INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

LinVec: A Stacked Ensemble Machine Learning Architecture for Analysis and Forecasting of Time-Series Data

Unmesh Mandal[1], Anirban Chakraborty[2]*, Phulen Mahato[2], Goutam Das[3]

1 Assistant Professor, Dept. of Computer Science, Bidhan Chandra College, Rishra, 31, G. T. Road (East), Rishra, Hooghly, 712248, West Bengal, India
2 Assistant Professor, Dept. of Computer Science, Barrackpore Rastraguru Surendranath College, 85 Middle Road, Kolkata, 700120, West Bengal, India
3 SACT-I, Dept. of Computer Science, Bidhan Chandra College, Rishra, 31, G. T. Road (East), Rishra, Hooghly, 712248, West Bengal, India

OPEN ACCESS

Received: 15-11-2022
Accepted: 17-01-2023
Published: 27-02-2023

Citation: Mandal U, Chakraborty A, Mahato P, Das G (2023) LinVec: A Stacked Ensemble Machine Learning Architecture for Analysis and Forecasting of Time-Series Data. Indian Journal of Science and Technology 16(8): 570-582. https://doi.org/10.17485/IJST/v16i8.2197

* **Corresponding author**.

theanirban.chakraborty@gmail.com

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

Objectives: The proposed work integrates multiple Machine Learning approaches in a single model to be used for the analysis and forecasting of Time Series data. **Methods:** In the present work, the concept of Stacked Ensemble learning is proposed that uses Multiple Linear Regression and Support Vector Regression techniques as the base models. A Meta Model is constructed based on Multiple Linear Regression with necessary modifications. The outputs from the base models are fed into the meta-model which is mended with the capability of combining the predictions from the two base models to produce better results than the individual constituent parts, after running a k-fold training procedure. **Findings:** The proposed model is capable of analyzing and predicting any Time Series data. In the present study, stock data of six companies enlisted in the National Stock Exchange of India are analyzed for the prediction of the next day's Open, High, and Low prices. The proposed work achieves better accuracy and reduces the error in prediction when compared to similar works done in the same field. **Novelty:** The amalgamated technique used in this work can be considered as a generalization of the stacked ensemble method in a broader aspect. The proposed model combines the strengths of multiple Machine Learning methods into a single model to achieve better performance than its individual counterparts. Further, several recent works have tried to predict only the next day's Open and Closing Prices of stocks, but for an intraday trader, prediction of the next day's Low and High prices of a stock are more significant than the closing prices. Very few works have predicted all of the Open, High and Low prices in a single study, our present work achieved this quite successfully.

Keywords: Machine Learning; Stacked Ensemble Model; Support Vector Regression; Multiple Linear Regression; Time Series data analysis; Stock Price Prediction

1 Introduction

There exists large number of works utilizing the concept of Machine Learning (ML) and Deep Learning (DL) approaches for the analysis and forecasting of time series data. Autoregressive Integrated Moving Average or ARIMA, Support Vector Regression (SVR) and Random Forest (RF) are some common ML techniques which are popular in this field. In the DL domain multiple works have been done using different Recurrent Neural Networks (RNN) such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU). Some other DL techniques like Deep Neural Networks (DNN), Convolutional Neural Network (CNN) are also popular[(1)] proposed a hyperparameter selection strategy for the ARIMA model for analysis and forecasting of time series data. They fused the concepts of Differential Evolution and Artificial Bee Colony algorithms in their proposed method[(2)] proposed a method for stock price forecasting on daily and up to the minutes data using SVR. The work suggested that the SVR can outperform the Random Walk Model in predicting Time Series data[(3)] predicted the next day closing price of Stocks using Artificial Neural Network (ANN) and RF techniques. The model was tested on stock prices of five companies. According to their results the ANN model outperformed the RF model. Y. Li,[(4)] studied the online user-generated content on the stock message board of eastmoney.com and tried to predict the effect of investor sentiment and expectation on stock price. They used the LSTM, Logistic Regression, Support Vector Machine, and Na¨ıve Bayes models to compare the results. X. Zhang et al.[(5)] predicted stock price movements by employing nonlinear models based on the sentiment and perception of the tweets from Xueqiu (Chinese Twitter-like social platform specialized for investors). Kumar et al.[(6)] designed a stock price prediction model using Radial Basis Function Networks (RBFN). They compared the results with Multi-Layer Feed Forward Neural Network (MLFFNN). In their study the RBFN model outperformed the MLFFNN model. Shah et al.[(7)] implemented a Deep Learning framework for stock price forecasting. They paired CNN with LSTM in a single model for the prediction of Nifty 50 index closing price. The CNN was used to extract the useful features and the LSTM part did the actual predictions.

Despite the presence of multiple works in the same domain, there exist some research gaps. The linear models like ARIMA perform well in capturing the linear relations in the data set. But they are quite inefficient in handling the non-linear relationships hidden in the historical time series data. Some recent works tried to take advantage of DL based models in the field of time series analysis. However, DL techniques work better on larger datasets with large number of features[(8)]. Whereas the Machine Learning techniques can produce satisfactory results even with small dataset with limited dimension[(9)]. Some of the previous works considered single Machine learning models such as SVR, RBFN, Ridge Regression, Lasso regression etc. for time series analysis. However, a single model lacks the versatility of learning. It has been observed that Ensemble Techniques help to improve overall performance by combining multiple Sub Models into a single hybrid model[(10)]. The higher performance is achieved by combining the strengths of multiple models and thus eliminating the weaknesses of individual counterparts.

