Stock Market Price Prediction Using LSTM RNN

Kriti Pawar, Raj Srujan Jalem and Vivek Tiwari

Abstract Financial Analysis has become a challenging aspect in today's world of valuable and better investment. This paper introduces the implementation of Recurrent Neural Network (RNN) along with Long Short-Term Memory Cells (LSTM) for Stock Market Prediction used for Portfolio Management considering the Time Series Historical Stock Data of Stocks in the Portfolio. The comparison of the model with the traditional Machine Learning Algorithms—Regression, Support Vector Machine, Random Forest, Feed Forward Neural Network and Backpropagation have been performed. Various metrics and architectures of LSTM RNN model have been considered and are tested and analysed. There is discussion on how the sentiments of the customer would affect the stocks along with the changes in trends.

Keywords Recurrent neural network · Long short-term memory
Trading · Portfolio optimization

1 Introduction

Predictions on the stock market have been considered as an important study object for many decades [1]. But its complexity and dynamic environment have been proven it to be a very difficult task [2, 3]. Predicting price and trend of the stock market are the indispensable aspects of investment and finance. Many researchers have worked and proposed their ideas to forecast the market price to make a profit while trading using various techniques such as technical and statistical analysis.

K. Pawar · R. S. Jalem · V. Tiwari (✉)
DSPM IIIT, Naya Raipur, India
e-mail: vivek@iiitnr.edu.in

K. Pawar
e-mail: kriti15100@iiitnr.edu.in

R. S. Jalem
e-mail: raj15100@iiitnr.edu.in

V. S. Rathore et al. (eds.), *Emerging Trends in Expert Applications and Security*, Advances in Intelligent Systems and Computing 841,
https://doi.org/10.1007/978-981-13-2285-3_58

Observing and predicting trends in the price of the stock market is challenging because of noise and uncertainties involved. There are numbers of factors that may affect the market value in a day such as country's economic change, product value, investors' sentiments, weather, political affairs, etc. [4]. The authors have also studied and researched on the trends and behaviour of stock prices and what all factors affect prices the most. Al-Nasseri et al. [5] have analysed the divergence of opinion and the impact of the disagreement on Stock Returns and Trading Volumes.

In this research, RNN along with LSTM is used for predicting the movement of the stock market. The stock market is also mainly affected due to the sentiment of customers or buyers that is their opinion on a particular product or service provided by the company is also one of the main additions to the fluctuations in stock prices. The research also compares the RNN-LSTM model with many Traditional Machine Learning algorithms. Several possibilities have been considered for the model and the model has been tested accordingly to several possibilities and is analysed for different configurations of the model.

2 Literature Survey

Aditya Gupta and Bhuwan Dhingra in [6] used Hidden Markov Model to predict the close price of the stocks of next day. They have used historical stock prices of different companies such as the Apple Inc., IBM Corporation, TATA Steel and Dell Inc. Inputs were High Price, Low Price, Open Price and Close Price. Model for each stock was supposed to be independent of every other stock. The model was first trained for a period of 7 months. The model was tested using MAPE values.

Lin et al. in [7] have been used SVM based approach to predict the price of stock market trends. They have solved the problem in two parts, i.e. feature selection and prediction of the direction of trends in the market. SVM correlation has been used to select the features which affect the price mostly. Linear SVM is applied to the data series to predict the direction. They have shown the system to select the good feature and control overfitting on stock market tendency prediction.

Dinesh and Girish in [8] developed a model based on linear regression, as in linear regression, there is given set of input for output and by developing a model based on mathematical foundation output is predicted. They have used Open Price, High Price, Low Price and volume as input to the model and independent variable and Close Price as the label, and had considered Date is used as a variable index. By comparing the linear regression model with polynomial and RBF regression approach, linear regression has proven to be the best among both.

Yang et al. in [9] first select the most relevant features for prediction of the stock price by calculating maximal information coefficient. They build their assembler model using three different outstanding classifiers on stock market trend prediction SVM, Random forest and AdaBoost and collectively named as SRAVoting. They validate their model on Chinese Stock Market and come to the conclusion that

SRAVoting gives higher accuracy than SVM but at the same time lesser buy/sell strategies than SVM.

