



Fusion of Wavelet Decomposition and N-BEATS for Improved Stock Market Forecasting

Neha Pramanick¹ · Vatsal Singhal¹ · Neeraj¹ · Jimson Mathew¹ · Mayank Agarwal¹ 

Received: 29 July 2023 / Accepted: 11 August 2024
© The Author(s), under exclusive licence to Springer Nature Singapore Pte Ltd. 2024

Abstract

Stock market forecasting is one of the most exciting areas of time series forecasting both for the industry and academia. Stock market is a complex, non-linear and non-stationary system with many governing factors and noise. Some of these factors can be quantified and modeled, whereas some factors possess a random walk behavior making the process of forecasting challenging. Various statistical methods, machine learning, and deep learning techniques are prevalent in stock market forecasting. Recently, there has been a paradigm shift towards hybrid models, showing some promising results. In this paper, we propose a technique that combines a recently proposed deep learning architecture N-BEATS with wavelet transformation for improved forecasting of future prices of stock market indices. This work uses daily time series data from five stock market indices, namely NIFTY 50, Dow Jones Industrial Average (DJIA), Nikkei 225, BSE SENSEX, and Hang Seng Index (HSI), for the experimental studies to compare the proposed technique with some traditional deep learning techniques. The empirical findings suggest that the proposed architecture has high accuracy as compared to some traditional time series forecasting methods and can improve the forecasting of non-linear and non-stationary stock market time series.

Keywords Time series · Forecasting · Stock market · Deep learning · N-BEATS

Mathematics Subject Classification 00-01 · 99-00

Introduction

Time series forecasting is currently a prominent trend in both industry and academia, offering solutions to various business challenges such as predicting future sales, electricity consumption, stock prices, and revenues. This forecasting aids businesses in making informed decisions to maximize profits. Particularly, stock market forecasting constitutes a critical subset of time series forecasting, involving the prediction of future stock values or indices. The stock market, characterized as a complex, non-linear, and non-stationary system, presents challenges due to governing factors and inherent noise. Some factors are quantifiable and modelable, while others exhibit random walk behavior, adding complexity to the forecasting process.

This article describes a new hybrid method that combines wavelet transformation (WT) and neural basis expansion analysis for interpretable time series (N-BEATS) to make stock market forecasting better. The proposed method holds the potential to benefit hedge funds, retail investors, investment banks, and various financial institutions, facilitating improved decision-making and trading strategies. The combination of wavelet transformation and N-BEATS [1] breaks down stock market data into rough estimates and precise coefficients. Each component is individually forecasted by separate N-BEATS models, and the stock price prediction is obtained by additively combining the final forecasts.

The real-world impact of this research reaches financial analysts, investors, and policymakers, furnishing them with a powerful instrument for making well-informed decisions. Enhanced precision in forecasting future stock prices contributes to effective risk management, portfolio optimization, and strategic investment planning. The study's outcomes carry substantial potential advantages for practical applications in various scenarios.

✉ Mayank Agarwal
mayank265@iitp.ac.in

¹ Department of Computer Science and Engineering, Indian Institute of Technology, Patna, Patna 801103, India

Utilizing daily time series data from five different stock market indices (DJIA, NIFTY 50, Nikkei 225, BSE SENSEX, and HSI), this study aims to implement and compare the proposed architecture with standalone N-BEATS, Long Short-Term Memory (LSTM), and Convolutional Neural Networks (CNN). Empirical studies indicate that the WT + N-BEATS architecture outperforms some traditional deep learning methods, demonstrating its effectiveness. The research hypothesizes that this hybrid approach will significantly enhance the accuracy of stock market forecasting, outperforming both standalone N-BEATS and traditional deep learning techniques. The anticipated outcomes include more accurate and interpretable predictions, contributing to advancements in automated trading systems and strategies, and providing valuable insights for stakeholders in the financial sector.

Scientific Contribution and Importance of the Proposed Stock Market Prediction Technique

This study's proposed stock market prediction technique significantly contributes to the field of financial forecasting. The stock market is a complicated, non-linear, and non-stationary system that is affected by many factors and has its own noise. This method solves the difficult problems it faces by combining the N-BEATS deep learning architecture with wavelet transformation. The major contributions of our proposed work are summarized as follows:

- The incorporation of both N-BEATS and wavelet transformations represents a noteworthy advancement in hybrid modeling for stock market prediction. This combination leverages the strengths of deep learning in capturing intricate patterns and the adaptability of wavelet transformation in handling non-stationary data.
- The empirical findings of our experimental studies demonstrate that the proposed technique consistently outperforms traditional deep learning methods. Its high accuracy in forecasting stock market indices, including NIFTY 50, DJIA, Nikkei 225, BSE SENSEX, and HSI, highlights its efficacy in improving predictive performance.
- The significance of this technique is particularly pronounced in its ability to enhance forecasting for non-linear and non-stationary stock market time series. By effectively capturing the random walk behavior inherent in certain factors, the proposed model contributes to a more nuanced understanding of market dynamics.
- The practical implications of this research extend to financial analysts, investors, and policymakers, providing them with a more robust tool for informed decision-making. The improved accuracy in predicting future stock

prices can aid in risk management, portfolio optimization, and strategic investment planning.

- Academically, this study contributes to the evolving landscape of time series forecasting by introducing a novel hybrid model that integrates deep learning and wavelet transformation. The findings provide valuable insights for researchers exploring innovative approaches to address the complexities of financial markets.

The paper will begin with a literature review in “[Literature Review](#)” section that will discuss some of the existing stock market forecasting methods. In “[Proposed Approach for Stock Market Prediction](#)” section, we brief the proposed approach for stock market prediction, and in “[Experimental Methodology](#)” section, we discuss the experimental methodology. Experimental results obtained in this work will be discussed in “[Experimental Results](#)” section. The conclusions are discussed in “[Conclusion](#)” section.