Further, it has been observed that, prediction of closing and opening prices of stocks been the key interest area for most of the researches done in this field. But practically, prediction of the next day's Open, High and Low prices will help a trader for proper planning of an intraday trade more effectively. Thus, there is a need to setup a framework that takes the advantages of ML and Ensemble techniques and can predict the said parameters with significant accuracy to minimize risks and maximize the profits of an intraday trader.

In the current work, an amalgamated ML approach, named LinVec (borrowed from the terms Multiple Linear Regression and Support Vector Machine), capable of capturing both the linear and non-linear relationships present in the dataset is proposed. The major driving force behind the proposal of such an Ensemble technique is that it improves the performance of the model when compared to its individual counterparts. In the proposed Ensemble method, two popular ML techniques, SVR and Multiple Liner Regression (MLR) are used as base models. Both of these base models are trained on the same data from the training set. Each model learns differently and produces separate sets of outputs. The results obtained from the base models, along with corresponding real target values are next fed to the meta-model, which is another MLR model with some necessary modifications. The meta-model is trained on the newly created data set using k-fold cross-validation and becomes capable of combining the predictions from the two base models to improve the results further. Considering more than a thousand test cases it has been inferred that the present amalgamate technique produces better results than its comprised individual parts.

The proposed model is capable of handling any Time Series data. The model analyses the dataset and tries to find the hidden time sensitive relationships among the features and the targets. The model may be trained on the most recent data available to forecast the future values for certain targets. To justify the capability of the proposed framework it is tested in the field of Stock Price prediction. The model can be trained on the historical price movement data available till the end of the present-day market to forecast the next day's price movements. Feature selection plays an important role in the preparation of an ML model capable of analyzing time series data. The price movement features of an individual stock, like daily open, close, high, low prices, delivery percentage, etc. and some features from the key as well as sectoral indices are taken into consideration. By analyzing all these features, the proposed model makes a prediction of the next day's price movement of an individual stock. The optimum values of the evaluation metrics justify the accuracy of the predictions and efficiency of the proposed model. The outcomes of

the proposed method are compared with similar work done by Henrique et al.[2].

The remaining part of the paper is organized as follows. Section 2 deals with the preliminaries. Section 3 describes the methodology. Section 4 presents the empirical results of the approach and the comparison of the results with a previous work done in the same field. Finally, the conclusions and future directions of the research are discussed in Section 5.

2 Preliminaries

2.1 Data Acquisition

The proposed model is designed to handle all types of data which are sensitive to time. In this work Stock Price movement, which is one of the most popular time series problem, is taken into consideration. Stocks of six companies are studied from the Indian Stock Exchanges. We have chosen two companies from each of the Large Cap, Mid Cap and Small Cap categories, resulting in six as a whole. Large-cap companies have well-established businesses and have a significant market share. They have market caps of INR 20,000 crore or more. Mid-cap companies are companies whose market cap is above INR 5,000 crore but less than INR 20,000 crore approximately, whereas the companies whose market capitalisation is of less than INR 5,000 crore are generally put in the Small Cap category. The National Stock Exchange of India (NSE) has many indices to provide information about the price movements of stocks and different forms of investments for the companies listed on Indian Stock Exchanges. Stock market indices are meant to capture the overall behaviour of equity markets[11]. NSE has key indices like the Nifty 50 Index[12] to track stock prices of diversified top 50 largecap companies which cover 13 sectors of the economy. The exchange (NSE) also has the NIFTY Midcap 50 Index[13] to track the top 50 midcap companies in India. It also employs NIFTY Smallcap 50[14] and NIFTY Smallcap 250[15] to capture the movement of the small-cap segment of the market. The daily key index data helps to find the overall sentiment of the market and thus can be useful to predict the momentum of an individual stock.

Another area of interest while considering stock market data is the sectoral indices. The sectoral index data provides an idea of how a particular sector is performing. If all the related stocks of a sector are performing well, it signifies that the demand for that sector is increasing. It is then very likely that the stock of our interest from the same sector will also grow. NSE has multiple sectoral indices[16] those track stocks from different sectors. For the present work, we have chosen companies from different sectors. We have studied stocks of six companies namely Reliance Industries Limited, Ashok Leyland Ltd., Birlasoft Ltd., Sun Pharmaceutical Industries Ltd., AU Small Finance Bank Ltd. and Sobha Ltd., all of which are listed in the NSE. They are chosen from the Large Cap (Bluechip), Mid Cap, and Small Cap categories based on their total market capitalization. Required historical data of individual stocks[17] and indices[18] are collected from the NSE website. For each of these companies, historical data are acquired in different time frames lying between 2018 to 2022, as reflected in Table 1, along with the corresponding category, key index, sectoral index etc.