In [10], they have compared Random Forest, SVM and Gradient Boosted Trees for forecasting Moroccan stock market for the short term. The empirical results showed that all the three models have given very satisfactory results and they have short time responses and hence shows that these methods can be usable for a short time. They have come up with the result that Random Forest and Gradient Boosted Trees is superior to Support Vector Machine. They have also suggested that proper feature selection and reduction is required for more accurate results.

In [11], they have used optimized ANN to predict the direction of the price movement of the next day of Japanese stock market. To improve the accuracy of the predicted direction they have introduced the Genetic Algorithm. Another method hybrid GA-ANN is also used in order to predict the direction of price movement. After comparing both methods, the second method is prone to give satisfactory result in term of accuracy. By adjusting weights and biases of ANN using GA, the model gave the Hit ratio of 86.39%. The proper feature selection is required to gain more accuracy.

The backpropagation neural network remains the universal and most fruitful prototype for multilayer networks [12]. The typical backpropagation neural network contains three layers: input, output, and should have at least one hidden layer. The networking potential for the selected size of the dataset depends on a number of neurons at each layer and the number of hidden layers to gain correct result [13].

The proposed scheme/method is to preprocess the dataset, which is followed by an altered backpropagation neural network scheme/algorithm with the attention to contemporary fashion and event. At last, the predicted value from the model is compared to the input value to minimize error. Here, we can concentrate on accuracy or, in other terms, minimize the error to get the accurate predicted value as compared to actual value. The problem with Feed-Forward neural network [14] has been overcome by supervised learning methods where prior knowledge is not required. This gives the better results than the Random Forest, we can make the layers recurrent for considering the sequential stock data.

3 Prediction Using LSTM RNN

3.1 Recurrent Neural Networks

Recurrent Neural Networks are the class of Neural Networks [15] where the units are recurrently connected. This allows them to use their internal memory for processing the sequence of inputs. This allows them to be used for handwritten recognition, text generation, the stock market or speech recognition. Recurrent Neural Networks are used in this project since long-term dependencies [16] in the data needs to be considered for the stock data. While due to the inability to store the memory for much amount of time, Vanishing Gradient descent problem may occur, i.e. after every iteration in the neural net, the data it holds gets vanished going deeper. Due to which Long Short-Term Memory cells instead of traditional Neuron-like cells are used.

3.2 Long Short-Term Memory Cells

Long short-term memory (LSTM) block [17] or network is an advancement to the simple recurrent neural network which can be used as a building component or block for an eventually better serial analysis using the recurrent neural network. LSTM block itself a recurrent network as it contains recurrent connections like connections as in a conventional recurrent neural network. LSTMs is designed specifically as a recurrent neural network architecture to consider them for long-term dependencies more accurately than the conventional Recurrent Neural Networks. According to [16], LSTM along with RNN have outperformed the Deep Neural Networks (DNNs) and the simple RNN models for predicting the movements in stock data or speech recognition.

Conventional DNNs can only provide modelling for a fixed sized sliding window where the network does not interdepend on the previous time steps which would so do not provide a good modelling for the stock data (Fig. 1).

Data is retrieved from the online open source financial data provider Yahoo Finance. For the training purpose, historical stock data of S&P 500 was considered as it has a large database. The data is then normalized and split into the training and the testing data. The training data is then used to train the built LSTM RNN model to train for predicting or forecasting the sequence of the stock data. Then, the model is tested on several stocks like the Apple Inc., Tesla Inc., Google and the forecasted versus the actual data is visualized through the plot in the results section (Fig. 2).