Literature Review

Stock market forecasting has always been an active area of research. Various theories, models, and methodologies have been proposed in literature for stock market forecasting. In this section we discuss few of the major theories proposed in literature. Fama [2], talks about the random walk nature of the stock market movements and discusses the prevalent theory of technical analysis and fundamental analysis. Several other statistical time series models have been employed for stock market forecasting. Many linear time series models like Autoregressive Integrated Moving Average (ARIMA), Autoregressive Moving Average (ARMA), and Exponential Smoothing have been used for stock market prediction as in [3, 4]. However, it has been argued that stock market data contains a lot of noise and non-linearity. Linear models like ARIMA do not perform well on noisy, non-stationary, and non-linear data. Luo et al. [5] have tried to fit the ARIMA model on stock market index price data after using wavelet denoising to obtain better forecasting results.

Various machine learning (ML) and deep learning techniques have been used for the task of stock market forecasting. Machine learning models like Decision Tree, Random Forest, and Support Vector Machines (SVM) are widely used in literature for forecasting tasks. Milosevic et al. [6] compared and contrast various ML algorithms like Random Forest, Support Vector Machine, Naive Bayes, and Logistic Regression to classify stocks as good and bad using stock price and some financial ratios. Zhang et al. [7] used a random forest algorithm for stock trend prediction. Patel et al. [8] used a hybrid of Support Vector Regressor, Random Forest, and Artificial Neural Network (ANN) for forecasting future values of the stock market index. They used 10

technical indicators as input features for a two stage fusion model.

With an increase in the availability of data and computational resources, researchers started using deep learning techniques like Artificial Neural Network (ANN), Convolutional Neural Network (CNN), Recurrent Neural Networks (RNN), and Long Short-Term Memory (LSTM) for stock market forecasting [9–12]. Wang et al. [13] used ANN for stock market forecasting using daily stock market return and volume as a major features for training the neural network. Bernal et al. [14] implemented Echo State Networks for forecasting future prices of S &P 500. Echo State Network is a subclass of RNN. LSTM, modified form of RNN is used to forecast future stock market prices by Moghar et al. [15]. They showed that LSTM models were capable of forecasting the opening price with good accuracy for assets under consideration. Recently after the success of attention mechanisms in Natural Language Processing (NLP) tasks, researchers have started studying LSTM with attention mechanisms to improve the forecasting power of LSTM networks. Qui et al. [16] used LSTM with attention mechanism on datasets of various stock market indices. They used attention based LSTM along with wavelet denoising to forecast the stock opening price and compared the results obtained with some widely used methods like simple LSTM. Eapen et al. [17] combined multiple pipelines of CNN and bi-directional LSTM units for forecasting variations in the stock price index showing a significant improvement on S &P 500 grand challenge dataset by using a combination of CNN and LSTM.

Many time series like daily stock prices display a time-varying non-stationary structure. Fryzlewicz et al. [18] tried to forecast these non-stationary time series using non-decimated wavelets. They used locally stationary wavelet processes for deriving a new forecasting mechanism. Renaud et al. [19] illustrated the use of wavelet transformations for modeling and forecasting non-stationary time series. They discussed various applications of wavelet transformation in time series forecasting and illustrate how wavelet transformation can be used for time series forecast.

Oreshkin et al. [1] used a novel architecture called N-BEATS for univariate time series forecasting. The authors of [1] proposed a deep neural network based on backward and forward residual links and a very deep stack of fully-connected layers. N-BEATS has given promising results on several well-known datasets, including M3, M4, and TOURISM competition datasets containing time series data. In this work, we used N-BEATS to make stock market forecasts and combine N-BEATS with wavelet transformation to improve stock market forecasting (Table 1).

In recent times, numerous researchers have employed cutting-edge deep learning frameworks. Wang et al. [20] utilized the Transformer framework, illustrating its substantial

outperformance compared to traditional methods and its ability to generate excess earnings for investors. In their work Ghotbi et al. [21] introduced a novel approach that combines the kinetic energy formula with indicator signals for price prediction. Additionally, they explored predictions using deep reinforcement learning (DRL), revealing promising outcomes in terms of both accuracy and profitability. The author in [22] utilized the Random Forest classifier to uncover concealed relationships between stock indices and the specific trend of a stock, yielding promising outcomes. Amin et al. [23] utilized a range of machine learning classifier models, such as gradient boosting, decision trees, and random forests. They trained the algorithms using tweet sentiments and relevant variables to predict fluctuations in stock prices of prominent firms, such as Microsoft and OpenAI. Behera et al. [24] propose a comprehensive approach that utilizes machine learning algorithms to forecast stock returns and a mean-VaR model to choose portfolios. Yanli et al. [25] introduced a novel hybrid model, SA-DLSTM, specifically developed for forecasting stock market trends and simulating trading activities. This model integrates an emotion-enhanced convolutional neural network (ECNN), denoising autoencoder (DAE) models, and a long short-term memory model (LSTM). Agrawal et al. [26] proposed the usage of correlation-tensor for stock market prediction. Mehta et al. [27] postulated sentiment analysis by employing a blend of machine learning and deep learning methodologies, such as support vector machine, MNB classifier, linear regression, Naïve Bayes, and long short-term memory. In their methodology, Vateekul et al. [28] implemented the walk-forward validation strategy. Yilin et al. [29] employed machine learning and deep learning prediction models to preselect companies prior to creating portfolios. Afterwards, they incorporated the prediction results to improve the mean-variance (MV) and omega portfolio optimization models. Table 2 provides a comparison of recent studies for stock market prediction.

