Comparative study on Prediction of Support and Resistance Levels with *k*-Nearest Neighbor and Long Short-Term Memory Methods.

Thambusamy Velmurugan[1], Thiruvalluvan Indhumathy[2]
[1]Associate Professor, [2]Research Scholar
PG and Research Department of Computer Science,
D. G. Vaishnav College, Chennai-600106, Tamil Nadu India
E-Mail: [1]velmurugan_dgvc@yahoo.co.in, [2]tindhu_m3@yahoo.com

Abstract

Predictive Analytics is a broad subject which uses mining techniques, artificial intelligence and statistics to analyze and predict for business organizations to make proactive decisions. In stock markets where money and time are very important aspects, time series analytical and forecasting methods, a part of predictive analytics is applied to make predictions that ultimately benefit the investors. Models from machine learning and deep learning are taken for predicting the support and resistance levels for the indication of entry/ exit price level in stock market. k-Nearest Neighbor (k-NN) and Long Short-Term Memory (LSTM) models were taken for predicting the support and resistance levels and comparison between the models were analyzed. Two historical datasets with a total of 5021 data each were taken for the analysis. From these 4016 data were taken as training data and 1005 data were taken as testing data. The accuracy was tested by using five types of error metrics and performance analysis was assessed using the training time to find the best model for prediction in stock exchange. Deep learning model is expected to perform better when compared to machine learning model that has to be proved through the expected results.

Keywords: *k-Nearest Neighbor, Long Short-Term Memory, Support and Resistance Levels, Error Metrics.*

1. Introduction

Stock market is a volatile field and is the primary indicator of the economic situation for a country. The country`s strength and development depend upon the dynamic factors that determine the value of shares. Investors and Analysts have to keep track of to avail the right opportunity for expecting higher returns, the conditions that affect the prices and the probable future direction. One such factor that contributes to the price fluctuations is trend indicators which are one of the many types of technical analysis. Though it is risky to predict the stocks, with the advent of machine learning techniques [1] prediction has become less tedious and could pave ways for different perspectives.

Prediction through machine learning and deep learning [2] is broad topic where there are many models with their pros and cons. Each model varies from one and another by different aspects. Support and resistance levels for financial datasets are traced using two models, *k*-Nearest Neighbor (*k*-NN) which is a machine learning model, Long Short- Term Memory (LSTM), a deep learning model and comparison between them is analyzed in this paper. Support is a term used in stock market where the price level will pause from going further down and reverse. This is the level where the stocks are bought. The resistance level is an indication for stocks to be sold as there will be no further rise in the prices. The above mentioned two models use Fibonacci retracement - one of the many technical indicators to predict the support and resistance levels which are depicted as horizontal lines drawn at Fibonacci percentages before the price continues in the original direction of its trend. The aim of this study tries to

- Find the model which outperforms the other model in terms of accuracy while predicting the dependant values.
- Compares the performance of machine learning model with deep learning model in terms of computation and training time.

This paper is organized as follows. Section 2 briefs about the models that are taken for comparison, its architecture and its applications. Section 3 analyses the performance and discusses the results that are obtained. The conclusion is presented in section 4.

2. Materials and Methods

A sub-field of predictive analytics is time series forecasting and time is an important factor that has to be considered for predicting the future trend and direction of prices in stock market. Traditionally classical statistical methods like ARIMA models or Exponential smoothing were used to forecast time series but the last decade saw the rise of computational intelligence techniques that were considered better than the traditional models. Artificial intelligence techniques excel classical methods because of the presence of two interesting features - nonlinearity and non-parametric. Machine Learning and Deep Learning models have these features and form a part of artificial intelligence techniques where the mechanism is modeled, extracting a series of values which form a basis to forecast the future values by fitting on the extracted data. Here in this paper, two approaches are taken for predicting the support and resistance level in stock market applied in two historical datasets. *k*-NN and LSTM are the two models taken for comparison where *k*-NN is a machine learning model and LSTM is a deep learning model.

