FORECASTING IN-SAMPLE AND OUT OF SAMPLE STOCK MOVEMENT EMPLOYING MULTI RESOLUTION ANALYSIS AND NEURAL NETWORKS

Kush Bhushanwar[1*], Dr. Kuntal Barua[2]

Article History: Received: 15.04.2023 **Revised:** 01.06.2023 **Accepted:** 15.07.2023

Abstract

Forecasting stock market trends in often challenging due to the randomness and volatility of the nature of its movement owing to several factors. Often it is difficult to formulate a numerical counterpart for the diverse set of governing variables which can potentially impact stock movement. Moreover, collecting such data through indirect data mining techniques may often contain substantial biased and skewed opinions pertaining to a particular sample set. Thus, a naturally pragmatic approach seems to be designing a model trained on historical data, which is able to forecast both in and out sample. This would render high degree of robustness and practical utility to the developed model. This paper presents an approach to combine multi-resolution analysis with deep neural networks to forecast the future movement of stock markets. The multi-resolution analysis is used to remove effects of baseline noise inherent to time series stock datasets, followed by pattern recognition using deep neural networks. A diverse set of S & P datasets have been chosen for analysis. The forecasting accuracy, error rates and regression have been used compare the performance of the model against benchmark existing models.

Keywords: Stock Market Movement, Multi-Resolution Analysis, in sample forecast, Out of sample forecast, Mean Absolute Percentage Error (MAPE).

[1*]Research Scholar, Institute of Engineering, SAGE University, Indore (M.P.), India
[2]Associate Professor, Institute of Advance Computing, SAGE University, Indore (M.P.), India

[1*]kush.bhushanwar@gmail.com, [2]kuntal.barua@gmail.com

Corresponding Author: Kush Bhushanwar[1*]
Email: [1*]kush.bhushanwar@gmail.com
[1*]Research Scholar, Institute of Engineering, SAGE University, Indore (M.P.), India

DOI: 10.31838/ecb/2023.12.s2.541

1. Introduction

Stock markets is a multi-variate time series model which can be influenced by a variety of factors, including global parameters. For instance, Economic indicators such as GDP growth, unemployment rates, inflation, and consumer confidence in major economies can impact investor sentiment and influence stock prices [1]. Positive indicators may lead to bullish markets, while negative indicators can lead to bearish trends. Trade tensions and agreements between countries can impact international trade and corporate earnings, which in turn can affect stock prices [2]. Tariffs, import/export restrictions, and trade negotiations can create uncertainty in the market. Political instability, conflicts, and geopolitical events can lead to market volatility [3]. Uncertainty caused by such events can make investors more risk-averse and lead to market sell-offs [4]. Global economic health and growth prospects can impact investor sentiment. If the global economy is thriving, investors may be more willing to invest in stocks. Conversely, economic slowdowns can lead to market downturns. Often, investing requires understanding the intricate interplay of all these factors and their potential impact on stock prices [5]. One of the most critical challenges which most forecasting models face is the out of sample forecast, which often leads to higher error rates for fluctuating datasets. However, its is extremely important to achieve relatively high forecasting accuracy for out of sample forecasts as this would render robustness and reliability for the model to be used in practical scenarios. As influences of global stock markets impact a particular stock market movement in real time, hence it is challenging to estimate such movement trajectories in advance [6]. For instance, if we consider the Indian stock market, the opening for stocks are impacted by updates from the Tokyo stock exchange, in Japan as it is ahead with respect to time zones. Moreover, the impact of the London or New York stock exchange would also be felt from the previous day's closing patterns. Thus, the impact of both stock markets in time zones ahead (for next day) and in time lagged time zones (from previous day) would impact the movement of the stock market for companies listed in global stock markets. These influences often lead to drastic non-monotonic movement in stock trends which are hard to predict. This makes it even more necessary to design algorithms which can forecast out of sample, with relatively high accuracy, as the forecasting results are readily used for investment decisions [7].

Machine learning has gained significant attention in the field of stock market prediction due to its potential to analyze large amounts of data, identify patterns, and make informed prediction. As Stock markets generate vast amounts of data, including historical price movements, trading volumes, company financials, economic indicators, news sentiment, and more. Machine learning algorithms can handle the complexity of this data and identify non-linear relationships that traditional statistical methods might miss [8]. Kim et al. in [9] proposed an approach to compute the effective transfer entropy (ETE) among the various stock prices of companies listed globally to investigate the impact on one of them with regards to others. This approach enables to correlate the behavioural patterns of stock movement as a function of global influencing parameters. Bouktif et al. in [10] proposed opinion mining of social media and micro-blogging sites which release publicly available datasets to create a composite training vector for stock market prediction. Eapen et al. in [11] proposed the ensemble design of two individually effective deep learning models viz. the LSTM and the CNN, which was shown to outperform the individual models in terms of forecasting accuracy. Wen et al. in [12] proposed a variable cross entropy based approach akin to the transfer entropy model which correlates various global influencing parameters. Guo et al. in [13] proposed updating and optimizing the training weights of the support vector regression (SVR) model based on the particle swarm optimization. The dataset was not optimized, rather the SVR model's training was optimized through the PSO. Raimundo et al. in [14] employed recursive multi-resolution DWT analysis on stock time series data to be used in conjugation with the conventional SVR model. The aim of the approach was to filter out the baseline noise from the dataset. Baek et al. in [15] proposed an LSTM-LSTM cascaded ensemble with an aim to prevent overfitting of the latter LSTM model through optimization of data through the first LSTM module. Selvin et al. in [16] explored the stck market forecasting accuracy of several benchmark models comprising of autoregressive integrated moving average (ARIMA), recurrent neural networks (RNNs), GARCH and LSTM. Billah et al. in [17] employed the ubiquitous Levenberg Marquardt (LM) model for forecasting stock trends. The minimization of the cost function based on the computation of Jacobian matrix rather than the Hessian matrix was evaluated for speed of convergence as well as forecasting accuracy. . Sen et al. in [18] proposed a GARCH based model to forecast stock prices of diverse Indian Companies incorporating volatility modelling of stocks. Sharma et al. in [19] presented a hybrid genetic algorithm (GA) optimized neural network model (GANN) for forecasting stock movement. The approach showed that the proposed work outperformed the nack propagation based ANN (BPANN) model in terms of MAPE. Sharaf et al. in [20] proposed a hybrid bi-directional LSTM model in conjugation with Covid-19 pandemic sentiment analysis. The sentiment analysis was implemented based on news feeds available in public domain.