Proposed Approach for Stock Market Prediction

In this study, we propose a novel approach for stock market prediction by leveraging the combined power of the wavelet transform and the N-BEATS (Neural Basis Expansion Analysis for Time Series) architecture. The stock market is a complex and dynamic system characterized by nonlinearities and non-stationarities. To address these challenges, we employ the wavelet transform, a mathematical tool capable of decomposing time series data into different frequency components, thereby revealing hidden patterns and structures. Subsequently, we integrate the N-BEATS architecture, a state-of-the-art deep learning

Table 1 Comparative Summary of Studies in Stock Market Forecasting

| Study reference | Methodology used | Results | Limitations |
|----------------------|-----------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|
| Amin et al. [23] | Gradient boosting, decision trees, random forests, ML classifier models | Using ML classifiers on tweet sentiments and relevant features to predict stock price movements for companies such as Microsoft and OpenAI | Dependence on external data (tweets); generalization may be limited |
| Behera et al. [24] | ML algorithms for stock return prediction, mean-VaR model for portfolio selection | Comprehensive approach using ML algorithms for stock return prediction and a mean-VaR model for portfolio selection | May be sensitive to the choice of ML algorithms and parameters; assumptions in mean-VaR model |
| Yanli et al. [25] | Hybrid model: SA-DLSTM (ECNN, DAE, LSTM) | Introducing SA-DLSTM, a hybrid model that combines ECNN, DAE, and LSTM to forecast stock market trends and simulate trading activities | Interpretability challenges; potential sensitivity to model components and hyperparameters |
| Agrawal et al. [26] | Correlation-tensor | Use of a correlation-tensor for stock market prediction | Limited explanation of how correlations are derived; potential dependence on data quality |
| Mehta et al. [27] | Sentiment analysis: SVM, MNB classifier, linear regression, Naïve Bayes, LSTM | Implementation of sentiment analysis using a combination of ML and DL methodologies for stock market forecasting | Dependence on sentiment analysis accuracy; interpretability challenges |
| Vateekul et al. [28] | Walk-forward validation strategy | Utilization of the walk-forward validation strategy in their methodology for stock market forecasting | May be computationally intensive; sensitivity to validation strategy parameters |
| Yilin et al. [29] | Machine learning and deep learning prediction models | Applying ML and DL prediction models to preselect companies before portfolio creation, then incorporating the results to enhance MV and omega portfolio optimization | Dependence on prediction model accuracy; generalization to other portfolios may vary |

model designed for time series forecasting that has shown promising results in various domains. Our proposed methodology involves the decomposition of stock market data using wavelet transforms into approximations and detailed coefficients. We then forecast each of these components individually using separate N-BEATS models. Combining these forecasts yields the final prediction.

Association Between Wavelet Transformation and Prediction Using Deep Learning Model

In our approach, the amalgamation of wavelet transformation and the N-BEATS architecture plays a pivotal role in refining stock market forecasting. Wavelet transformation is an important step before any analysis can begin. It finds hidden patterns in time series data by breaking it up into separate frequency components. The enriched data is then input into the N-BEATS deep learning model, renowned for its ability to capture intricate temporal dependencies. The cooperative utilization of wavelet transformation and N-BEATS establishes a synergistic relationship, enhancing the data with hidden structures and enabling N-BEATS to leverage this information for precise and dynamic predictions. This synergy effectively addresses the challenges presented by the non-linear and non-stationary nature of stock market data, resulting in an enhancement in forecasting performance.

We chose the wavelet transform for time series forecasting because of its distinctive ability for multi-resolution analysis. Unlike conventional linear models or machine learning techniques, wavelet transform stands out in breaking down time series data into different frequency components, offering a localized perspective that captures both high- and low-frequency details. This flexibility is essential for handling non-stationarities and uncovering concealed structures in financial time series, ultimately improving accuracy in predicting dynamic and unpredictable market conditions. The unique strengths of wavelet transform make it a valuable selection, adding to a more efficient and all-encompassing approach to predictive modeling.

Experimental Methodology

The suggested methodology utilizes WT and N-BEATS to forecast the closing values of stock market indices with a 1-day lead time. We employ WT to decompose the univariate signal, enhancing forecasting accuracy by considering the random and noisy characteristics of stock market data. The N-BEATS architecture is advantageous because of its simplicity and generic nature, allowing for efficient training

without the requirement of specialized time series feature engineering. The experimental methodology involves several stages, including the selection of stock market datasets for evaluation, data decomposition for analysis and forecasting, utilization of a forecasting model for forecasting future closing prices, and subsequent evaluation of performance. Figure 3 depicts the schematic depiction of the proposed methodology.

Dataset Description

To thoroughly test and validate the proposed technique, we utilized daily time series data sourced from five distinct global stock exchanges [30]. The analysis and experimentation involved major indices from each of these exchanges, encompassing datasets with diverse lengths and characteristics in terms of noise and volatility. The selected stock market indices represent a range of economies, including mature ones such as the USA and Japan, as well as a developing economy like India, known for its high volatility and noise [31].

Due to the inherent noise present in stock market data [32], accurate forecasting results can be challenging. To address this issue, we employed wavelet denoising on all the datasets considered in this study. The data underwent transformation using the moving window technique to render it suitable for training and forecasting purposes. In this study, the models were trained to predict the stock index price one day ahead using the stock index prices from the previous N days. The researchers set N to 19, which corresponds to approximately 20 business days in a month. The initial aim was to forecast the 1-day-ahead closing price of a stock index using data from the past month. Through grid search, we optimized the window size N and selected 19 as the optimal value. Subsequently, all datasets were transformed using the moving window technique with a window size of 19 to forecast the 1-day-ahead closing price of the stock market index, as illustrated in Fig. 1.