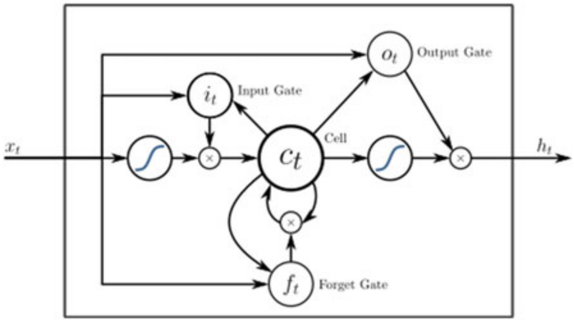


Fig. 1 Long short-term memory cell

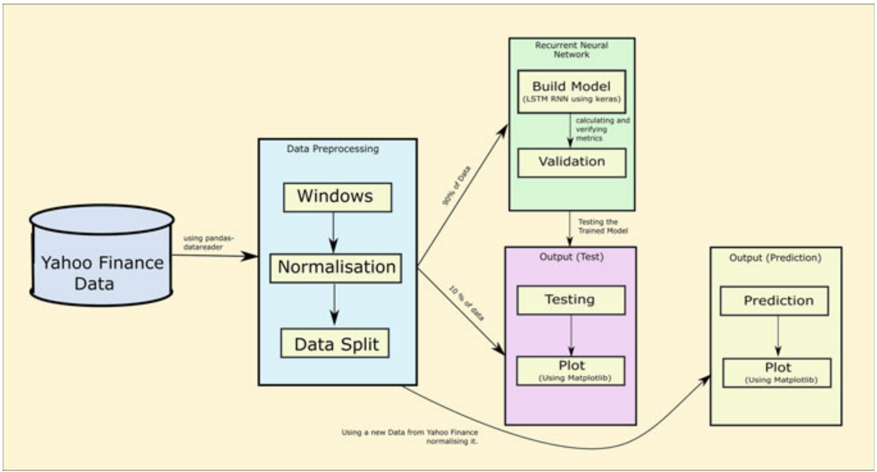


Fig. 2 Architecture of the proposed model

Algorithm: Prediction using LSTM RNN

Input: Historical Stock Prices of stocks.
Output: Predicted Stock Prices for n Data Points.

Data <= Historical Stock Data to be retrieved from **Yahoo Finance.**

Adj Close <= Adjacent Close values retieved from Data.

Function Preprocessing(Adj Close, sequence_length)

Input: Adj Close is the Adjacent Close values that are retrieved from the Data.
Output: The Data is normalised and split to Train and Test

Data_windows <= windows(Adj Close, sequence_length)

Function Normalise(Data_windows)
Normalised_data = []
For i in Data_windows:
Normalised_window = [((float(p) / float(i[0])) - 1) **for** p in i]
Normalised_data.append(Normalised_window)
Return Normalised_data
row <= 90% of the shape of normalised data
Train_data <= [: row, : -1]
Train_label <= [: row, -1] *//as sequence prediction is done the same data is divided to*
Test_data <= [row:, : -1] *train_data, train_label, test_data and test_label.*
Test_label <= [row:, -1]

Function model(a, b, epoch, batch_size);

Input: a and b are the Train_Data and Train_label respectively.
Output: the Trained model is obtained.

Network = **sequential()**
Network.**add**(LSTM(input, output, dropout)) *//Input Layer of LSTM RNN*
Network.**add**(LSTM(cells, activation, dropout))$_1$… Network.**add**(LSTM(cells, activation, dropout))$_k$…Network.**add**(LSTM(cells, activation, dropout))$_n$ *//Hidden Layers*

Network.**add**(LSTM(output, output_activation) *//Output Layer*

Network**.compile**(loss_function, optimizer) *//Defining Optimisation of the model*

Network**.fit**(a, b, epoch, batch_size, validation) *//Training the model*

Function Plot(model, validation_data, validation_label)
Input: model is the trained model i.e., Network is given as the input, validation_data is the Test_data and validation_label is the Test_label.
Output: Plot is obtained i.e., Predicted vs the validation_label

Prediction = model.**predict**(validation_data) *//Predicting the validation_label for validation_data*
Plot(predicted vs validation_label) *//Plotting graph of Predicted vs the True Data*

4 Results

All the results are obtained by different configurations of the model with the loss function of mean squared error and an optimizer of Adam (Adam is an optimization algorithm which is used to update the weights of the network iterating based on training data). Adam was created by Jimmy Ba from the University of Toronto and Diederik Kingma from OpenAI in their paper titled 'Adam: A Method for Stochastic Optimization'.

There are several LSTM architectures we can build as shown in Fig. 3, using the LSTM structure or the combination of LSTM and RNN. In this paper, several architectures are tested to find the best model with the lowest loss value.

The loss observed for the LSTMP architecture after three epochs are 0.5770.

The loss observed for the Deep LSTM after three epochs are 3.1464e−04, which is much better than the other models.