2. Proposed Methodology

The methodology developed in this paper tries to address the inherent challenges in stock market forecasting along with the research gaps in existing literature. Such limitations and challenges are identified asL

1) Time series stock market datasets are inherently noisy in nature, thereby making pattern recognition challenging.
2) The stock movement often shows high variability for in-sample and out of sample forecasts, where the latter typically results in higher error rates.
3) Data mining techniques applied to publicly available datasets often exhibit substantial skewed biases, thereby rendering a skewed connotation to the training vector in favour or against a particular stock.
4) Employing a training algorithm which can render high prediction accuracy for multiple benchmark datasets.

The methodology developed in this paper attempts to address all these issues, thereby attaining high forecasting accuracy.

Multi Resolution Analysis

Multi resolution analysis essentially entails splitting the dataset into lower and higher resolution (frequency) components so as to filter out baseline noise. One of the most effective multi-resolution analysis tools is the discrete wavelet transform (DWT) which separates the low and high resolution components in the transform domain. The (DWT) is a mathematical technique used in signal processing to decompose a signal into different frequency components. It has been applied to stock market prediction as a way to extract useful features from stock price time series data [21].The DWT breaks down a time series signal into different scales or levels of detail. It decomposes the original signal into approximation and detail coefficients at various scales. In the context of stock market data, the DWT can help to identify different frequency components in the price movement[22]. Thus, the DWT provides a multiscale analysis of the data. This can be particularly useful in stock market prediction, as different time scales may have different impacts on stock prices. For instance, short-term fluctuations and long-term trends could be captured through different wavelet scales [23].Thus combining DWT with other techniques, such as machine learning models and traditional financial analysis, to create a comprehensive prediction strategy. Mathematically, the DWT is defined as:

$$\mathbf{Z(x, k_{sc}, k_{sh}) = L^{n} \sum_{i} z(x) M \left[\frac{n - ik_{sh}}{k_{sc}}\right]} \quad \mathbf{(1)}$$

Here,
Z is denotes dataset in transform domain
z(x) denotes the dataset
k_{sc} denotes the scaling factor.
k_{sh} denotes the shifting factor.
n denotes the decomposition level
L^{n} denotes the dilation factor
M denotes the DWT kernel.

The essence of the DWT algorithm lies in the fact that the high resolution co-efficients (C_D) entail the baseline noise while the low resolution co-efficients (C_A) happen to entail the fundamental information content. Thus, recursively filtering out the C_D values while keeping the C_A values allow for the data to be filtered from baseline noise in the transform domain.

Dataset Preparation and Training

Once the baseline noise is removed through the application of the DWT, the next step is the dataset preparation for training. To forecast accurately with respect to in-sample and out of sample, it is necessary to prepare the data set in such a way that it captures patterns which can be interpolated to forecast out of sample [24]. This is done using the weighted mean of the recent samples to create a composite dataset given by:

$$\mathbf{X_{Comp} = X_{MR}(t), T_{MR}(t), T_{MR}^{moving}(t - iP) \forall\, i \epsilon 1 \ldots.. n} \quad \mathbf{(2)}$$

Here,
X_{Comp} denotes the composite training vector.
$X_{MR}(t)$denotes the multi-resolution filtered time series input variables.
$T_{MR}(t)$ denotes the multi-resolution filtered time series target variable.
T_{MR}^{moving} denotes the moving recent samples of the multi-resolution filtered time series target variable.
P denotes the sample period for the moving window.
i denotes the index of the window.

The index of the window can be chosen based on the point of prediction for the in-sample or out of sample forecast. It should be noted though that choosing a very wide window would sabotage the initial purpose of creating the moving window by incorporating a very wide sample space [25]-[26].. On the contrary, creating a very narrow window size P, would result in ineffective capturing of the data samples. Ideally, the following relation should hold true:

$$\boldsymbol{L_{sample}(P) \approx L_{samples}(F)} \quad \textbf{(3)}$$

Here,
$L_{sample}(P)$ denotes the length of the moving window in terms of samples.
$L_{sample}(F)$ denotes the length of the forecast span in terms of samples

The next step is the design and training of the deep neural network. The cost function deciding the truncation of the deep neural network chosen in this paper happens to be the mean squared error (mse), defined as:

$$\boldsymbol{J = \underbrace{min}_{over\ i} \frac{1}{n}\sum_{i=1}^{n}(p_i - \hat{t}_i)^2} \quad \textbf{(4)}$$

Here,
J denotes the cost function.
i denotes the iterations over which the value of J needs to be minimized, monotonically.
n denotes the number of samples for forecast.
p denotes the predicted value.
$\hat{t}$ denotes the target.