Table 1. Companies from different categories with corresponding sectors

Company Name (NSE Code)	Key Index	Sectoral Index	Time Frame		#Samples
Reliance Industries Ltd. (RIL)	NIFTY 50	Nifty Oil & Gas	12-Feb- 2020	09-Feb- 2022	496
Ashok Leyland Ltd. (ASHOKLEY)	NIFTY MIDCAP 50	Nifty Auto	01-Jan- 2018	30-Dec- 2019	490
Birlasoft Ltd. (BSOFT)	NIFTY SMALLCAP 50	None (NIFTY IT to be considered)	01-Jan- 2020	30-Dec- 2021	499
Sun Pharmaceutical Industries Ltd. (SUNPHARMA)	NIFTY 50	Nifty Pharma	01-Apr- 2020	05-Apr- 2022	500
AU Small Finance Bank Ltd. (AUBANK)	NIFTY MIDCAP 50	Nifty Bank	01-Jan- 2020	31-Dec- 2021	500
Sobha Ltd. (SOBHA)	NIFTY Smallcap 250	Nifty Realty	04-May- 2020	13-May- 2022	507

For the present study, approximately a hundred features from individual stocks, sectoral and key indices have been considered and tested on the model. After testing rigorously, only twelve features, which provide optimized RMSE, MAPE and R^2 values are selected. 10 out of these 12 features are considered from the properties of individual stock which are: Daily Open Price, High Price, Low Price, Last Price, Close Price, Average Price, Total Traded Quantity, No. of Trades, Deliverable Qty. and Delivery Percentage. The other two features are taken from the indices. They are the Daily Open Values of the Key Index and the Daily Close Values of the Sectoral Index.

The data set has 3 target variables: Next Day's Open, High and Low prices. The features and the targets of the data set are tabulated by using the following rules. The next day's open, high and low prices of a stock are considered as the target for the

present day. During the learning phase, the model analyses all the features of present-day and the next day's open, high and low prices, as the real target values and tries to find the relation among them (targets and features). During the test phase, the model only takes the present-day data and tries to predict the next day's real price values of that stock.

2.2 Data Standardization

The prepared data set has different features with different numerical ranges. For example, the stock values are in the range of hundreds or thousands (INR), whereas the numbers of trades and deliverable quantity values are in the range of Lakhs or Crores. Due to this heterogeneity, some features may dominate others during the training procedure. To avoid this problem all the features are needed to be brought down to a common scale. For this purpose, data standardization is performed all of the features based on the following formula:

$$z = \frac{(x-u)}{s} \tag{1}$$

Where the standard score of a sample is x, u represents the Mean of the training samples, and s represents the Standard deviation of the training samples.

2.3 Evaluation metrics

To evaluate the price prediction performance of the proposed model, we used three evaluation metrics: Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE) and R^2 (coefficient of determination) regression score

The RMSE, MAPE and R^2 Score can be calculated according to Eqns. 2, 3 and 4 respectively.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \widehat{x_i})^2}{N}} \tag{2}$$

Where, RMSE=Root Mean Square Error, N=Number of non-missing data points x_i=i_{th} real sample value $\widehat{x_i}$=i_{th} estimated value by the model

$$MAE = \frac{1}{n}\sum_{t=1}^{n} |\frac{A_t - F_t}{A_t}| \tag{3}$$

Where, MAPE=Mean Absolute Percentage Error, n=number of non-missing samples, A_t=Actual Value, F_t=Forecast Value

$$R^2 Score = 1 - \frac{SS_{res}}{SS_{tot}} \tag{4}$$

Where, SS_{res}=Sum of squares of the residual errors, SS_{tot}=Total sum of the errors

3 Methodology

We propose a hybrid model based on Stacked Ensemble Machine Learning technique that implements the concept of Stacked Generalization. The implementation utilizes Support Vector Regression (SVR) and Multiple Linear Regression (MLR) as the base models and another MLR as the Meta model. The base and Meta models are trained on the same training set. The two base models get trained differently and learn distinct properties of the same dataset. The Meta model is utilized to combine the predictive capability of both of the base models. The final predictions are observed from the Meta model, which are more accurate than the predictions of the individual base models.

3.1 Stacked Generalization

The Stacked Generalization or Stacking is an ensemble machine learning algorithm, where the meta-model learns how to combine the outputs produced by the base models in an efficient way so that the final model outperforms the individual base models. This technique uses two or more heterogeneous Base Models/Estimators fit on the same training data. These base models get trained on the same data but capture different aspects/facts. Here the problem is which among the base models to trust. The solution is to use another machine learning model, called Meta Model/Final Estimator, that learns which base model to trust more and when. Firstly, all the base models are trained on the same training data. The predictions from the base

models become the features and the expected outputs become the target of the training data set for the meta-model. The base estimators are trained on the whole training data. But the final estimator is generally trained using the K-Fold-Cross-Validation technique. In this technique, the whole data set is split into K consecutive non-overlapping groups/folds. Each fold is then used as a validation set and the remaining K-1 groups are used as training data sets for a model.

3.2 SVR for Stock market's trend analysis

Support Vector Machine (SVM) is popularly and widely used for classification problems in machine learning. A classification method based on SVM maps the independent variables of N samples available into a space of more dimensions and is typically used to classify observations between groups. This method uses $\{(X_k, y_k)\}_{k=1}^{N}$, training observations to build a linear model using non-linear classification thresholds, mapping variables on a greater number of dimensions. The separation between classes is achieved using an optimal hyperplane, calculated based on N observations, where X is the independent variable vector and y is the classification $y_k \in \{-1, 1\}$ for each sample. The classification hyperplane represented in Eq. 5, satisfies the conditions stated by Eqs 6 and 7.

$$W^T \phi(X_k) + b = 0 \tag{5}$$

$$W^T \phi(X_k) + b \geq 1 para y_k = 1 \tag{6}$$

$$W^T \phi(X_k) + b \leq -1 para y_k = -1 \tag{7}$$

In our study, the objective is to predict the future prices of Stocks. Thus, the goal is not to classify the results into groups, but rather to estimate the real values. Therefore, we use SVR to obtain a regression model. In any regression problem, we try to estimate a function that approximates mapping from an input domain to real numbers, on the basis of training samples. SVR is a supervised learning algorithm that can be used to solve regression problems. It can handle linear and non-linear regression problems both.