A concise overview of each dataset is outlined below:

- The Nifty 50 dataset represents a collection of historical stock market data related to the Nifty 50 index. The Nifty 50 is an index comprising 50 actively traded Indian stocks, representing various sectors of the Indian economy. This dataset likely includes information such as daily opening and closing prices, high and low prices, trading volumes, and other relevant financial indicators for the individual stocks within the Nifty 50 index.
- The DJIA dataset encompasses historical stock market data related to the Dow Jones Industrial Average, a prominent stock market index in the United States. It com-

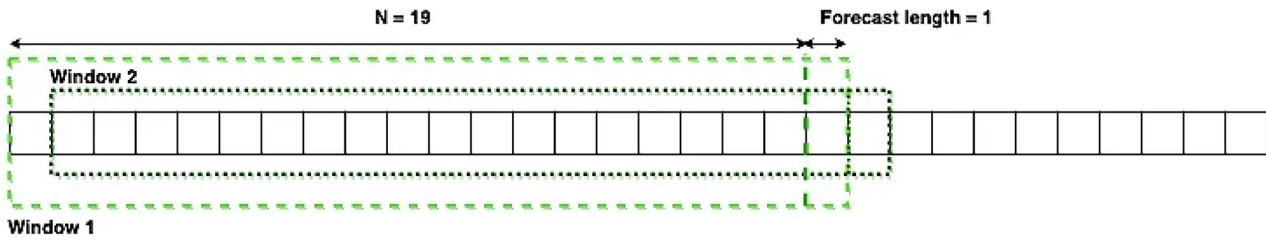


Fig. 1 Moving window

prises 30 large and well-established companies, reflecting the performance of key sectors in the American economy.

- The Nikkei 225 dataset contains historical data associated with the Nikkei 225 index, which represents the stock market performance of 225 major companies listed on the Tokyo Stock Exchange. This index is a key indicator of the Japanese equity market.
- The BSE SENSEX dataset provides historical stock market information for the SENSEX index on the Bombay Stock Exchange (BSE) in India. SENSEX is a benchmark index consisting of 30 well-established and financially sound companies listed on the BSE.
- The HSI dataset covers historical stock market data linked to the Hang Seng Index in Hong Kong. The Hang Seng Index reflects the performance of 50 major companies listed on the Hong Kong Stock Exchange, providing insights into the financial health of the Hong Kong market.

Table 2 provides a summary of all the datasets used in this study.

Data Pre-processing

Implementing a data pre-processing technique facilitates experimentation, ensuring smooth processing and effective learning. In the proposed approach, data normalization is a critical step in pre-processing for time series forecasting. This step significantly improves algorithm performance and convergence, fosters consistent model behavior, and enhances generalization across diverse datasets. We accomplish data scaling using the min-max normalization technique, as represented in Eq. 1.

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{1}$$

Data can be transformed using the min-max normalization approach into a specified range, usually [0, 1]. To ensure that every data point has the same scale and that every feature is equally significant, min-max normalization is used. For each characteristic, the lowest value is converted to 0,

the highest value to 1, and all other values are converted to a decimal between zero and one. By applying a linear modification to the original data, min-max normalization maintains the relationships between the values in the original data. Rescaling features is a typical data science technique that keeps large-scale characteristics from overshadowing small-scale features during machine learning model training. To summarise, min-max normalisation is a commonly employed method for rescaling data to a particular range, guaranteeing that every characteristic has an identical scale and is of equal significance for analysis and modelling.

Data Decomposition

The architecture designed to address the challenges posed by the noisy and non-stationary nature of stock prices involves the application of discrete wavelet transforms (DWT). Renaud et al. [19] and Fryzlewicz et al. [18] have demonstrated the effectiveness of wavelet transformation in decomposing univariate signals for subsequent forecasting, inspiring this approach. The mathematical model of wavelet transformation, outlined in [33], serves as the foundation for this architecture. The data decomposition is performed in following stages:

- Selection of Wavelet Transform Type:
 - The architecture focuses on utilizing DWT due to their applicability in decomposing time series data.
- Decomposition into Coefficients:
 - The DWT decomposes the time series data into approximation coefficients and detailed coefficients. These coefficients play a crucial role in capturing both the high- and low-frequency components of the signal.
 - According to Bunnoon et al. [34], a given signal $x(t)$ can be represented as Eq. 2.

$$x(t) = \sum_i c_{j_0,i} \phi_{j_0,i}(t) + \sum_{j>j_0} \sum_i \omega_{j,i} 2^{\frac{j}{2}} \psi(2^j t - i) \tag{2}$$

Here, $x(t)$ is an input signal, i refers to the scaling index, j refers to the level index, $\phi_{j_0,i}$ is the scaling func-

tion for the coarse scale coefficients that capture low frequency information of the input signal. $\omega_{j,i}$ and $c_{j_0,i}$ in Eq. 2 are the scaling functions of the detailed coefficient and ψ is the mother wavelet function. For the purpose of this paper, Daubechies 4 is used as the mother wavelet.

- **Filtering Process:**
The signal undergoes decomposition through high-pass and low-pass filters. The low-pass filter isolates high-frequency components, while the high-pass filter manages the residual part.
- **Downsampling:**

Downsampling the output of the filters by 2 yields the final approximation coefficients and detailed coefficients for the time series.

- **Separate Time Series Output:**
The resulting approximation and detailed coefficients form separate time series. These can be extracted for further analysis and forecasting, offering valuable insights into the underlying patterns of stock prices.