The back propagation training rule comprising of the error gradient as well as the 2nd order partial rate of change of forecasted errors need to be applied as feedback vectors at each iteration. Mathematically,

$$\boldsymbol{X_i^f = \begin{bmatrix} \frac{\partial^2 e}{\partial x1 \partial w1} & \cdots & \frac{\partial^2 e}{\partial x1 \partial wn} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 e}{\partial xn \partial w1} & \cdots & \frac{\partial^2 e}{\partial x1 \partial wn} \end{bmatrix}_i, \nabla[\frac{1}{n}\sum_{i=1}^{n}(p_i - \hat{t}_i)^2]} \quad (5)$$

Here,
X_i^f denotes the feedback input at each iteration.
$\begin{bmatrix} \frac{\partial^2 e}{\partial x1\, \partial w1} & \cdots & \frac{\partial^2 e}{\partial x1\, \partial wn} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 e}{\partial xn\, \partial w1} & \cdots & \frac{\partial^2 e}{\partial x1\, \partial wn} \end{bmatrix}$ denotes the 2nd order Hessian Matrix.
$\nabla[\frac{1}{n}\sum_{i=1}^{n}(p_i - \hat{t}_i)^2]$ denotes the rate of change of cost function for each iteration i.

It is important to note that the 2nd order normal derivative could have also been computed in terms of the Jacobian, given by:

$$\mathbf{J_k} = \begin{bmatrix} \frac{\partial^2 \mathbf{e_1}}{\partial \mathbf{w_1^2}} & \cdots & \frac{\partial^2 \mathbf{e_1}}{\partial \mathbf{w_m^2}} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 \mathbf{e_n}}{\partial \mathbf{w_1^2}} & \cdots & \frac{\partial^2 \mathbf{e_n}}{\partial \mathbf{w_m^2}} \end{bmatrix} \quad (6)$$

However, the ease of the computation of the 2nd order derivative would miss out specifically on the impact of the rate of change of the errors as a function of inputs, $\frac{\partial e}{\partial x}$ which serves an important purpose in this context as the moving window period P needs to interpolate the samples out of sample forecast. The additional computation burden would result in higher iterations to convergence but would also aid the in sample and more specifically the out of sample forecast [27].The proposed algorithm is presented next:

The results obtained though the implementation of the proposed algorithms is presented subsequently. The major performance metrics evaluated are the Mean Absolute Percentage Error (MAPE), R^2 and the forecasting accuracy).

Proposed Algorithm:

Start
Step.1: Split data into training and testing samples.
Step.2: Bifurcate the target variable $\hat{t}$ further into 2 categories for in-sample and out of sample forecasts.
Step.3: Define maximum iterations to convergence: $\boldsymbol{maxitr}$.
Step.4: Choose value of P: $\boldsymbol{L_{sample}(P) \approx L_{samples}(F)}$
Step.5: Define the value for validations checks $(\boldsymbol{val})$ for $\boldsymbol{k-fold}$ validation.
Step.6: Define the cost function J for least squares optimization,

$$\boldsymbol{J = \underbrace{min}_{over\ i} \frac{1}{n}\sum_{i=1}^{n}(p_i - \hat{t}_i)^2}$$

3. Experimental Results

Benchmark S & P datasets have been used for the in-sample and out of sample forecasts. The datasets tested are that of Amazon, Apple, Google, Infosys, Microsoft, Reliance, Tesla and Uber. The in sample and out of sample forecasts have been shown for each of the datasets. The time series data is first plotted followed in figure 1 for Google. This is followed by the multi-resolution analysis at level 3 using DWT. Subsequently the denoising of the time series data using DWT is depicted in figure 4. To validate the filtering capability of DWT, the normal and cumulative histograms have been presented in a juxtaposed manner so as to identify the characteristics of the original data, filtered data and residuals. A stark similarity among the original data sample and that of the filtered data indicates that the filtered samples indeed contain the maximal information content of the data. Whereas a palpable contrast in the statistical properties of the original data samples and that of the residual indicate the lack of correlation among the data samples indicating the fact that the residuals contain the baseline noise. For the sake of analysis, the Tesla stocks over a period of 10 years has been considered in the first case, as the stocks of Tesla exhibit substantial randomness and noise baseline over the specified tenure.