SVR can transform a nonlinear regression problem into linear regression through the implementation of a kernel function, which projects the original feature space into a higher-dimensional space. A hyper-plane is then used to fit the projected space and the estimated parameters can be used for subsequent prediction. SVR uses the same principle as the SVMs. It gives us the flexibility to define how much error is acceptable in our model and will find an appropriate line (or hyperplane in higher dimensions) to fit the data. In simple regression, we try to minimize the error rate but in the case of SVR a certain threshold bound is used to fit the error inside. SVR tries to approximate the optimized value within a specified margin called ε (epsilon)-tube. The objective function and constraints of the SVR model are as follows:

$$Minimize : \frac{1}{2} |W|^2 \tag{8}$$

$$Constraints : |y_i - w_i x_i| \leq \tag{9}$$

Where W is a vector normal to the hyperplane. y_i is the target, w_i is the coefficient, and x_i is the predictor (feature).

Slack variables are introduced in SVR to relax the stiff conditions. The new objective function and constraints, after the addition of the slack variables, are as follows:

$$Minimize : \frac{1}{2} (|W|^2 + C \sum_{i=1}^{n} |\xi_i| \tag{10}$$

$$Constraints : \{|\{y_i - w\}| \leq \varepsilon + |\xi_i|\} \tag{11}$$

The constant C is the regularization parameter. The strength of the regularization is inversely proportional to C. It should be strictly positive.

3.3 Multiple Linear Regression (MLR)

MLR is an extension of Linear regression (LR) that uses multiple independent variables as input and produces the value of one dependent variable as output using the following formula:

$$y = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \cdots + b_nx_n \tag{12}$$

Where, y = Dependent Variable, b_0 = Y-Intercept (constant term), $x_1, x_2..., x_n$ = Independent Variables, $b_1, b_2..., bn$= Regression Coefficients/slopes.

3.4 Architectural overview of the proposed model

The proposed method uses an amalgamated model based on Stacked Generalization. The overview of the architecture is shown inFigure 1. The proposed model has two layers. The first layer (Level 0) comprises of two base ML models. One is based on SVR and another is based on MLR. The final layer (level 1) contains a meta-model based on MLR. Final predictions are produced by the meta-model. We have tested many combinations of configurations on our data set. As no standard mechanisms exist for fixing up the values of hyperparameters, hence we fed different inputs and following a trial-and-error approach we fix them to generate the most accurate results.

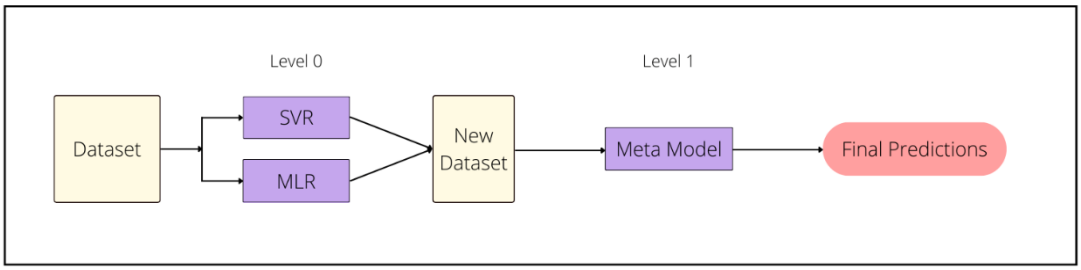

Fig 1. Architecture of the proposed method

During the training, the whole train data set is fed to both of the base models. The meta-model is trained on the cross-validated predictions of the base models using the k-fold cross-validation technique. The SVR model present at level 0 is trained on the full training data set. The kernel is set to Linear. The regularization parameter C is set to 1.0 and epsilon (decision boundaries from the hyperplane) is set to 0.1. During training the attributes of the model i.e. the support vectors, y-intercept value and 12 coefficients (each for one feature in the data set) are calculated. The second model in level 0 which is based on MLR is also trained with the whole training data set. In this model, values of 13 parameters: one y-intercept and twelve coefficients for 12 features present in the data set are needed to be tuned.

The MLR meta-model is trained on the prediction of the base models generated using k-fold cross-verification technique. First, the whole training data set is divided into almost identical k groups/folds. In a single iteration one group out of this k folds is considered as the validation data set and the remaining k-1 groups are considered to be the training data set. First, the base learners are trained on the training set (selected from the k-folds) and then the predictions are generated on the validation data set. This process is repeated for the remaining k-1 folds. Each time the predictions from the base learners are stacked to create an augmented data set. This augmented data set along with the real target values are used as the training data set for the meta-learner/final estimator.

During the training on the augmented data set, the meta-model calculates a single y-intercept and two regression coefficients (one for each base model prediction). The value of k plays a crucial role in generating accurate results, but unfortunately, no concrete method is there for deciding it. Hence, starting with k=2, a type of trial-and-error method was followed and found that k=25 produces the best result for our model. The procedure of training/testing of the models and finding of the value of k are detailed in the pseudocode depicted in Figure 2.