2.1 *k*-Nearest Neighbor

One of the simplest easy-to-implement supervised machine learning algorithm that can be used to solve both classification and regression problems is *k*-Nearest Neighbors (*k*-NN) that was first introduced by Fix & Hodges and later elucidated by Cover & Hart. The Algorithms implementing *k*-NN predicts new data points based on similarity measures from the entire data. The similarity can be measured in many different ways [3], [4] and the most commonly used measure is the distance between the data points.
Though considered to exhibit lazy – learning method it is proven to be effective in classification problems [5], [6]. It is also proven to be effective in regression problems [7] and is applied to time series problems [8], [9], economics, outlier detection and many other fields where the knowledge about data is scarce. Due to its easy implementation *k*-NN is enhanced when combined with other techniques [10] and improvised versions have given satisfactory results.

2.2 Long Short -Term Memory

The Long Short -Term Memory (LSTM) is a model proposed to avoid the problem of vanishing gradient that occurs in recurrent neural network models [11]. It is considered to be effective and scalable for problems related to sequential patterns. A typical LSTM unit comprises of an input gate, a forget gate, an output gate, a memory cell and a hidden state.

- The forget gate is responsible for discarding unnecessary and insignificant information that are no longer needed through filters for boosting the performance of models.
- The information that should enter inside the cell is determined by the input gate with the memory cell each with different activation functions.
- Selecting and displaying the information that are adept is done by the output gate.
- The hidden state decides on what information should be taken to the next sequence.

LSTM is used in a various applications ranging from Natural language processing [12], [13], time series analytics [14], classification problems [15], communication [16] and prediction in stock markets [17], [18].

2.3 Dataset

Two historical Equity stock datasets were taken from Kaggle an online community for dataset repository. The datasets contains a total of 5021 instances each from Jan/3/2000 to Feb/28/2020. For prediction the attributes Date and Close were taken and out of 5021 instances 4016 were taken for training purpose and 1005 for validation purpose. The dataset is in Excel comma separated value format (csv) and a sample of the dataset is shown as Table 1. & Table 2.

Table 1: Sample data from dataset1

Date	Prev Close	Open	High	Low	Last	Close	VWAP	Volume	Turnover
1/3/2000	24.7	26.7	26.7	26.7	26.7	26.7	26.7	112100	2.99E+11
1/4/2000	26.7	27	28.7	26.5	27	26.85	27.24	234500	6.39E+11
1/5/2000	26.85	26	27.75	25.5	26.4	26.3	26.24	170100	4.46E+11
1/6/2000	26.3	25.8	27	25.8	25.9	25.95	26.27	102100	2.68E+11
1/7/2000	25.95	25	26	24.25	25	24.8	25.04	62600	1.57E+11

Table 2: Sample data from dataset2

Date	Prev Close	Open	High	Low	Last	Close	VWAP	Volume	Turnover
1/3/2000	157.4	166	170	166	170	170	169.52	33259	5.64E+11
1/4/2000	170	182	183.45	171	174	173.8	174.99	168710	2.95E+12
1/5/2000	173.8	170	173.9	165	168	166.95	169.2	159820	2.7E+12
1/6/2000	166.95	168	170	165.3	168.95	168.3	168.44	85026	1.43E+12
1/7/2000	168.3	162.15	171	162.15	170.75	168.35	166.79	85144	1.42E+12

3. Experimental results and Performance analysis

Regression models make use of error metrics to assess the quality of the model and are judged by how its predictions match up against actual values. Error metrics get the difference between actual value and the estimates done by these models as residuals. These residuals play an important role and its values that are determined at every point in the dataset are used for the model`s assessment. Here two models are assessed with five types of error metrics, the values are used for comparison and the best fit is sought out. Out of 5021 sequences data 4016 samples were taken for training purpose and 1005 samples for validation purpose. For training the model Adam is set as the optimizer. Google cloud engine was used as a training platform [Machine type: n1-standard-2 (2 vCPUs, 7.5 GB memory), CPU platform: Intel Core i5] and used Windows 7, Keras (Frontend) and Tensorflow (Backend) as the learning environment.