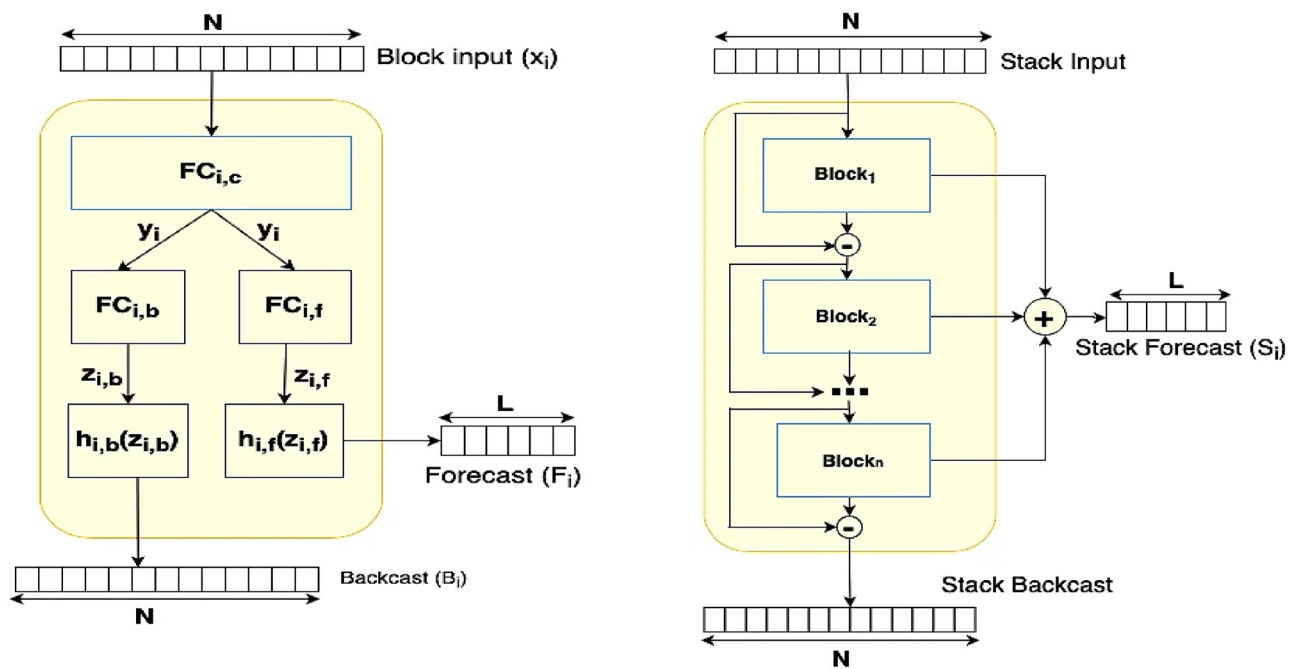
In Algorithm 1 the data decomposition technique is represented.

Algorithm 1 Data decomposition using wavelet transform

- 1: **Input:** Time series data $x(t)$
- 2: **Output:** Approximation coefficients $c_{j_0,i}$ and detailed coefficients $\omega_{j,i}$
- 3: **Step 1: Selection of Wavelet Transform Type**
- 4: Choose Discrete Wavelet Transform (DWT) for its applicability in decomposing time series data.
- 5: **Step 2: Decomposition into Coefficients**
- 6: Apply DWT to decompose $x(t)$ into approximation coefficients $c_{j_0,i}$ and detailed coefficients $\omega_{j,i}$.
- 7: **Step 3: Filtering Process**
- 8: Decompose the signal through high-pass and low-pass filters.
- 9: Low-pass filter isolates high-frequency components, while the high-pass filter manages the residual part.
- 10: **Step 4: Downsampling**
- 11: Downsample the output of filters by 2 to obtain final approximation and detailed coefficients.
- 12: **Step 5: Separate Time Series Output**
- 13: The resulting coefficients form separate time series for further analysis and forecasting.

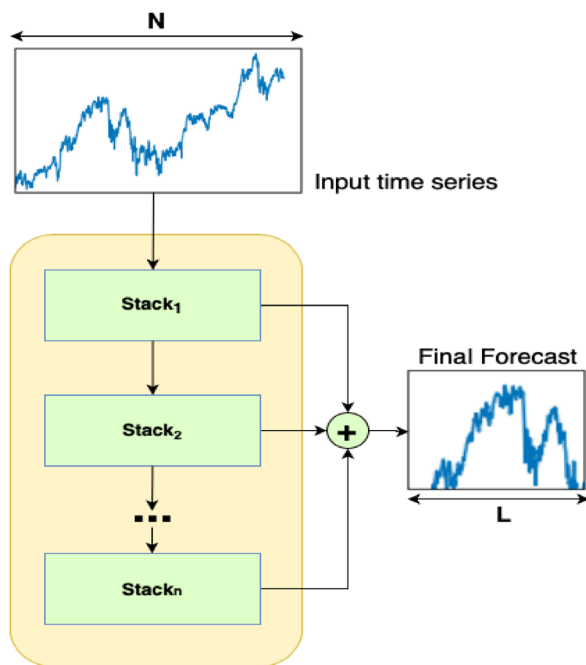
Table 2 Dataset Description

| Index | Country | Period | Days | Mean | SD | Minimum | Maximum |
|------------|-----------|--------------------------------|------|-----------|-----------|-----------|-----------|
| DJIA | USA | January 2000 to September 2019 | 4967 | 13,866.79 | 4998.56 | 6547.04 | 27,359.16 |
| NIFTY 50 | India | January 2000 to November 2019 | 4954 | 5090.22 | 3186.99 | 854.20 | 12,151.15 |
| Nikkei 225 | Japan | January 2005 to December 2019 | 3671 | 14,999.50 | 4544.56 | 7054.97 | 24,270.61 |
| BSE SENSEX | India | January 2000 to December 2019 | 4922 | 16,899.61 | 10,656.05 | 2600.12 | 41,681.53 |
| HSI | Hong Kong | January 2005 to December 2019 | 3688 | 22,053.99 | 4321.95 | 11,015.83 | 33,154.12 |