Pseudocode: LinVec

- Initialize n //Number of records in the Training Dataset
- Initialize k=2 //for k-fold Cross Validation
- Repeat till k<=n

//Training Process Begins

- Divide the scaled Training Dataset (TD) into identical k numbers of subgroups
- Initialize i=1, j=1
- Initialize AUGSDTR=NULL
  //AUGSDTR=Augmented Stacked Dataset for training the Meta Model
- Repeat till i <= k
  - Consider TD_i as Validation Set and $TD_{j\ to\ k\ (j!=i)}$ as training set
  - Train SVR model at Level 0 with $TD_{j\ to\ k\ (j!=i)}$ datasets
  - Train MLR model at Level 0 with $TD_{j\ to\ k\ (j!=i)}$ datasets
  - Record the predictions of SVR model at Level 0 on validation set TD_i as $SVRTrainPR_i$
  - Record the predictions of MLR model at Level 0 on validation set TD_i as $MLRTrainPR_i$
  - Stack $SVRTrainPR_i$ and $MLRTrainPR_i$ and append with AUGSDTR
  - Set i=i+1
- Train the Meta model on AUGSDTR dataset with real targets

//Testing Process Begins

- Set AUGSDTST=NULL
  //AUGSDTST=Augmented Stacked Dataset for testing the Meta Model
- Feed the test data set to SVR model at Level 0 and record the predictions as SVRTestPR
- Feed the test data set to MLR model at Level 0 and record the predictions as MLRTestPR
- Stack SVRTestPR and MLRTestPR and save in AUGSDTST
- Make final predictions on AUGSDTST with the Meta Model
- Calculate the RMSE, MAE and R^2 Score, based on the comparison of the final predictions with
  the real targets of the test set
- The Evaluation Metrics along with value of k are recorded
- Set k=k+1

- That particular value of k is chosen corresponding to which the optimum RMSE, MAE and R^2 Score are produced

Fig 2. Pseudocode to describe the Training/Testing procedure of the proposed work

4 Results and Discussion

The steps followed during the experiments are depicted in Figure 3. The data are collected from NSE website and preprocessed before feeding to the model. Feature Scaling is performed on the Train and test datasets to normalize all the values in the same ranges. The features are selected after hundreds of iterations of training and testing. The results from the training set are used to train the base and the meta-model. Next, the proposed hybrid model is tested with the test data instances to predict the next day's Open, High and Low prices. Predictions of the next day's price values are performed for the six companies belonging to the Large, Mid and Small Cap categories. The proposed model is able to reduce the RMSE and MAPE values and improve the R^2 Score significantly for all of the companies belonging to different categories

For the sake of simplicity, the plots consisting of Real and Predicted prices for three companies each from Large, Mid and Small-cap categories are considered. Figures 4, 5 and 6 represent the Real vs. Predicted Next day Open, High and Low prices of Reliance Industries Ltd. which belongs to the Large Cap Category.

Figures 7, 8 and 9 represent the real and predicted prices of the next day's Open, High and Low prices of Birla Soft Ltd. respectively. The company belongs to the Mid Cap category.

Figures 10, 11 and 12 represent the real and predicted prices of the next day's Open, High and Low prices of Ashoke Leyland Ltd. respectively. The company is in the Small Cap category.

The closeness of the real and predicted values in all of the plots signifies that the proposed model is strong enough to find the hidden time series patterns in the daily stock price movement data and can predict the next day's prices with significant accuracy across all of the Large, Mid and Small-cap categories yielding practical benefits to the traders and investors.

Table 2 depicts the obtained RMSE, MAPE and R^2 Scores, based on predicted and real values. The closeness of the real vs. predicted values in the plots along with low values of RMSE, MAPE and high value of R^2 Scores signify the effectiveness of the

