which is why it is not easy to predict. Most of the approaches to stock prediction are developed on technical and constitutional analysis of stocks. To predict stock prices several models and methods have been developed in past years. Among them, the artificial neural network (ANNs) model is very popular because of its ability to learn patterns from data and derive a solution from unknown data. Ayodele, Aderemi, and Charles [1] obtained data from New York Stock (NYSE) and Nigeria Stock Exchange (NSE) and used the autoregressive integrated moving average (ARIMA) models to predict the stock price. They used this model because it has a strong potential for short-term prediction and for stock prediction it can compete beneficially with other existing prediction techniques. But they used only one model and it will not generate a very accurate output. In another work, by Qun, Lingyu, and Gaowei [17] tried the prediction of stock opening price by a robust time series learning model. The used sentimental analysis model with LSTM time series learning model for the prediction of stock price. But they focused on news sources rather than previous data to detect the stock market but all news cannot be reliable. Saloni, Sahitya, and Sudheer [9] gathered a big amount of time series data and by using deep learning models, tried to boost the accuracy of stock price by analyzing the data in relation to related news articles. They collected data-set which includes the daily stock prices of "S&P500" companies for five years with more than 265000 articles on fanatical news related to those companies. Then they used deep learning models like ARIMA, Facebook Prophet, RNN, LSTM, and RNN LSTM Multivariate model on those data-set for predicting the stock price. But they used those models individually to predict rather than combining the algorithms with each other. Ping-Feng and Chih-Sheng [10] focused on forecasting stock price problems and proposed utilization of the unique strength of the ARIMA model and the SVM model using a hybrid methodology. Erkan, Gulgun, and Tugrul [4] focused on some neural network models such as multi-layer perceptron (MLP), dynamic artificial neural network (DAN2) and the hybrid neural networks which use generalized auto-regressive conditional heteroscedasticity (GARCH) to extract new input variables. They stated that these models are known to be effective and dynamic in stock market predictions. So, they tried to evaluate the effectiveness of these models. In our work, we used both FBProphet and LSTM models and combined them. After that, we used a gradient tree regressor to create a relation between the true value and output of our combinational models. And then used that relationship to predict the output of the next 10 days which gives us more accurate results.

III. TERMINOLOGIES

To deal with time-series data, remembering old information is very crucial for accurate prediction. RNN LSTM offers this memory option to hold the previous state data while predicting the future state.

A. Long Short term memory: LSTM

LSTM is a special version of RNN used to handle long-term dependencies or uncertainty.LSTM hidden layers consist of memory blocks that contain one or more memory cells and gates. The input, output and forget gates are used to do read, write and reset operations. LSTM mainly consists of sigmoid and tanh layers. Firstly, the forget gate layer decides which information to keep or through away from the cell state. Next, a sigmoid input gate layer and Tanh layer were multiplied and added to the previous cell state to calculate the next cell state. Lastly, the cell state is put through a Tanh and multiplied by the output of the sigmoid gate to find out the output [16] . This output will contribute as the input for the next cell state. The equations are given below

$$f_t = \sigma(W_f[h_t - 1, x_t] + b_f) \tag{1}$$

$$i_t = \sigma(W_i[h_t - 1, x_t] + b_i) \tag{2}$$

$$C_t = \tanh(W_c[h_{t-1}, x_t] + b_c) \tag{3}$$

here Ct provides the change content. Updating cell equation can be denoted as below:

$$c_t = f_t * c_{t-1} + i_t * C_t \tag{4}$$

output equation

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_0) \tag{5}$$

"Equation (1),(2), (3), (4), (5)"denotes the full insight of LSTM network This is how LSTM helps to remember previous state data and predicts future state [16] [3] .

B. Facebook Prophet

The prophet is a very sensitive time series forecasting approach that uses add-on models to anticipate seasonal, weekly, and daily trends, as well as holiday results. The prophet is very keen on finding missing data and fluctuations in trends, and it regularly treats unknown data. The prophet explains how to generate a more accurate prediction that is considerably quicker than previous time series forecasting methodologies. It works best when combined with a series of time periods that have significant seasonal outcomes and a few seasons of historical data. When compared to other models, this model takes extremely minimal computing time. The prophet is equivalent to Stan's models in terms of getting predictions in a couple of seconds. It allows us to acquire reliable weather predictions from messy data with minimal human effort. The Prophet has numerous "human" seasons and periods of the year.