(a) block level architecture

(b) stack level architecture



(c) end to end architecture

Fig. 2 N-BEATS architecture

Table 3 Hyperparameter values for grid search for WT + N-BEATS

| Hyperparameter | Possible values |
|------------------------|----------------------------------------------------------------------------------------------------------|
| backcast_length | [19] |
| forecast_length | [1] |
| stack_types | N-BEATSNet.TREND_BLOCK, N-BEATSNet.SEASONALITY_BLOCK, N-BEATSNet.GENERIC_BLOCK, N-BEATSNet.GENERIC_BLOCK |
| nb_blocks_per_stack | [2, 4, 6] |
| thetas_dim | [(4, 4, 4, 4), (8, 8, 8, 8), (12, 12, 12, 12)] |
| share_weights_in_stack | [True, False] |
| hidden_layer_units | [64, 128, 256] |
| learning_rate | [1×10^{-5} , 1×10^{-4} , 1×10^{-3}] |
| batch_size | [8, 16, 32] |

N-BEATS Model for Time Series Forecasting

The N-BEATS architecture, introduced by [1], presents a deep neural network design specifically tailored for time series forecasting. It leverages backward and forward residual links along with a deep stack of fully connected layers, demonstrating its versatility and efficiency. The architecture is notable for its simplicity, generic applicability, and independence from time series-specific feature engineering. The forecasting is performed in following stages:

- **Basic Building Block:** Given an input series of length N as a univariate time series, N-BEATS aims to forecast a series of length L representing future values of the input series. The basic structure of N-BEATS is depicted in Fig. 6. For this paper, N was chosen to be 19, and L was chosen to be 1. The reason for choosing N to be 19 is that every month has about 20 business days, so in this paper, N-BEATS is used to forecast one day ahead closing price of a stock index using the past month data. The fundamental building block of the N-BEATS architecture

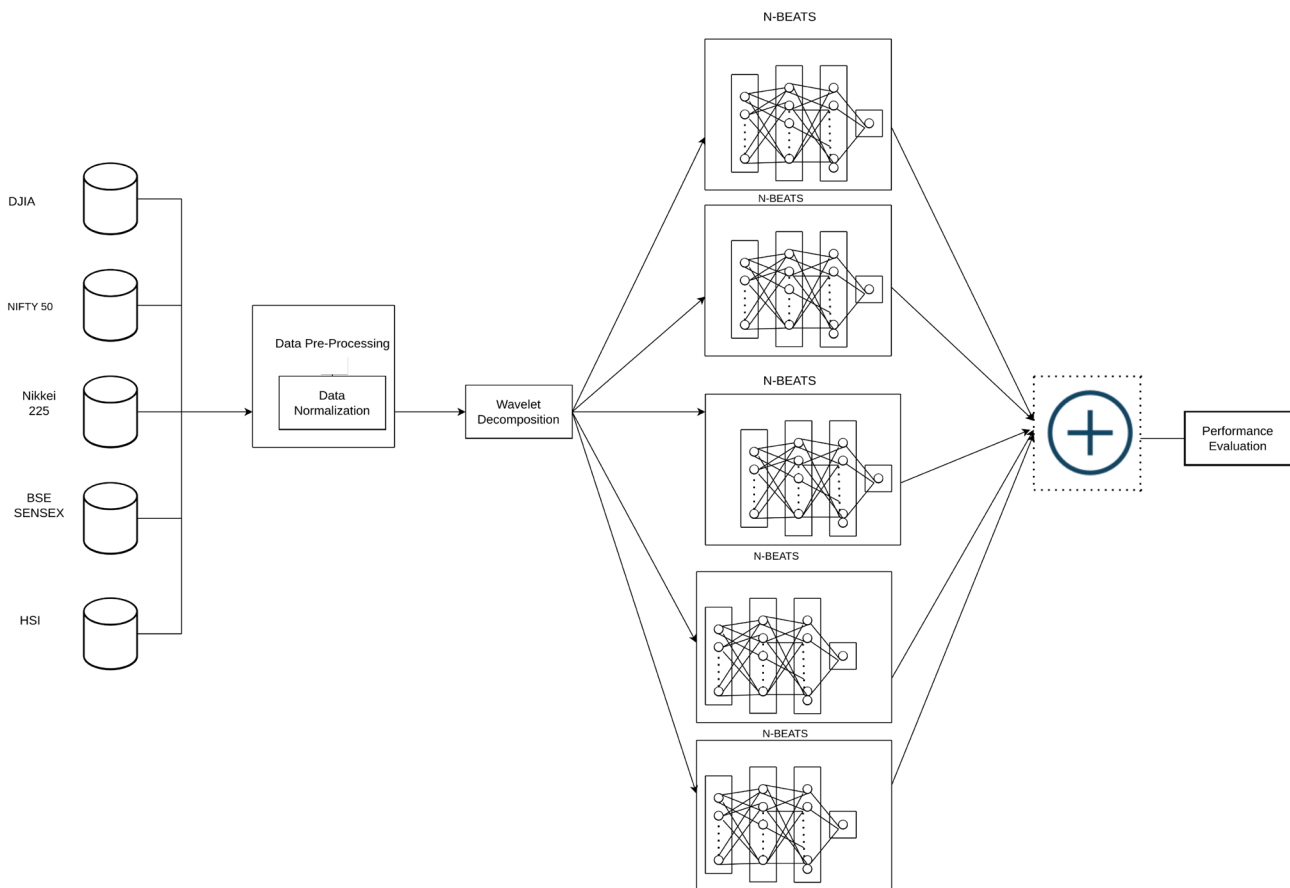


Fig. 3 Overview of implementation using WT + N-BEATS

comprises a four-layered, fully connected network with Rectified Linear Unit (ReLU) activation function. This block is applied to the input series, generating an intermediate vector y_i using Eq. 3.