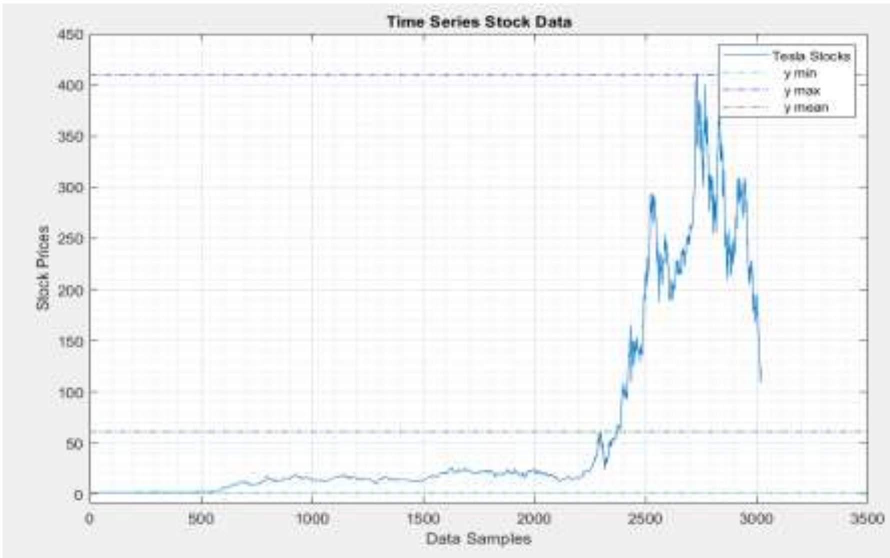

Fig.1Time series dataset for Google

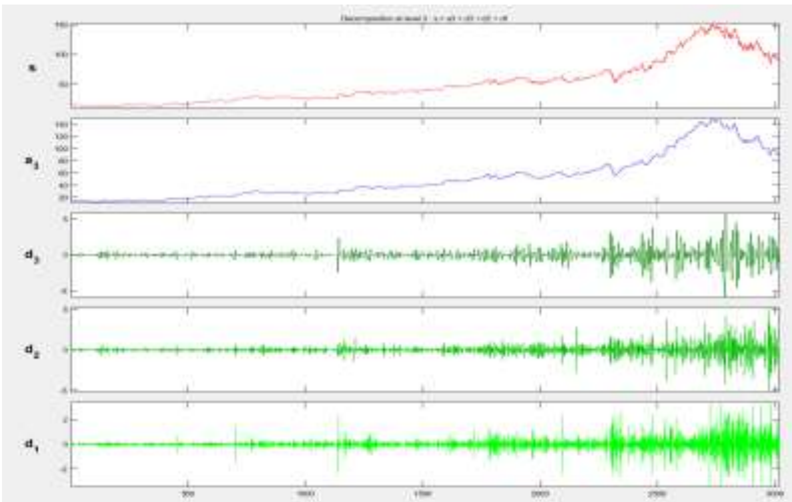

Fig.2 Multi-Resolution Analysis at level 3

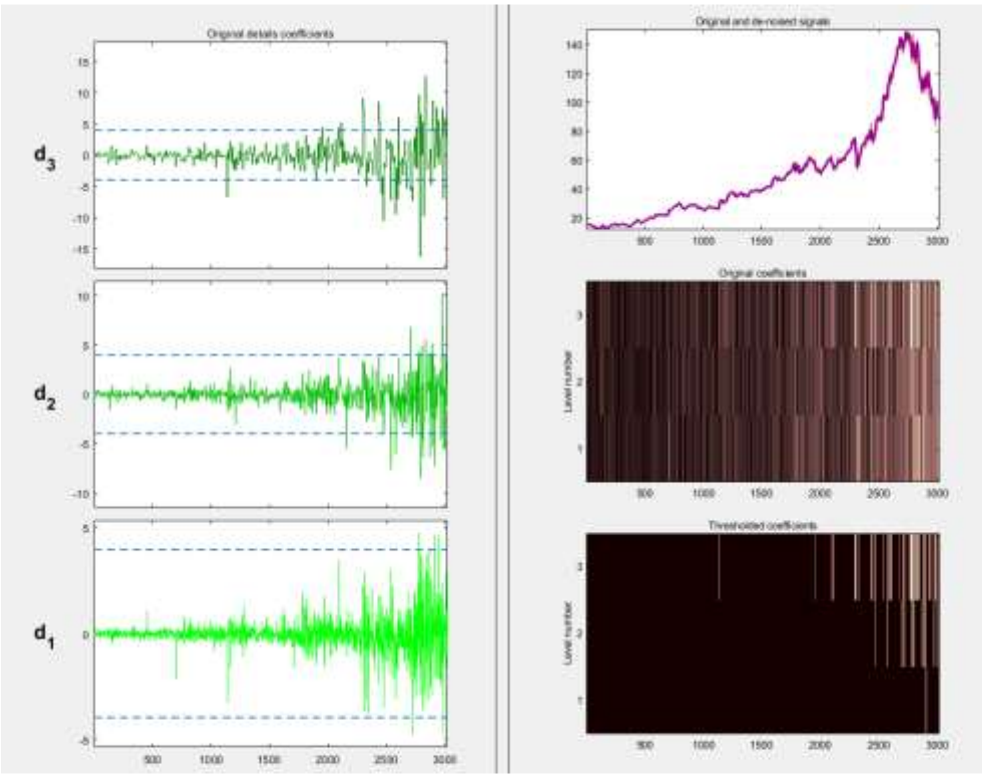

Fig.3 Filtered Data using DWT at level 3

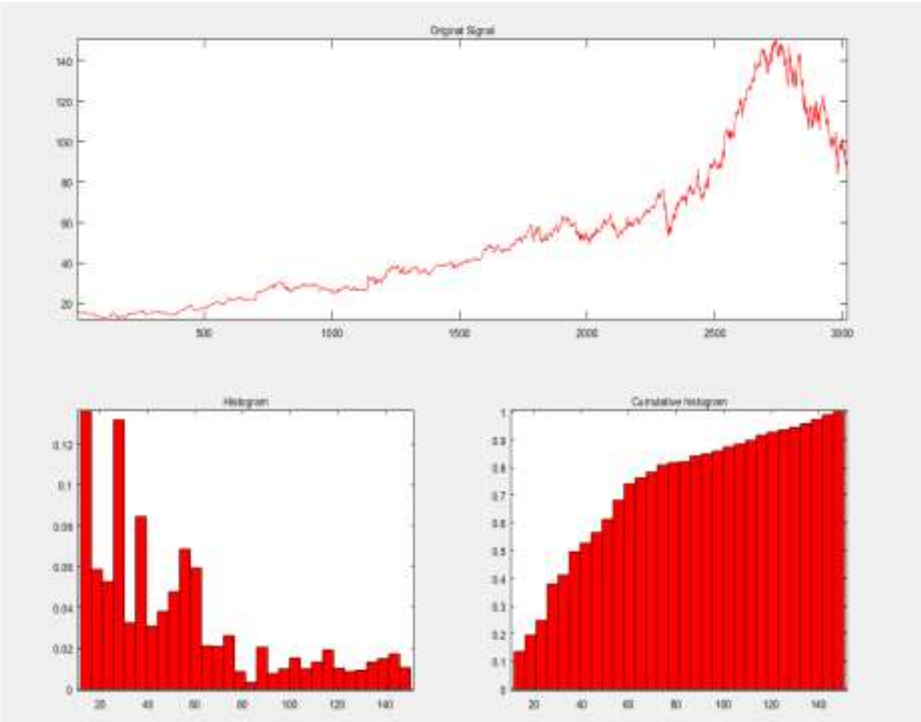

Fig.4Stochastic Histogram of original samples

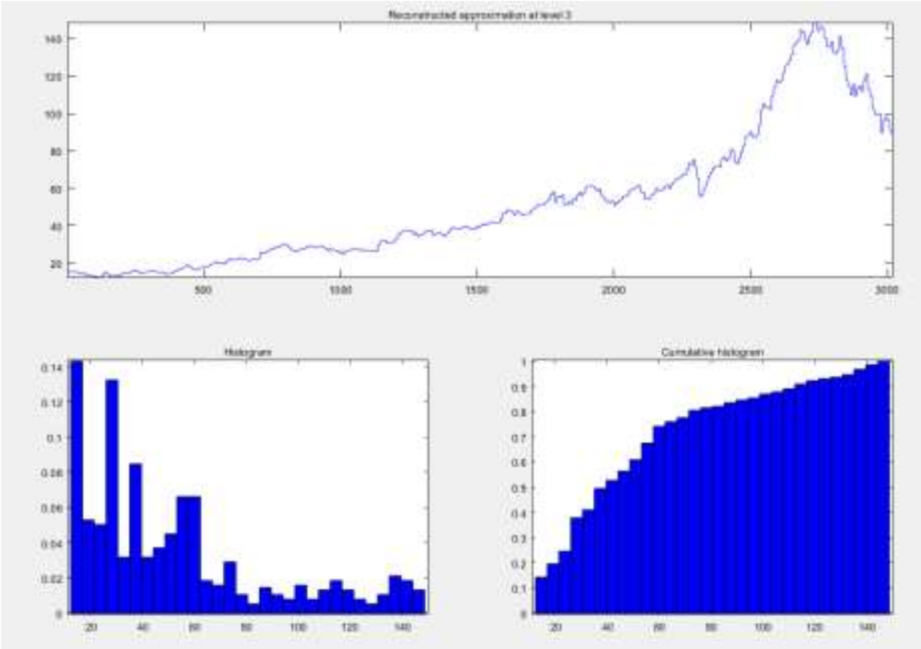

Fig.5 Stochastic Histogram of filtered samples

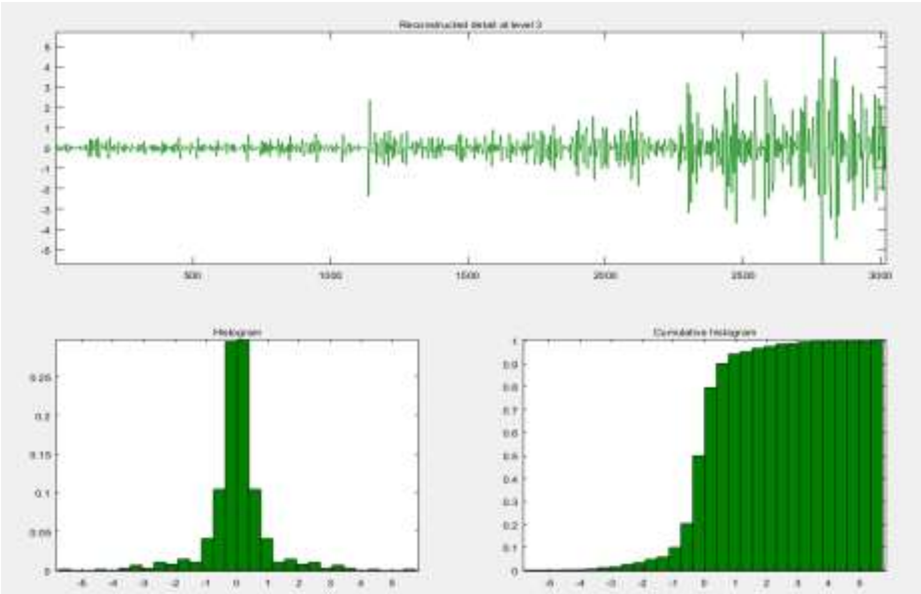

Fig.6 Stochastic Histogram of residuals

Table 1. Statistical Analysis of Data

Statistical Source	Statistics	Statistical Evaluations	DWT-Components
Normal and Cumulative Histogram	Min	11.83	Original Samples
	Max	150.7	
	Mean	51.96	
Normal and Cumulative Histogram	Min	12.27	Filtered Samples
	Max	148.6	
	Mean	51.96	
Normal and Cumulative Histogram	Min	-5.735	Residuals
	Max	5.735	
	Mean	0	

The analysis of the normal and cumulative histogram of the original samples, filtered samples and residuals in terms of the minimum, maximum and mean values, are depicted in figures 4, 5 and 6 respectively. The values are also tabulated in table 1 for ready reference. It can also be clearly observed that the normal and cumulative histograms' statistical values for the original and filtered samples are identical, indicating similarity in information content. However, the large divergence of the statistical values of the original data and residuals clearly indicates that the residuals are statistically non-identical to the original samples and hence contain the noise effects. This validates the proposed multi-resolution approach for noise filtering. A similar approach has been employed for all the datasets. The next approach would be analysing the in and out of sample prediction performance for the proposed approach. The results for each of the approaches have been presented next.