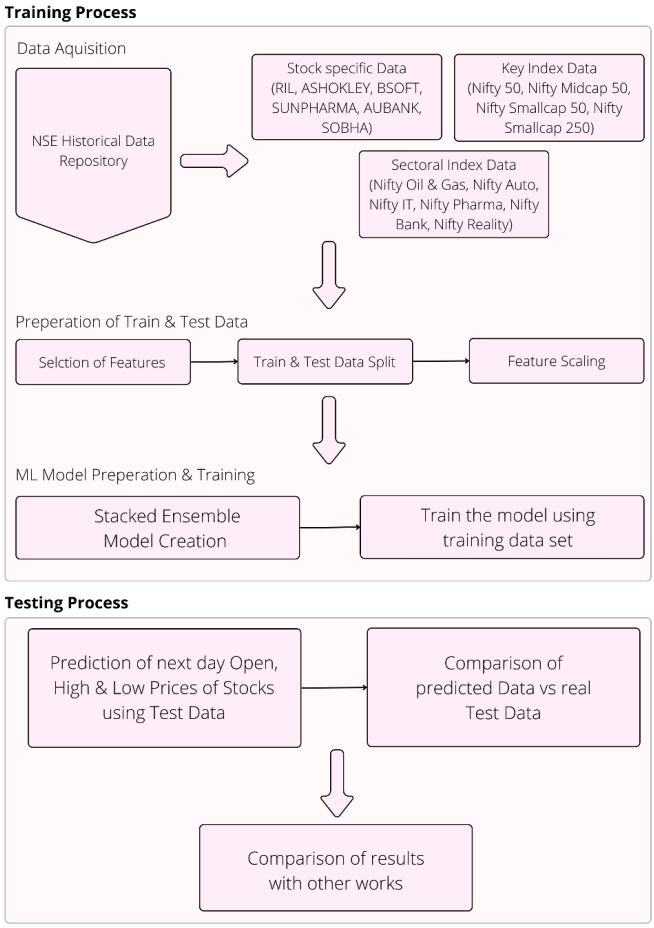

Fig 3. Steps followed during the Experiments

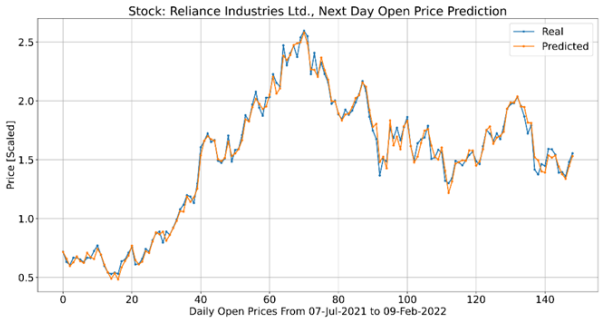

Fig 4. Real vs. Predicted next-day Open price of RIL (Large Cap)

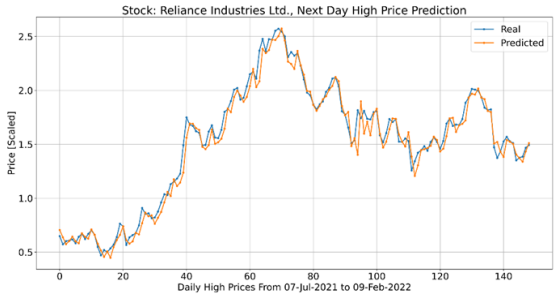

Fig 5. Real vs. Predicted next-day High price of RIL (Large Cap)

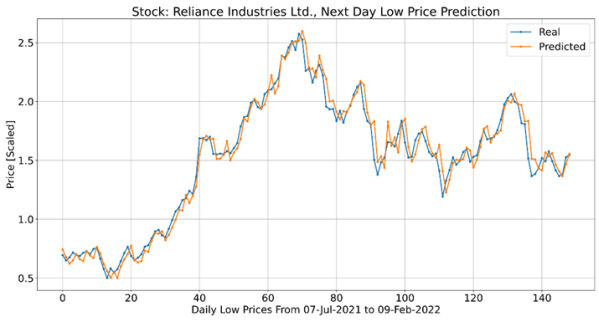

Fig 6. Real vs. Predicted next-day Low price of RIL (Large Cap)

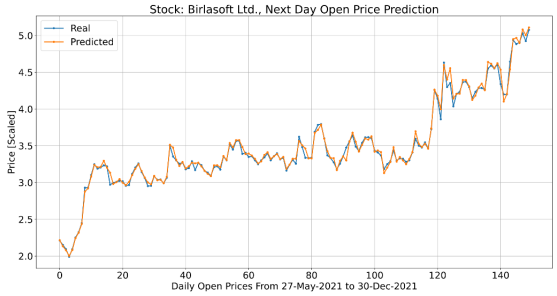

Fig 7. Real vs. Predicted next-day Open price of BSOFT (Mid Cap)

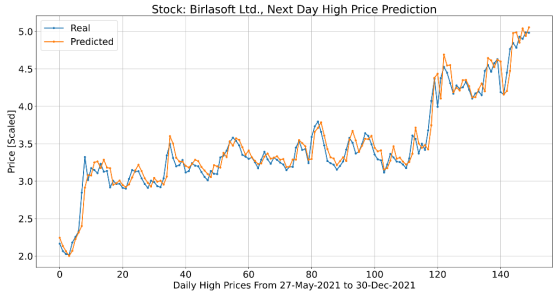

Fig 8. Real vs. Predicted next-day High price of BSOFT (Mid Cap)

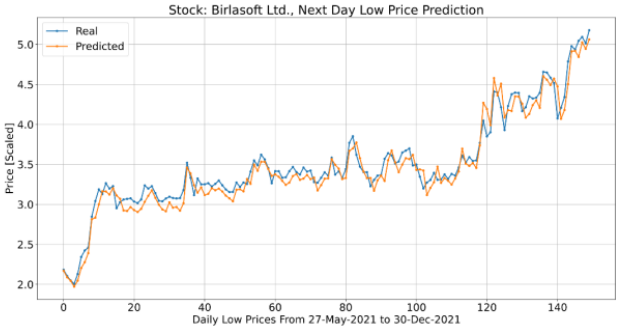

Fig 9. Real vs. Predicted next-day Low price of BSOFT (Mid Cap)

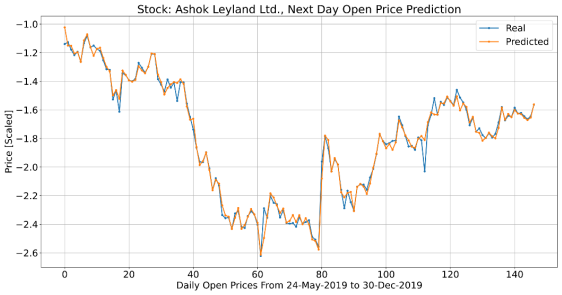

Fig 10. Real vs. Predicted next-day Open price of ASHOKLEY (Small Cap)

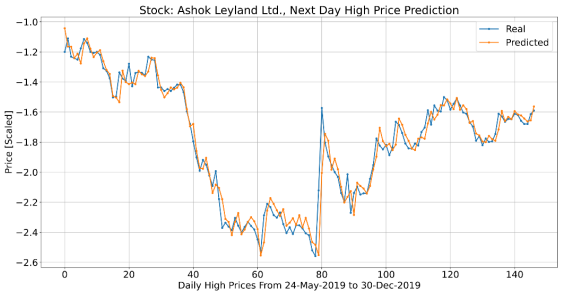

Fig 11. Real vs. Predicted next-day High price of ASHOKLEY (Small Cap)

proposed model.