$$y_i = FC_{i,c}(x_i). \tag{3}$$

- **Forward and Backward Expansion Coefficients:** The results from the fully connected network are fed into two separate fully connected layers ($FC_{i,f}$ and $FC_{i,b}$), producing forward expansion coefficients ($z_{i,f}$) and backward expansion coefficients ($z_{i,b}$) as depicted in Eqs. 4 and 5.

$$z_{i,f} = FC_{i,f}(y_i) \tag{4}$$

$$z_{i,b} = FC_{i,b}(y_i). \tag{5}$$

- **Block Level Forecasts and Backcasts:** The forward expansion coefficients ($z_{i,f}$) are projected onto a set of functions to yield block-level forecasts (F_i), while the backward expansion coefficients ($z_{i,b}$) are used to calculate backcasts (B_i). These backcasts are then utilized to calculate the input for the next similar block, as shown in Eqs. 6 and 7.

$$F_i = h_{i,f}(z_{i,f}) \tag{6}$$

$$B_i = h_{i,b}(z_{i,b}). \tag{7}$$

- **Stacking Blocks and Series Calculation:** Multiple blocks are stacked together to form a stack, and the input for each block in the series is calculated by subtracting the backcasts from the previous block’s input (Eq. 8). This stack of blocks is then connected in a series, forming the overall structure of N-BEATS (Fig. 2).

$$x_i = x_{i-1} - b_{i-1}. \tag{8}$$

- **Forecast Output:** The forecasts from each block within a stack are summed together to calculate each stack’s

forecast output (S_j), as demonstrated in Eq. 9. The final forecast output (F_{final}) is obtained by summing the forecast outputs from all the stacks, as outlined in Eq. 10.

$$S_j = \sum_{i=1}^n F_i \tag{9}$$

$$F_{final} = \sum_{j=1}^m S_j. \tag{10}$$

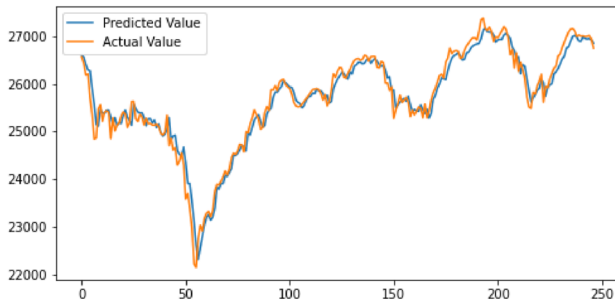
Performance Evaluation

Evaluating performance is a critical aspect of gauging the efficacy of forecasting models, and three commonly utilized metrics for this purpose include mean absolute error (MAE), root mean squared error (RMSE), and mean absolute percentage error (MAPE).

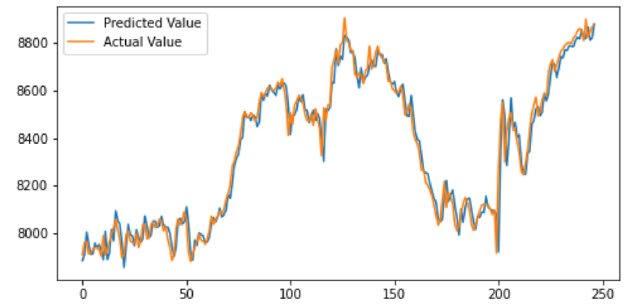
MAE quantifies the average absolute disparities between predicted and actual values, providing a straightforward measure of forecasting accuracy. In contrast, RMSE incorporates squared errors, emphasizing larger deviations. Presenting both MAE and RMSE in the same units as the predicted values facilitates easy interpretation. MAPE provides insights into the relative accuracy of forecasts and is particularly beneficial for cross-dataset performance comparisons as it computes the average percentage difference between predicted and actual values. Each metric contributes a distinctive viewpoint to the evaluation process, empowering practitioners to gain a comprehensive understanding of the strengths and limitations of a forecasting model. The selection of these metrics depends on the specific objectives and attributes of the forecasting task under consideration. The Mean Absolute Error is calculated as:

Table 4 Neural network architecture and configuration details

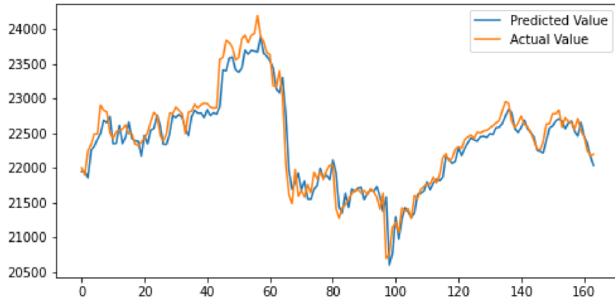
| Neural network architecture and configuration | Value |
|-----------------------------------------------|--------------------------------------------------------------|
| Model | N-BEATS |
| Number of hidden layers | 4 |
| Size of input | 4 |
| Number of neurons in hidden layers | 128 |
| Activation function for hidden layers | Rectified Linear Unit (ReLU) |
| Activation function for output layer | Linear |
| Dropout technique | Standard dropout ($p = 0.1$) (Derived using Grid-SearchCV) |
| Batch size | 10 |
| Epochs | 100 |