3.1 Discussion

The models that are used in this paper uses the variable date as independent variables and the closing price as dependent variables and then the estimated outputs are checked against the actual values that is tried to predict. The values of the closing price lie in the range 21 to 2050 for dataset1 and 163 to 2566 for dataset2. The RSME, Bias, MAE, MSE & MAPE are calculated, presented as Table 3 for dataset1, and as Table 4 for dataset2 with which the accuracy is sought, thereby assessing the model with best fit.

Table 3: Error Metrics for dataset1- Axis bank

Error Metric	*k*-NN	LSTM
RMSE Score	433.7251905772074	14.016484920882379
Bias	-4.167280876494079	6.084096560535199
MAE	380.2034362549801	10.716072915369768
MSE	188117.5409412351	196.4618495373233
MAPE	380.2034362549801	10.716072915369768

Table 4: Error Metrics for dataset2- HDFC bank

Error Metric	*k*-NN	LSTM
RMSE Score	1007.7112902015714	53.23914149577187
Bias	863.3828851261619	-25.596211273071773
MAE	891.7156540504648	28.617456881458548
MSE	1015482.0443997158	2834.4061872068187
MAPE	891.7156540504648	28.617456881458548

It can be seen from Table 3. that the value of all the Error metrics for LSTM is close to lowest value of the dependant variable whilst the difference between the error values and the dependant variable for *k*-NN is huge. The minimal difference between the estimated values and the dependant values indicate high level of accuracy attained. This being the case, it is clear that LSTM has the minimal difference and so for dataset1 the predictions made by the LSTM model is accurate than *k*-NN and the level of accuracy attained by LSTM is high. Inferring the data from Table 4. for dataset2 the residual values obtained for LSTM is less than the lowest value of the dependant variable indicating a very high level of accuracy. The error values for *k*-NN indicate a huge difference between the dependant values. This huge difference indicates that the level of accuracy is not desirable when compared to the error values of LSTM. So for both the datasets, LSTM have achieved a high level of accuracy, ultimately setting the model to be preferred over *k*-NN.

The Computation time represents the seconds to complete training & testing data and the computation time for the models is displayed in Table 5. The Epochs were taken to analyze the time taken for training and testing LSTM model. With epoch of 2 for dataset1 and epoch of 3 for dataset2, the time is more than one second. As this is a regression problem, the training time for *k*-NN is calculated by sklearn, a machine learning library and the cross validation set to 5. Cross validation is where the data is split into the set number for training and testing for evaluation purposes. It is inferred from Table 5. that the time taken to predict the values by *k*-NN is less than one second for dataset1 and dataset2.

Table 5: The computation time for the datasets

Computation time	*k*-NN	LSTM
Dataset1	0.8588s	76s
Dataset2	0.8513s	80s

The graphical representation is given below for further analysis for both the models. In all the graphs, the independent variable is plotted in x-axis and the dependent variable is plotted in y-axis. The training data, with 4016 instances is depicted in green color lines as they go up and down, representing the movement of the closing price which is the dependant data and the testing data as red color lines with 1005 instances. The unseen data or the forecasted data is depicted in blue.