$$y(t) = s(t) + g(t) + h(t) + \epsilon_t \tag{6}$$

In this case, h(t) reflects the impacts of vacations, while g(t) indicates the s(t) reflects periodic variations in the trend function. The incorrect phrase ϵ_t reflects any idiosyncratic modifications that occur after the parametric transformation. It is assumed that ϵ_t is regularly distributed. [7]

C. Mann-Kendall Test

A prominent non-parametric approach for finding significant trends in time series data is the Mann-Kendall trend test. However, the original Mann-Kendall test did not account for serial correlation or seasonality effects. Moreover, The observed data are autocorrelated in many real-world circumstances, leading to the inaccurate interpretation of trend test findings. Seasonality exists in water quality, hydrologic, meteorological, and other natural time series. To circumvent the limitations of the original Mann-Kendall test, many modified Mann-Kendall tests have been devised. [5]

IV. METHODOLOGIES

For our time-series approach, the two algorithms we used to predict closing prices, as already mentioned are FBProphet and RNN LSTM. By predicting the data set of the last 30 days from the data gathered, we predict the closing stock prices of not more than 10 days in the future. Any more would cause the trend to go linear, dismissing any validity of the predicted data. The following diagram in "Fig. 1" gives an overall summary of our approach.

The data set that we used for bench-marking our model consists of 1000 days of stock price data in the following format(High, Low, Close, Volume). At the very beginning, we are considering the closing prices of the stock and using it to find the future trend. As we already mentioned that we are using the FBProphet model and the RNN LSTM model to find the trend.

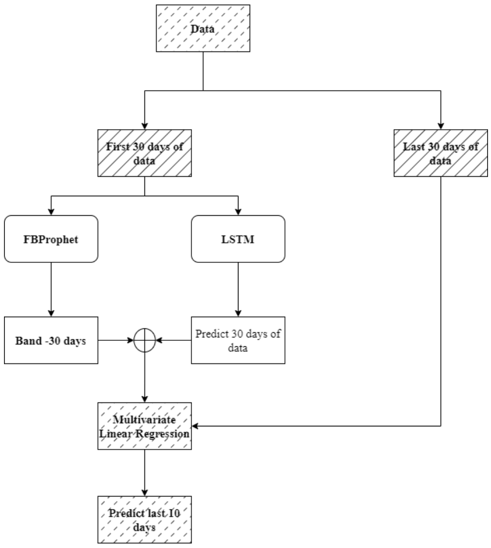

Fig. 1. Proposed Approach.

The work is mainly based on combining the popular FBProphet model and the RNN LSTM network and finding the the trend of the stock market in the short term. The output prediction was fixed (10 days), as usually in real life, stock traders only need up to 10 days of data for a short-term trade. The training data consists of "HSI" stock prices for the period July 10, 2015, to July- 20-2020, and we are deducting the pandemic data because the pandemic is an extreme phenomenon. At first, we clipped the last 30 days of data to find the relation between the FBProphet, RNN LSTM model, and the actual output of the data, as we already know the desired output for our model, we used that information to train our regression model and later whenever we are predicting the stock trend, this regression the model will help us find the relation between the FBProphet, RNN LSTM and the actual output. We are considering many factors while training our regression models, like upper-trend, lower-trend weekly-higher, and weekly-lower. These indicators will give us more insights into the data we cannot get from the primary data. We all know that there are many regression models in the machine learning domain, And it is hard to go with one particular model. That is why we create a pipeline with multiple regression models mainly consisting of models like SVR, Linear Regression, Multivariate linear regression, decision tree regression, and random forest regression. We evaluate the models' error and r2 score. We found that multivariate linear regression works best to find the relation between the input and output data. So finally, we decided to go with the multivariate linear regression model.

When we used RNN LSTM architecture to find the trend, it needed to be more accurate to make a decision, so we used the FBProphet model. Which is a more reliable method to predict the future, but it will also not be super accurate. What FBProphet does is it gives us a boundary of how the trend should follow. Furthermore, it also finds the weekly higher values, weekly lower values, and trend upper and lower values. We can see from "Fig. 2" that if we input some historical data to FBProphet, it outputs the red line and yellow line where the red line indicates the upper bound of where the market should go and the yellow line indicates the lower bound of where the market should go. Finally, we cannot. Remove the insights from RNN-LSTM output because it gives us some insights that could help predict the perfect trend. So at this stage, we combine fbprofet and RNN-LSTM output data, which is not very accurate, but it gives us meaningful insights about the future trend of the market. We then train this singular combined dataset using Multivariate linear regression. We already discussed why we are using this regression algorithm.