Table 2. Results Summary of Stacked Ensemble Mode

Stocks	**Open Price**			**High Price**			**Low Price**		
	RMSE	MAPE	R2 Score	RMSE	MAPE	R2 Score	RMSE	MAPE	R2 Score
RIL	0.04737	0.02581	0.99208	0.08115	0.04106	0.97802	0.08571	0.04611	0.97251
ASHOKLEY	0.04065	0.01334	0.9897	0.08043	0.0302	0.95691	0.06485	0.02843	0.97507
BSOFT	0.04938	0.0088	0.99305	0.12986	0.02773	0.95297	0.11939	0.02836	0.95943
SUNPHARMA	0.05398	0.01612	0.98911	0.09647	0.02562	0.96535	0.08572	0.02709	0.97253
AUBANK	0.0559	0.02731	0.98014	0.09455	0.05311	0.94104	0.13632	0.06601	0.88183
SOBHA	0.06794	0.01933	0.9876	0.13121	0.04236	0.95435	0.13875	0.04654	0.9502

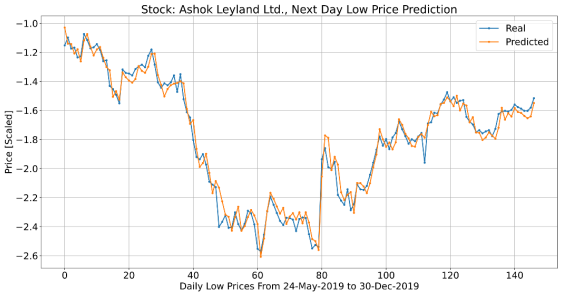

Fig 12. Real vs. Predicted next-day Low price of ASHOKLEY(Small Cap)

Table 3 depicts a comparative analysis between the proposed model, with the results obtained from the method proposed by Henrique et al.[(2)]. Table 3 signifies that in most cases our proposed model leads to a better prediction, resulting in better RMSE and MAPE scores, compared to the other technique considered. Furthermore, the model proposed by Henrique et al.[(2)] predicts only the daily prices of stocks. However, our proposed model leads to the prediction of Open, High and Low prices of a stock; providing a practical benefit to an intraday trader while choosing any stock for trade.

Table 3. Comparison of Results

Results by Henrique et al.[(2)] on Daily Price				**Results of the Proposed Work**			
Company Name	Country	RMSE	MAPE	Company Name	RMSE	MAPE	R2 Score
Banco do Brasil	Brazil	0.16427	0.20141	RIL	0.04737	0.02581	0.99208
Alpargatas	Brazil	0.08676	0.31555	ASHOKLEY	0.04065	0.01334	0.9897
Metal Leve	Brazil	0.14270	0.10383	BSOFT	0.04938	0.0088	0.99305
Angie's List	USA	0.10194	0.42575	SUNPHARMA	0.05398	0.01612	0.98911
Ping an Insurance	China	0.31341	0.31624	AUBANK	0.0559	0.02731	0.98014
IMAX China Holding	China	0.19135	0.27265	SOBHA	0.06794	0.01933	0.9876

Figure 13 depicts the pictorial representation of the comparison of RMSE values calculated on the results produced by Henrique et al.[(2)] and the proposed method. The diagram clearly reveals that the proposed method significantly reduces the RMSE values and outperforms the existing method with very large margins.

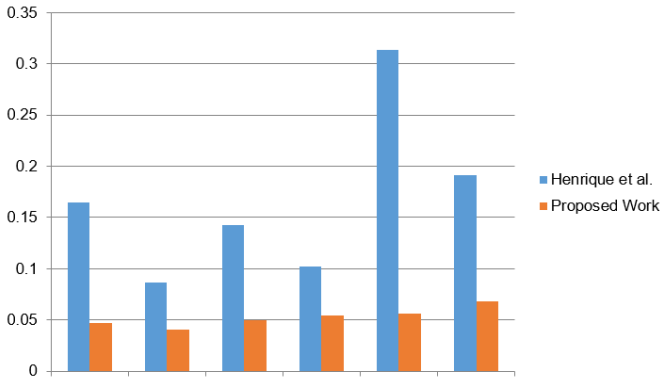

Fig 13. Comparison of RMSE values

Figure 14 represents the pictorial comparison of MAPE values produced by the proposed model with the results obtained by Henrique et al.[(2)]. The MAPE values are reduced significantly when compared to the existing method for all of the test cases. This clearly depicts the strength of the proposed model in terms of time series prediction over the existing work under consideration.