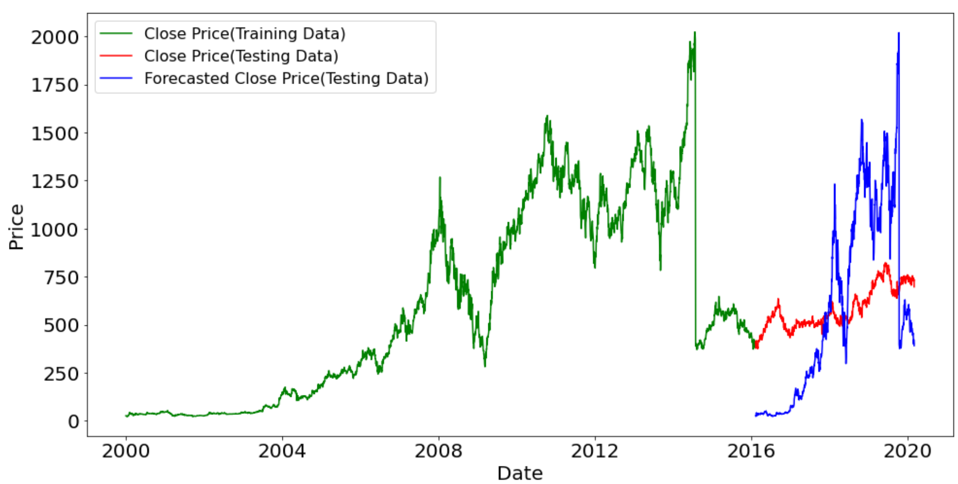

Fig.1: Result of *k*-NN model for dataset1-Axis Bank

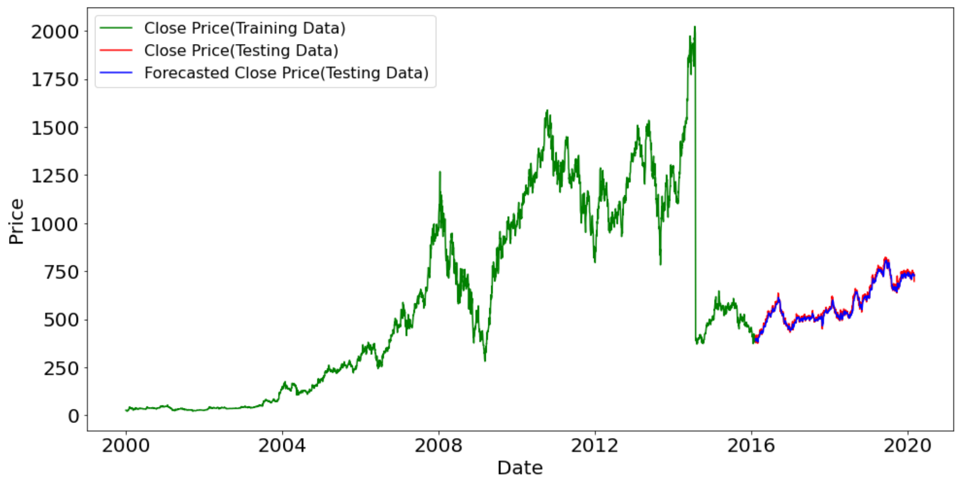

Fig.2: Result of LSTM model for dataset1-Axis Bank

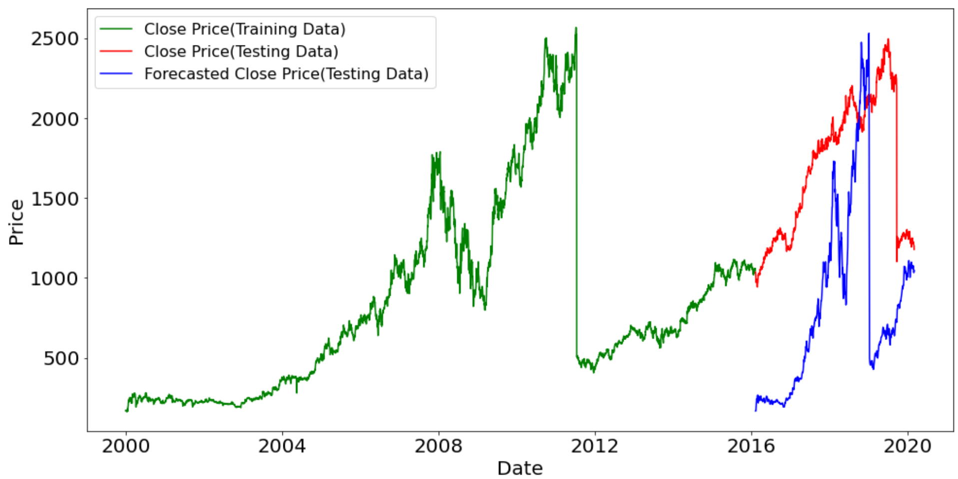

Fig. 3: Result of *k*-NN model for dataset2-HDFC Bank

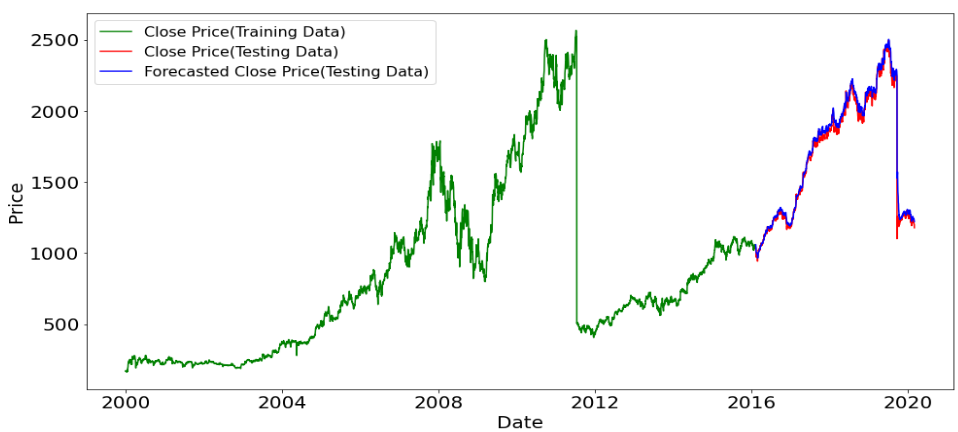

Fig.4: Result of LSTM model for dataset2-HDFC Bank

The testing data and the forecasted data i.e., the unseen data in Fig. 1 depicting *k*-NN for dataset1 shows that they are apart and only in a few areas, the lines seem to be close to each other. By using LSTM model for dataset1, the testing and the unseen data in Fig. 2 is very closely related and explicitly shows high level of accuracy attained. Comparing both the models for dataset1 with the testing and unseen data of Fig.1 and the testing and unseen data from Fig.2, graphically LSTM shows only a minimal difference between the testing and unseen data. In predicting the unseen data for dataset2 by *k*-NN model with testing data, as seen in Fig. 3, the distance between the lines are far which shows the level is not in the desirable range when compared to Fig. 4, which shows that the prediction level for dataset2 is quite high. It can be inferred while comparing Fig. 3, the LSTM have modeled the testing data accurately as seen in Fig. 4. For both the datasets, graphically it is seen that LSTM have a high level of accuracy than *k*-NN.

4. Conclusion

There are many technical analyses in stock exchange and Support and Resistance levels are found out to follow where the trend lines are directed. Prediction is especially very difficult to know the direction of these trend lines and the application of knowledge to make decisions on buying and selling based on predictions is very risky for the users. The advent of deep learning has in a way helped