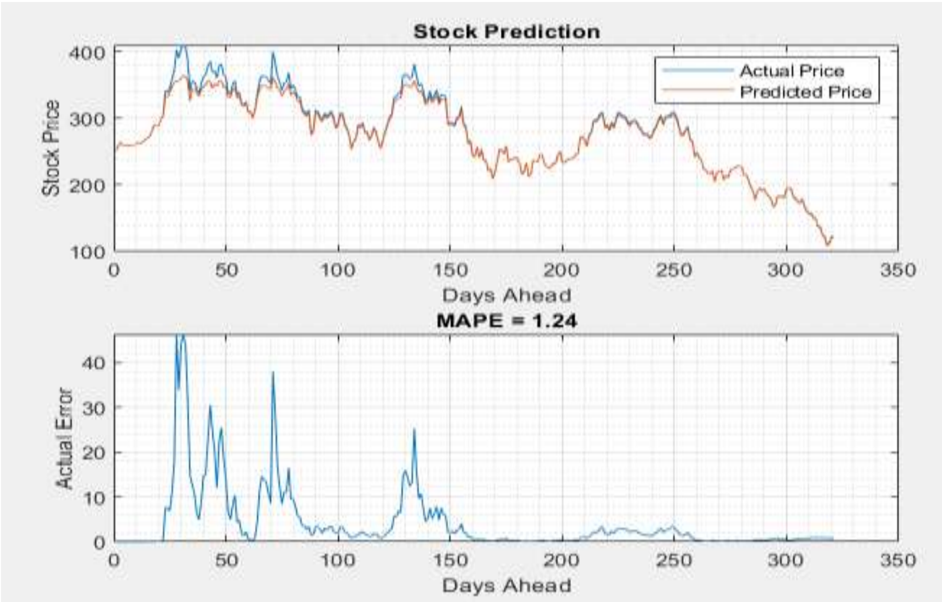

Fig.7Prediction: In Sample: Long Term

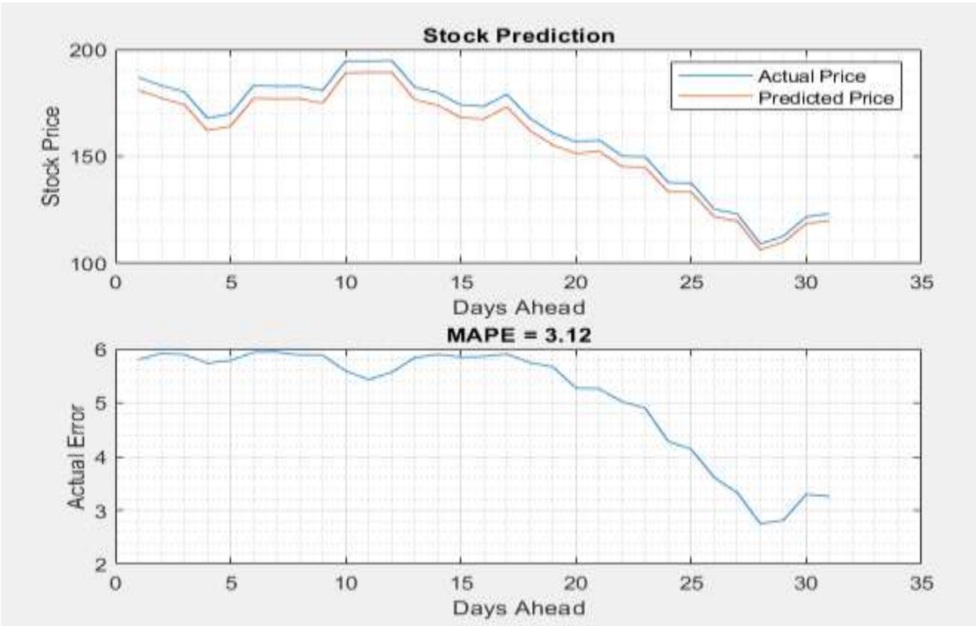

Fig.8 Prediction: Out of Sample: Mid Term

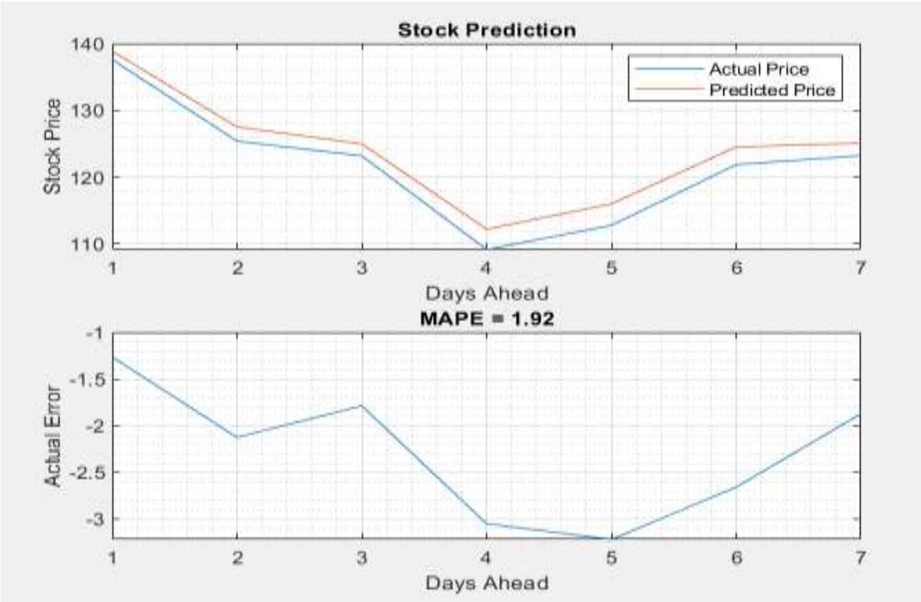

Fig.9 Prediction: Out of Sample: Short Term

The experimental results obtained can be categorized under the following heads:

1) In sample, long term: For long term investments in stocks of a particular company, typically stake holders opt for long term forecasting results ranging from few to several months. As the number of samples in such a forecast would be relatively high, the data split has been chosen as **in-sample,** i.e. the testing samples are juxtaposed with the training samples, lying in continuity. The total testing samples have been taken as 310 days. The MAPE obtained is 1.24%. The overall regression (R^2) achieved is 0.9987.

2) Out of sample, midterm: The next forecasting model is that for the midterm forecasting which may extend from a few days to several days. As the testing sample size can be much smaller than the training sample size, both sample sets can be separated so as to check the out of sample forecasting performance. The number of samples in this case has been chosen as 30 days (or 1 month). The MAPE obtained is 3.12%. The overall regression (R^2) achieved is 0.985

3) Out of sample, short term: The out of sample short term is an extension of the midterm forecasting model with the sample size being 7 days. The separation between the training and testing samples has been deliberately kept large so as to check the model's robustness. The MAPE obtained is 1.92%. The overall regression (R^2) achieved is 0.992

The summary of the MAPE values for all the stocks has been tabulated in table 2. The long, mid and short term forecasting have been analysed. Table 3 compares the error rates of the proposed approach against contemporary baseline techniques.

Table 2. Summary of MAPE for various stocks

S. No.	**Dataset**	**MAPE (In sample/ Long Term) (320 days)**	**MAPE (Out of sample/ Mid Term) (30 days)**	**MAPE (Out of sample/ Short Term) (7 days)**
1.	Amazon	1.11	1.86	0.98
2.	Apple	0.96	2.84	1.36