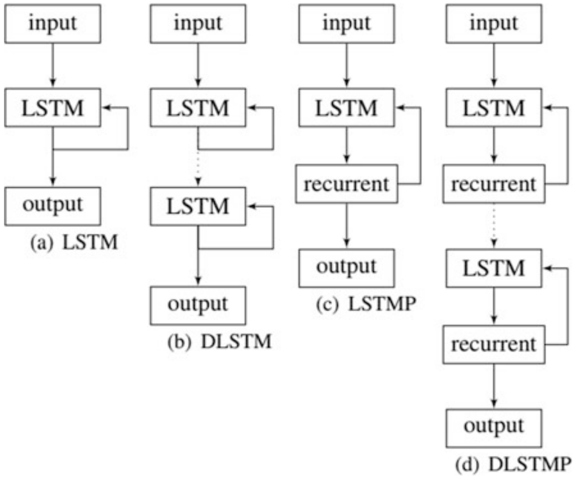


Fig. 3 Several architectures of LSTM

Table 1 Several LSTM architectures with their loss observed

	I	II	III	IV	V	VI
Cells	128	128	256	256	512	512
Layers	1	2	1	2	1	2
Loss[a]	2.554e −04	2.835e −04	2.307e −04	2.455e −04	2.080e −04	2.0540e-04

[a]The loss function used is "Mean Squared Error"

As the LSTM architecture has got the best loss value, the behaviour is also checked by varying the inner architecture, i.e. the number of cells and the number of layers keeping constant the activation function of the hidden layer as hyperbolic tangent function and of the output layer as the rectified Linear Unit function as these are observed to give the best results. All the results are obtained for five epochs (Table 1).

The historical stock data of AAPL (Apple Inc.), GOOG (Google), and TSLA (Tesla, Inc.) have been considered from the Yahoo Finance and have been normalized to fit the data into the model for prediction and the predictions have been taken for an epoch = 10 (10 iterations through the entire data and the model is optimized) with a batch size (number of data points to be considered at a time to be 128) (Figs. 4, 5 and 6).

For Epoch = 10.

Fig. 4 Predicted versus true data movements of Tesla Inc.

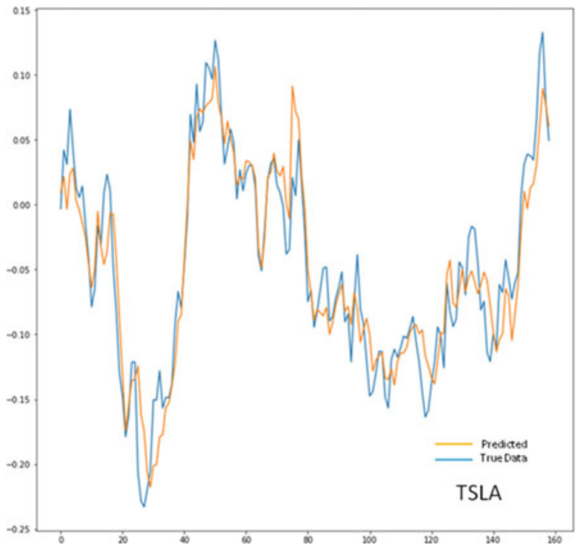
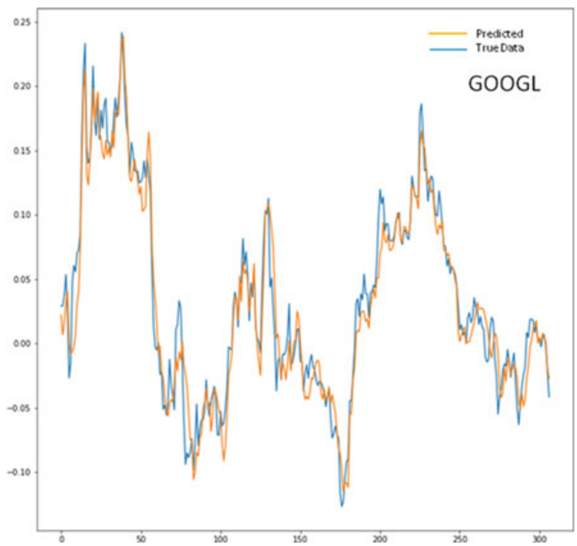


Fig. 5 Predicted versus true data movements of Google Inc.