overcome risks based on predictions. The level of accuracy for predicting the values is high for LSTM than *k*-NN which was assessed using five different error metrics. From the above discussion it is known that the training time taken by *k*-NN is preferred against LSTM but when the cross-validation is set to high in *k*-NN, the performance is affected and the training time taken by *k*-NN is directly proportional to cross-validation. Considering this factor, the computation time differs with the level of accuracy attained which depends on how the training and the testing data are split and the number of training it is set to undergo. When the above factor is taken up to find the best fit, LSTM is a desirable model to predict the support and resistance levels where the entry/exit of stocks can be analyzed. This research predicted the support and resistance levels through Fibonacci retracement using two different models, inferred that the best fit for both the datasets is LSTM and the work can be extended further by predicting multiple technical factors at the same time in future.

Acknowledgement

We would like to thank D. G. Vaishnav College, Chennai and Madras University which provided moral support for research work and in developing and comparing deep learning model, Long Short-Term Memory and a machine learning method, *k*-NN for predicting support and resistance indicators in stock markets.

References

[1] S. Soni, "Applications of ANNs in stock market prediction: a survey", InternationalJournal of Computer Science & Engineering Technology, Vol 2, No. 3, pp. 71-83, 2011.

[2] A. Thakkar and K. Chaudhari, "Fusion in stock market prediction: A decade survey on the necessity, recent developments, and potential future directions", Information Fusion, pp. 95-107, 2020, DOI.org:/10.1016/j.inffus.2020.08.019.