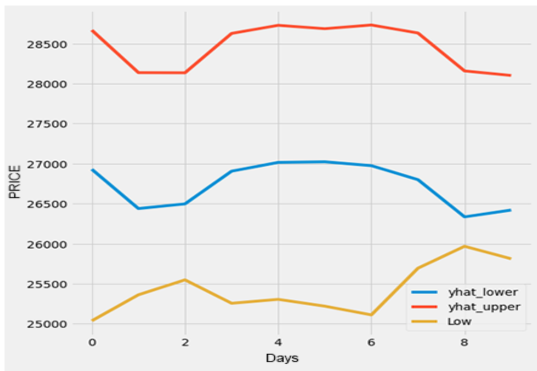

Fig. 2. Data Prediction of FBProphet.

"Fig. 3", indicates the output trend from the RNN LSTM model. As we can see from the following figures, now we have a region of how the market should go, and a line indicates where the market should follow, so now we need another line indicating the actual market value.

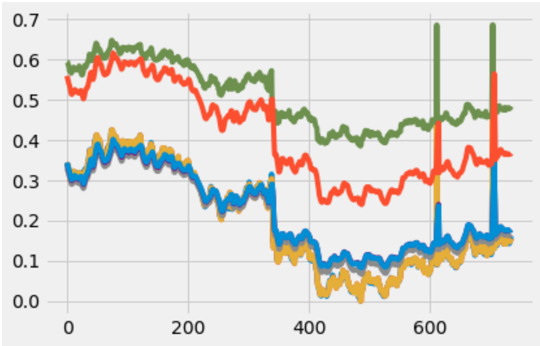

Fig. 3. RNN LSTM Output Trend.

"Fig. 4" indicates the actual value of the market. We need to train our regression model as we already have the input and the output. So now, it is just a supervised learning task. As already mentioned, for the input of the multivariate regression model, we are using the combined output of FBProphet and the RNN LSTM.

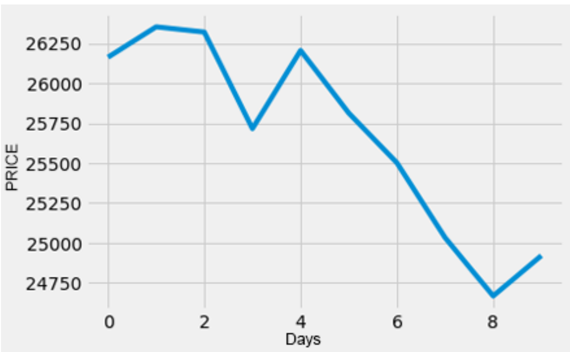

Fig. 4. Actual Market Value.

So now we have a trained model that can find the relation between our two model outputs and the actual output. Now we run the model for future data where we need to know the actual market value, but we need the FBProphet and the RNN LSTM output to find the trend. As we did with the training data for the model, we did a similar thing to find the combined output of the FBProphet and the RNN LSTM model then we put it as an input of the trained multivariate linear regression model. Finally, in the output, we get the final market trend.

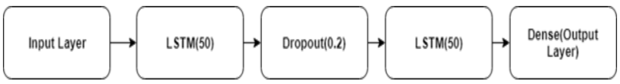

Fig. 5. RNN LSTM Architecture.

"Fig. 5" describes the RNN LSTM architecture of our model, which is a 50*50 neural network model. To find this model, we initially started the training with a small model which consisted of only a 10*10 neural network with no dropout then we use a grid-search of all the hyper-parameters and found the model that we proposed has the best accuracy. Apart from this, we are using Mean Absolute Error as our error function. As an optimizer, we used Adam. For the input of the RNN, we used the historical price of 4 days and LSTM-RNN to predict the next day's price. That is how we are iterating through all our data and making the prediction for RNN LSTM. Finally, for evaluating the model, we are using Mann-Kendall Test, from which we can get the increasing and decreasing trends for the predicted output and the desired output. Furthermore, for some data, it cannot find the trend. For those cases, we are neglecting the results.