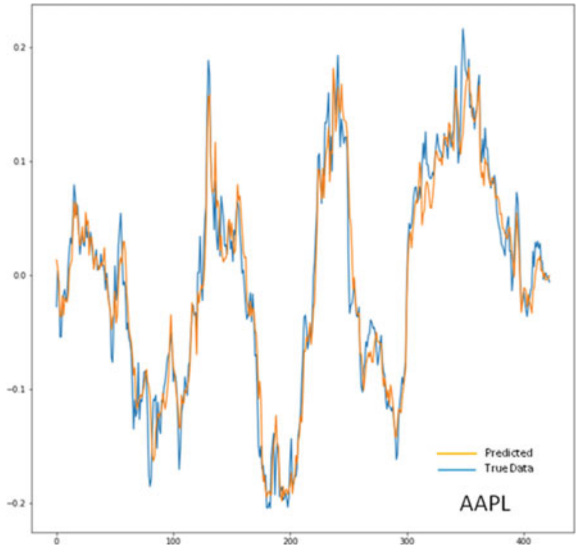


Fig. 6 Predicted versus true data movements of Apple Inc.

5 Discussion

When an investor has been predicted and understood the behaviour of the market based on rising and fall in the price of the assets. The Next problem comes in the trading strategy is how much proportion of his share he could distribute in different stocks in his portfolio [18], he should have prior knowledge before investing. The adjustment of the proportion of wealth in different stocks to gain profit is a very important factor to understand. Markowitz's portfolio optimization [19] is purely based on the mathematical foundation and gives the very satisfactory result to distribute ones share in the different market [20].

We have calculated return as

$$r = (adjClose(i) - adjClose(i-1))/adjClose(i)$$

annual return by multiplying 12 to return, the variance of return, the covariance matrix of return, weights and sum should be 1,

$$\begin{aligned} expected\ return &= transpose\ of\ weight\ *\ return, \\ volatility &= (transpose\ of\ weights * covariance\ of\ return * weights)^{1/2}, \\ Sharpe\ ratio &= expected\ return/volatility \end{aligned}$$

We then adjust the weight to gain profit and select the portfolio which has max Sharpe ratio and in volatility. By performing Markowitz Portfolio Optimization technique, we can understand how much we must invest our money in a market.

6 Conclusion

To develop the prediction model, the implementation process should be gone through relevant data collection, data preprocessing to remove noise and missing values. Analysing the best Algorithm followed by model evaluation. The research introduced in this paper uses the Recurrent Neural Network with LSTM cells to predict the movement of stock market exchange.

The results show that RNN-LSTM model prone to give more accurate result than the traditional machine learning algorithms.

This model can be proved to be productive for individual traders as well as for corporate investors. They can get the future behaviour of market price movement and take the proper action to make a profit.

In future work, the model should be considered different features and aspects of the market to make prediction more accurate. Also, we intend to use reviews of the users on the product to predict the change in the market.

7 Future Implementation

Stock Data not only depends on the Trend in the Historical Data, it also mainly depends on the product value or the satisfaction of the customers with the company's market [21]. So, the future implementation includes analysing the sentiment of the customers reviewed on the products related to a company or its domain and add this analysis to the prediction using RNN-LSTM. Research work has been carried out using Naive Bayes Classifier with Large Movie Review Data as the Training set and then tested (sentiment analysis) on Amazon Review Data.

Thus, the results obtained through both the analysis are considered to obtain better forecasting of the portfolio.

Acknowledgements This research was partially supported by DSPM International Institute of Information Technology Naya Raipur (IIIT-NR). We thank our colleagues from IIIT-NR who provided insight that greatly helps us in this research. We would like to show our gratitude to Dr. Vivek Tiwari, Asst. Prof. CSE, IIIT-NR for mentoring us and sharing his experience and knowledge with us during this research. We thank every person associated with this research directly or indirectly.

References

1. Li, R., Fu, D., & Zheng, Z. (2017). An Analysis of the Correlation between Internet Public Opinion and Stock Market. In 2017 4th International Conference on Information Science and Control Engineering (ICISCE) (pp. 150–153). IEEE.
2. Tiwari, V., Gupta, S., & Tiwari V. (2010). Association rule mining: A graph based approach for mining frequent itemsets. International Conference of Networking and Information Technology (ICNIT) (pp. 309–313). IEEE.