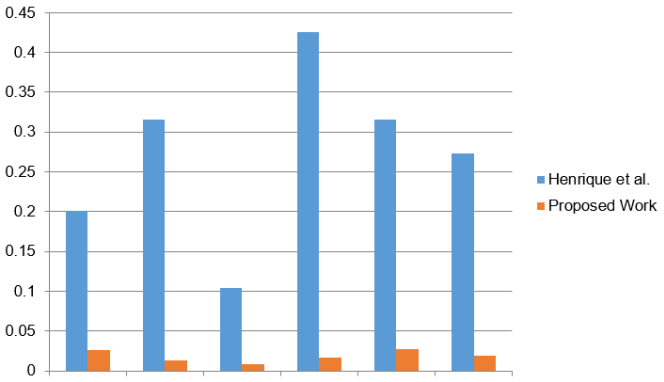

Fig 14. Comparison of MAPE values

5 Conclusion

Analysis of time series data such as predicting stock price movement is a challenging task due to constantly changing market conditions, which are dependent on multiple parameters resulting in very complex patterns. The data available at various stock exchanges help very little to predict the future behavior of that stock. However, our proposed model is efficient enough to find out and analyze the hidden patterns, hence producing quite accurate future predictions, resulting in low RMSE, MAPE values and high R^2 Scores.

The main influence on the daily stock price movement is the events or news related either to that company, to the sector to which the company belongs, or to any local or global news that can affect the price movement. Any such after-market news can abruptly change the building pattern in the stock price movement. If somehow, we can quantify the effect of this news and can include it in the study, maybe the ML models can provide better estimations.

Acknowledgement

The authors are grateful to the reviewers and facilitators whose constructive comments were useful in improving the content on this document. This work was supported by Bidhan Chandra College, Rishra and Barrackpore Rastraguru Surendranath College; by providing the platforms and means.

References

1) Kumar R, Kumar P, Kumar Y. Multi-step time series analysis and forecasting strategy using arima and evolutionary algorithms. *International Journal of Information Technology*. 2022;14(1):359–373. doi:10.1007/s41870-021-00741-8.
2) Henrique BM, Sobreiro VA, Kimura H. Stock price prediction using support vector regression on daily and up to the minute prices. *The Journal of finance and data science*. 2018;4(3):183–201. doi:10.1016/j.jfds.2018.04.003.
3) Vijh M, Chandola D, Tikkiwal VA, Kumar A. Stock closing price prediction using machine learning techniques. *Procedia computer science*. 2020;167:599–606. doi:10.1016/j.procs.2020.03.326.
4) Li Y, Bu H, Li J, Wu J. The role of text-extracted investor sentiment in chinese stock price prediction with the enhancement of deep learning. *International Journal of Forecasting*. 2020;36(4):1541–1562. doi:10.1016/j.ijforecast.2020.05.001.
5) Zhang X, Shi J, Wang D, Fang B. Exploiting investors social network for stock prediction in china's market. *Journal of computational science*. 2018;28:294–303. doi:10.1016/j.jocs.2017.10.013.
6) Kumar R, Srivastava S, Dass A, Srivastava S. A novel approach to predict stock market price using radial basis function network. *International Journal of Information Technology*. 2021;13(6):2277–2285. doi:10.1007/s41870-019-00382-y.
7) Shah A, Gor M, Sagar M, et al. A stock market trading framework based on deep learning architectures. *Multimedia Tools and Applications*. 2022;81:14153–14171. doi:10.1007/s11042-022-12328-x.
8) Lecun Y, Bengio Y, Hinton G. Deep learning. *nature*. 2015;521(7553):436–444. doi:10.1038/nature14539.
9) Zhang Y, Ling C. A strategy to apply machine learning to small datasets in materials science. *Npj Computational Materials*. 2018;4(1):1–8. doi:10.1038/s41524-018-0081-z.
10) Dietterich TG. Ensemble methods in machine learning. *In: International Workshop on Multiple Classifier Systems*. 2000;p. 1–15. Springer. doi:10.1007/3-540- 45014-9_1.
11) NSE-National Stock Exchange of India Limited: About Indices. . Accessed: 2022- 05-15. Available from: https://www1.nseindia.com/products/content/equities/indices/indiavix.htm.
12) NSE-National Stock Exchange of India Limited: NIFTY 50 Index. . Accessed: 2022- 05-20. Available from: https://www1.nseindia.com/products/content/equities/indices/nifty50.htm.

13) NSE-National Stock Exchange of India Limited: NIFTY Midcap 50 Index. . Accessed: 2022-05-20. Available from: https://www1.nseindia.com/products/content/equities/indices/niftymidcap50.htm.
14) NSE-National Stock Exchange of India Limited: NIFTY Smallcap 50 Index. . Accessed: 2022- 05-20. Available from: https://www.nseindia.com/products-services/indices-niftysmallcap50-index.
15) NSE-National Stock Exchange of India Limited: NIFTY Smallcap 250. . Accessed: 2022-05-20. Available from: https://niftyindices.com/indices/equity/broad-based-indices/nifty-smallcap-250.
16) NSE-National Stock Exchange of India Limited: Sectoral Indices. . Accessed: 2022-05-20. Available from: https://www1.nseindia.com/products/content/equities/indices/sectoralindices.htm.
17) NSE-National Stock Exchange of India Limited: Security-wise Archives (Equities). . Accessed: 2022-04-25. Available from: https://www1.nseindia.com/products/content/equities/equities/eqsecurity.htm.
18) NSE-National Stock Exchange of India Limited: Historical Index Data. . Accessed: 2022-05-15. Available from: https://www1.nseindia.com/products/content/equities/indices/historicalindexdata.htm.