V. Result and Analysis

In this section, we will discuss the result of our model and how we obtained it, and some comparison of results due to tinkering with our model. As mentioned before we used the historical data of HSI to test our model. We used RNN LSTM, FBProphet. At first, we tried to predict the market trend using RNN LSTM but as mentioned before, the out result becomes linear after a certain moment and more on that later. Later we introduced FBProphet but the range from FBProphet was not enough to predict the trend but it has some insights about how the market will move. This is why we took one year's worth of historical data and took away the last 30 days of data. Then we used RNN LSTM to predict the trend of the next 30 days. Now that we have the predicted output and also the actual trend of those 30 days, then we used multivariate linear regression to find a correlation between these two data. Later we used that correlation to predict the future trend of the stock market.

Now clarifying why we didn't use RNN LSTM solely. We know why RNN is different than all of the Neural Net models because it works recurrently which means it uses its output as input. When we try to predict the stock market trend for a long period, for example, 20 or 25 days, the trend gets linear. The reason behind this problem is the output is constantly being used as input. We have used different types of regressional models to find the correlation between input and predicted data. The name of the regression models is specified in the table below. scaled linear regression worked well in terms of mean square error and r2 score.

TABLE I
COMPARATIVE STUDY OF ALGORITHMS DURING TRAINING

Algorithm	MSE	R2 score
ScaledLR	355.059	0.2562396275909937
LASSO	421.20	-0.04670113259034414
Elastic Net	507.41	-0.5189848286093712
KNeighbours Regressor	522.33	-0.6096178263130851
CART	460.69	-0.25217047398252346
GBM	371.51	0.18570835427807153
Support Vector regressor	424.94	-0.06536019221981948

The table above shows the comparison of Mean Square Error and R2 Score between different algorithms used to find the correlation between true data and predicted data.

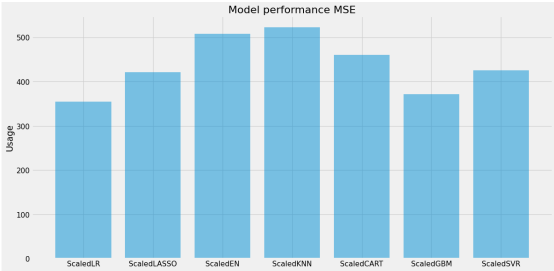

Fig. 6. MSE score of different regression during training.

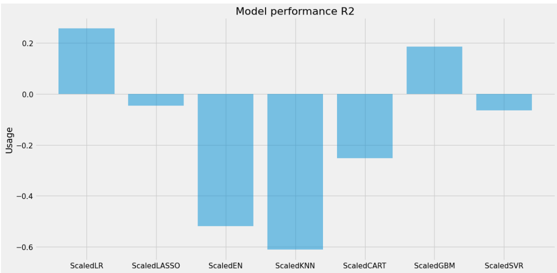

Fig. 7. R2 score of different regression during training.

In the following graph, the real value of the stock market is compared with the value predicted by the model.

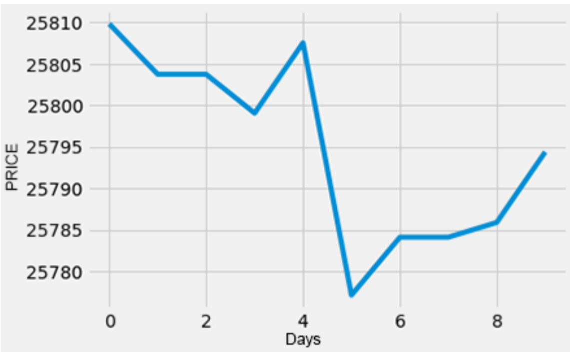

Fig. 8. Value generated by the model.

TABLE II
COMPARISON OF ACTUAL AND PREDICTED TREND

Iteration number	No. of correct trend	No. of incorrect trend
Iteration 50	28	13
Iteration 100	67	27
Iteration 150	93	37

From "Fig. 4" and "Fig. 8", it can be seen that the model comes very close to the actual trend of the market. Here, our actual goal is to find out the trend of the stock but not the price point. To analyze it more precisely, we took around 4.1 years' worth of data and iterate over it with a batch size of 10 days per iteration. In this way, we can see from the above table that at the 50th iteration (between day 1 to day 500) we got 28 correct trends and 13 incorrect trends only. We evaluated the actual and model output trend using the Mann-Kendall test system that we described earlier. This way at the 150th iteration we got 93 correct trend values and 37 incorrect trend values. Also, the mann-kendall test recognizes some trends as "no trend" where the actual stock market is volatile.