3. Kunal, S., Saha, A., Varma, A., & Tiwari, V. (2018). Textual Dissection of Live Twitter Reviews using Naive Bayes. *Procedia Computer Science, 132*, 307–313. Elsevier.
4. Khaidem, L., Saha, S., & Dey, S. R. (2016). Predicting the direction of stock market prices using random forest. arXiv preprint arXiv:1605.00003.
5. Al-Nasseri, A., & Ali, F. M. (2018). What does investors' online divergence of opinion tell us about stock returns and trading volume?. *Journal of Business Research, 86*, 166–178.
6. Lin, Y., Guo, H., & Hu, J. (2013). An SVM-based approach for stock market trend prediction. In Neural Networks (IJCNN), The 2013 International Joint Conference on (pp. 1–7). IEEE.
7. Gupta, A., & Dhingra, B. (2012). Stock market prediction using hidden markov models. In Engineering and Systems (SCES), 2012 Students Conference on (pp. 1–4). IEEE.
8. Bhuriya, D., Kaushal, G., Sharma, A., & Singh, U. (2017). Stock market prediction using linear regression. International conference of electronics, communication and aerospace technology (ICECA), Coimbatore, India. IEEE.
9. Yang, J., Rao, R., Hong, P., & Ding, P. (2016). Ensemble model for stock price movement trend prediction on different investing periods. In Computational Intelligence and Security (CIS), 2016 12th International Conference on (pp. 358–361). IEEE.
10. Labiad, B., Berrado, A., & Benabbou, L. (2016). Machine learning techniques for short term stock movements classification for Moroccan stock exchange. In Intelligent Systems: Theories and Applications (SITA), 2016 11th International Conference on (pp. 1–6). IEEE.
11. Mingyue, Q., Cheng, L., & Yu, S. (2016). Application of the Artifical Neural Network in predicting the direction of stock market index. In 2016 10th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS) (pp. 219–223). IEEE.
12. Mithani, F., Machchhar, S., & Jasdanwala, F. (2016). A modified bpn approach for stock market prediction. In Computational Intelligence and Computing Research (ICCIC), 2016 IEEE International Conference on (pp. 1–4). IEEE.
13. Sharma, A., Bhuriya, D., & Singh, U. (2017). Survey of stock market prediction using machine learning approach. International conference of electronics, communication and aerospace technology (ICECA), Coimbatore, India. IEEE.
14. George, S., & Changat, M. (2017). Network approach for stock market data mining and portfolio analysis. In Networks & Advances in Computational Technologies (NetACT), 2017 International Conference on (pp. 251–256). IEEE.
15. Dewan, A., & Sharma, M. (2015). Prediction of heart disease using a hybrid technique in data mining classification. In Computing for Sustainable Global Development (INDIACom), 2015 2nd International Conference on (pp. 704–706). IEEE.
16. Sak, H., Senior, A., & Beaufays, F. (2014). Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In Fifteenth annual conference of the international speech communication association.
17. Heaton, J., Polson, N., & Witte, J. (2017). Deep learning for finance: deep portfolios. *Applied Stochastic Models in Business and Industry*, *33*(1).
18. Mrcela, L., Mercep, A., Begusic, S., & Kostanjcar, Z. (2017) Portfolio optimization using preference relation based on statistical arbitrage. In Smart Systems and Technologies (SST), 2017 International Conference on (pp. 161–165). IEEE.
19. Fang, J., & Xiaoyun, M. (2017) Dynamic Multi-Mode Portfolio Optimization Strategy for Markovian Arrival Process. In Robots & Intelligent System (ICRIS), 2017 International Conference on (pp. 139–142). IEEE.
20. Das, S., & Goyal, M. (2012) Rebalancing a two-asset Markowitz portfolio: A fundamental analysis. In Computational Intelligence for Financial Engineering & Economics (CIFE), 2012 IEEE Conference on (pp. 1–8). IEEE.
21. Yang, Y., & Hasuike, T. (2017) Construction of Investor Sentiment Index in the Chinese Stock Market. In Advanced Applied Informatics (IIAI–AAI), 2017 6th IIAI International Congress on (pp. 23–28). IEEE.