[3] B. Karabulut, G. Arslan and H. M. Ünver, "A Weighted Similarity Measure for k-Nearest Neighbors Algorithm", Celal Bayar University Journal of Science, Vol.15, Issue. 4, pp. 393-400, 2019, DOI: 10.18466/cbayarfbe. 618964.

[4] V. B. Prasath, H. A. A. Alfeilat, O. Lasassmeh and A. B. A.Hassanat, "Distance and Similarity Measures Effect on the Performance of K-Nearest Neighbor Classifier–A", arXiv preprint arXiv:1708.04321, 2017.

[5] G. Guo, H. Wang, D. Bell, Y. Bi and K. Greer, "KNN model-based approach in classification", In OTM Confederated International Conferences" On the Move to Meaningful Internet Systems", pp. 986-996, Springer, Berlin, Heidelberg, 2003, DOI.org/10.1007/978-3-540-39964-3_62.

[6] M. Aci, C. İnan, and M. Avci, "A hybrid classification method of k nearest neighbor, Bayesian methods and genetic algorithm", Expert Systems with Applications, Vol. 37, No. 7, pp. 5061-5067, 2010, DOI:10.1016/j.eswa.2009.12.004.

[7] T. Ban, R. Zhang, S. Pang, A. Sarrafzadeh and D. Inoue, "Referential knn regression for financial time series forecasting", In International Conference on Neural Information Processing, pp. 601-608. Springer, Berlin, Heidelberg, 2013.

[8] F. H. Al-Qahtani, and S. F. Crone, "Multivariate k-nearest neighbour regression for time series data—A novel algorithm for forecasting UK electricity demand", In The 2013 international joint conference on neural networks (IJCNN), IEEE, pp. 1-8, 2013.

[9] F. Martínez, M. P. Frías, F. Charte and A. J. Rivera, "Time Series Forecasting with KNN in R: the tsfknn Package", The R Journal, Vol. 11, No. 2, pp. 229-242, 2019.

[10] N. Zhang, A. Lin and P. Shang, "Multidimensional k-nearest neighbor model based on EEMD for financial time series forecasting", Physica A: Statistical Mechanics and its Applications, Vol. 477, pp. 161-173, 2017, dx.doi.org/10.1016/j.physa.2017.02.072.

[11] K. Smagulova and A. P. James, "A survey on LSTM memristive neural network architectures and applications", The European Physical Journal Special Topics, Vol. 228, No. 10, pp. 2313-2324, 2019, DOI: 10.1140/epjst/e2019-900046-x.

[12] Z. Huang, W. Xu and K. Yu, "Bidirectional LSTM-CRF models for sequence tagging", arXiv preprint arXiv:1508.01991, 2015.

[13] L. Verwimp, J. Pelemans, and P. Wambacq, "Character-word LSTM language models", arXiv preprint arXiv:1704.02813, 2017.

[14] G. Lai, W. C. Chang, Y. Yang, and H. Liu, "Modeling long-and short-term temporal patterns with deep neural networks", In The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, pp. 95-104, 2018, DOI.org/10.475/123_4.

[15] S. Ravuri and A. Stolcke, "Recurrent neural network and LSTM models for lexical utterance classification", In Sixteenth Annual Conference of the International Speech Communication Association, Germany, 2015.

[16] Y. Luan, Y. Ji, and M. Ostendorf,"LSTM based conversation models", arXiv preprint arXiv:1603.09457, 2016.

[17] M. Hiransha, E. A. Gopalakrishnan, V. K. Menon and K. P. Soman, "NSE stock market prediction using deep-learning models", Procedia computer science, Vol. 132, pp. 1351-1362, 2018, DOI: 10.1016/j.procs.2018.05.050.

[18] G. Ding and L. Qin, "Study on the prediction of stock price based on the associated network model of LSTM", International Journal of Machine Learning and Cybernetics, Vol. 11, No. 6, pp. 1307-1317, 2020, DOI.org/10.1007/s13042-019-01041-1.