VI. Conclusion

As the stock market is highly volatile and the use of any single approach to predict far ahead is improbable. Our time series approach of combining FBProphet, RNN LSTM to capture the influence and to train a multivariate linear regression model produces satisfactory results up to a certain period of how the market will behave. We used multiple other training models, to test whichever conforms to the most accurate results of the market. The model we chose produced the most accurate results in determining the trend of the stock market. This work would allow investors in the stock market to better determine whether it would be better to invest now or to sell the stocks.

References

[1] A. A. Ariyo, A. O. Adewumi, and C. K. Ayo, "Stock price prediction using the arima model," in *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*. IEEE, 2014, pp. 106–112.

[2] M.-C. Chan, C.-C. Wong, and C.-C. Lam, "Financial time series forecasting by neural network using conjugate gradient learning algorithm and multiple linear regression weight initialization," in *Computing in Economics and Finance*, vol. 61, 2000, pp. 326–342.

[3] A. Graves, S. Fernández, and J. Schmidhuber, "Bidirectional lstm networks for improved phoneme classification and recognition," in *International conference on artificial neural networks*. Springer, 2005, pp. 799–804.

[4] E. Guresen, G. Kayakutlu, and T. U. Daim, "Using artificial neural network models in stock market index prediction," *Expert Systems with Applications*, vol. 38, no. 8, pp. 10 389–10 397, 2011.

[5] M. Hussain and I. Mahmud, "pymannkendall: a python package for non parametric mann kendall family of trend tests." *Journal of Open Source Software*, vol. 4, no. 39, p. 1556, 2019.

[6] H. D. Huynh, L. M. Dang, and D. Duong, "A new model for stock price movements prediction using deep neural network," in *Proceedings of the Eighth International Symposium on Information and Communication Technology*, 2017, pp. 57–62.

[7] S. Kaninde, M. Mahajan, A. Janghale, and B. Joshi, "Stock price prediction using facebook prophet," in *ITM Web of Conferences*, vol. 44. EDP Sciences, 2022, p. 03060.

[8] N. Merh, V. P. Saxena, and K. R. Pardasani, "A comparison between hybrid approaches of ann and arima for indian stock trend forecasting," *Business Intelligence Journal*, vol. 3, no. 2, pp. 23–43, 2010.

[9] S. Mohan, S. Mullapudi, S. Sammeta, P. Vijayvergia, and D. C. Anastasiu, "Stock price prediction using news sentiment analysis," in *2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService)*. IEEE, 2019, pp. 205–208.

[10] P.-F. Pai and C.-S. Lin, "A hybrid arima and support vector machines model in stock price forecasting," *Omega*, vol. 33, no. 6, pp. 497–505, 2005.

[11] A. N. Refenes, A. Zapranis, and G. Francis, "Stock performance modeling using neural networks: a comparative study with regression models," *Neural networks*, vol. 7, no. 2, pp. 375–388, 1994.

[12] E. W. Saad, D. V. Prokhorov, and D. C. Wunsch, "Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks," *IEEE Transactions on neural networks*, vol. 9, no. 6, pp. 1456–1470, 1998.

[13] R. Schumaker and H. Chen, "Textual analysis of stock market prediction using financial news articles," *AMCIS 2006 Proceedings*, p. 185, 2006.

[14] W. Shen, X. Guo, C. Wu, and D. Wu, “Forecasting stock indices using radial basis function neural networks optimized by artificial fish swarm algorithm,” *Knowledge-Based Systems*, vol. 24, no. 3, pp. 378–385, 2011.
[15] J.-J. Wang, J.-Z. Wang, Z.-G. Zhang, and S.-P. Guo, “Stock index forecasting based on a hybrid model,” *Omega*, vol. 40, no. 6, pp. 758–766, 2012.
[16] Y. Yu, X. Si, C. Hu, and J. Zhang, “A review of recurrent neural networks: Lstm cells and network architectures,” *Neural computation*, vol. 31, no. 7, pp. 1235–1270, 2019.
[17] Q. Zhuge, L. Xu, and G. Zhang, “Lstm neural network with emotional analysis for prediction of stock price.” *Engineering letters*, vol. 25, no. 2, 2017.