3.	Google	1.08	3.12	1.76
5.	Infosys	1.78	2.45	1.24
6.	Microsoft	0.91	1.44	1.05
7.	Reliance	1.41	2.71	1.38
8.	Tesla	1.24	3.12	1.92

Table 3. Comparison against Baseline Techniques

S. No.	Technique	MAPE
1.	Kim et al., 2020.	43%
2.	Boutkif et al., 2020	40%
3.	Sen et al., 2021.	5.6
4.	Sharma et al.	2.255 (short term)& 8.15 (mid term)
5.	Sharaf et al., 2022.	7.7%
6.	**Proposed Technique (Mean Accuracy)**	1.76 (short term) & 3.12 (mid-term)

In this approach, multi-resolution filtering has been employed with a moving mean as the data pre-processing method prior to applying the processed data to a deep neural network employing back propagation. The MAPE has been computed for 3 cases wherein the long term forecast (in sample) happens to predict 320 days in advance, the short and midterm models (both out of sample) happen to forecast 30 and 7 days respectively. The MAPE of the proposed system can be seen to clearly outperform the baseline techniques in terms of MAPE, thereby indicating the superiority of the proposed approach.

4. Conclusion

The paper investigates the performance of a trained deep neural network model with regards to both in-sample and out of sample forecast for a diverse set of stock datasets. The approach employs the 3rd level multi-resolution analysis of the dataset retaining the low resolution components and removing the high resolution components of the data in the transform domain. The statistical analysis of the histograms clearly indicate that the DWT acts as an effective noise filter for the baseline noise contained in the datasets. A 7, 30 and 320 day forecast model is designed for short, mid and long terms forecasts. The results are evaluated in terms of the MAPE and R^2 values, comparing whom with existing baseline techniques clearly indicate that the proposed approach beats the existing approaches in terms of MAPE of forecasting. Thus the model is robust and suited to Variational forecast samples. Future directions of the work can seek to extract intelligible opinion minded data from public social media forums to create an exogenous variable to bolster the forecasting accuracy.

5. References

1. HJ Park, Y Kim, HY Kim, “Stock market forecasting using a multi-task approach integrating long short-term memory and the random forest framework”, Elsevier 2022, vol.114., art.108106.
2. M. Kesavan, J. Karthiraman, R. T. Ebenezer and S. Adhithyan, "Stock Market Prediction with Historical Time Series Data and Sentimental Analysis of Social Media Data," 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2020, pp. 477-482.
3. X. Yuan, J. Yuan, T. Jiang and Q. U. Ain, "Integrated Long-Term Stock Selection Models Based on Feature Selection and Machine Learning Algorithms for China Stock Market," in IEEE Access, 2020, vol. 8, pp. 22672-22685.
4. P Rajendiran, PLK Priyadarsini, “Survival study on stock market prediction techniques using sentimental analysis”, Materials Today Proceedings, Elsevier 2023, vol.80., no.3, pp.3229-3234.
5. L. Conegundes and A. C. M. Pereira, "Beating the Stock Market with a Deep Reinforcement Learning Day Trading System," 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 2020, pp. 1-8.
6. S. Ravikumar and P. Saraf, "Prediction of Stock Prices using Machine Learning (Regression, Classification) Algorithms," 2020 International Conference for Emerging Technology (INCET), Belgaum, India, 2020, pp. 1-5.
7. S. Chen and C. Zhou, "Stock Prediction Based on Genetic Algorithm Feature Selection and Long Short-Term Memory Neural Network," in IEEE Access, vol. 9, pp. 9066-9072.
8. SI Lee, SJ Yoo, “Multimodal deep learning for finance: integrating and forecasting international stock markets”, Journal of Supercomputing, Springer 2020, vol.76, pp.76, 8294–8312.
9. S Kim, S Ku, W Chang, JW Song, “Predicting the Direction of US Stock Prices Using Effective Transfer Entropy and Machine Learning Techniques”, IEEE Access 2020, Vol-8, pp: 111660 – 111682.

10. S Bouktif, A Fiaz, M Awad, Amir Mosavi, “Augmented Textual Features-Based Stock Market Prediction”, IEEE Access 2020, Volume-8, PP: 40269 – 40282.
11. J Eapen; D Bein; A Verma, “Novel Deep Learning Model with CNN and Bi-Directional LSTM for Improved Stock Market Index Prediction”, 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), IEEE 2019 pp. 0264-0270.
12. M Wen; P Li; L Zhang; Y Chen, “Stock Market Trend Prediction Using High-Order Information of Time Series”, IEEE Access 2019, Volume 7, pp : 28299 – 28308.
13. Y Guo, S Han, C Shen, Y Li, X Yin, Y Bai, “An adaptive SVR for high-frequency stock price forecasting”, Volume-6, IEEE Access 2018, pp: 11397 – 11404.
14. MS Raimundo, J Okamoto, “SVR-wavelet adaptive model for forecasting financial time series”, 2018 International Conference on Information and Computer Technologies (ICICT), IEEE 2018, pp. 111-114.
15. Y Baek, HY Kim, “ModAugNet: A new forecasting framework for stock market index value with an overfitting prevention LSTM module and a prediction LSTM module” Journal of Expert System and Applications, Elsevier 2018, Volule-113, pp: 457-480.
16. S Selvin, R Vinayakumar, E. A Gopalakrishnan ; Vijay Krishna Menon; K. P. Soman, “Stock price prediction using LSTM, RNN and CNN-sliding window model”, 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE 2017, pp. 1643-1647.
17. M Billah, S Waheed, A Hanifa, “Stock market prediction using an improved training algorithm of neural network”, 2016 2nd International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE), IEEE 2016, pp. 1-4.
18. J. Sen, S. Mehtab and A. Dutta, "Volatility Modeling of Stocks from Selected Sectors of the Indian Economy Using GARCH," 2021 Asian Conference on Innovation in Technology (ASIANCON), IEEE, 2021, pp. 1-9.
19. DK Sharma, HS Hota, K Brown, R Handa, “Integration of genetic algorithm with artificial neural network for stock market forecasting”, International Journal of System Assurance Engineering and Management, Springer, 2022, vol.13, pp.828–841.
20. M Sharaf, EED Hemdan, A El-Sayed, NA El-Bahnasawy, “An efficient hybrid stock trend prediction system during COVID-19 pandemic based on stacked-LSTM and news sentiment analysis”, Multimedia Tools and Applications, Springer 2023, vol. 82, pp.23945–23977.
21. K. Virupakshappa, M. Marino and E. Oruklu, "A Multi-Resolution Convolutional Neural Network Architecture for Ultrasonic Flaw Detection," 2018 IEEE International Ultrasonics Symposium (IUS), Kobe, Japan, 2018, pp. 1-4.
22. Aldhahab, T. Alobaidi, A. Q. Althahab and W. B. Mikhael, "Applying Multiresolution Analysis to Vector Quantization Features for Face Recognition," 2019 IEEE 62nd International Midwest Symposium on Circuits and Systems (MWSCAS), Dallas, TX, USA, 2019, pp. 598-601.
23. Z. Zeng and M. Khushi, "Wavelet Denoising and Attention-based RNN- ARIMA Model to Predict Forex Price," 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 2020, pp. 1-7.
24. R. Chiong, Z. Fan, Z. Hu and S. Dhakal, "A Novel Ensemble Learning Approach for Stock Market Prediction Based on Sentiment Analysis and the Sliding Window Method," in IEEE Transactions on Computational Social Systems, 2022.
25. H. G. T. Thu, T. N. Thanh and T. L. Quy, "A Neighborhood Deep Neural Network Model using Sliding Window for Stock Price Prediction," 2021 IEEE International Conference on Big Data and Smart Computing (BigComp), Jeju Island, Korea (South), 2021, pp. 69-74.
26. H Dong, X Guo, H Reichgelt, R Hu, “Predictive power of ARIMA models in forecasting equity returns: a sliding window method”, Journal of Asset Management, Springer, 2020, vol.21., pp.549–566.
27. DW Jia, ZY Wu, “A Laplace asymptotic integral-based reliability analysis method combined with artificial neural network”, Applied Mathematical Modelling, Elsevier 2022, vo.105, pp.406-422.