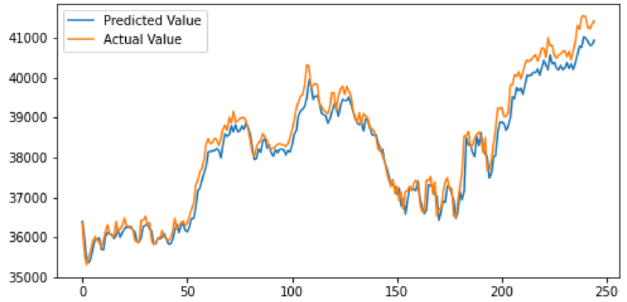
(a) DJIA



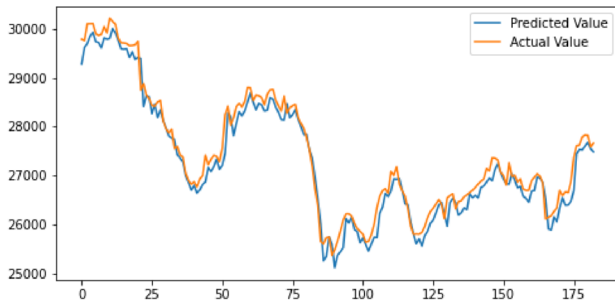
(b) NIFTY 50



(c) Nikkei 225



(d) BSE Sensex



(e) Hang Seng Index

Fig. 4 Prediction results of LSTM model

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \tag{11}$$

The Root Mean Squared Error is computed as:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \tag{12}$$

The Mean Absolute Percentage Error is given by:

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100\% \tag{13}$$

where n represents the number of observations or data points, y_i is the actual (observed) value for the i th data point and is \hat{y}_i the predicted (forecasted) value for the i th data point.

Implementation and Discussion

Initially, all the datasets were divided into three parts: the training set, the validation set, and the test set. The training set was used for training the models and calculating the parameters. The validation set was used to find the hyperparameters for the model (Table 3), and the test set was reserved for out-of-sample forecasting and comparison of performances among various models. The min–max

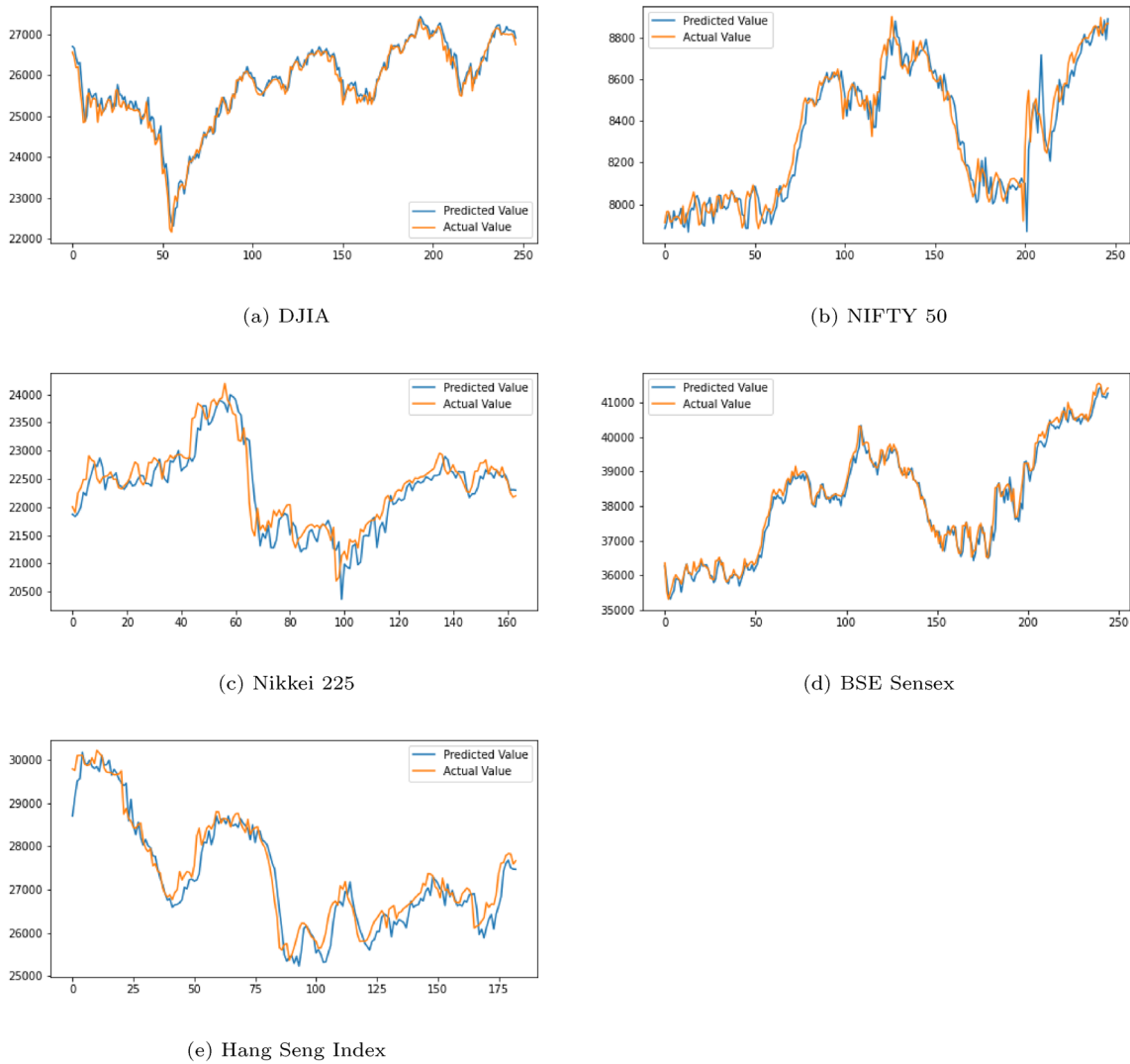


Fig. 5 Prediction results of CNN model

normalization technique expressed in Eq. 14 was used for scaling the data.

We have done parameter tuning for this purpose with following details:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}. \quad (14)$$

The data was decomposed into approximation coefficients and detailed coefficients using wavelet transformation. Daubechies 4 wavelet was used for the wavelet transformation in the experimental studies of this paper. The approximation coefficients and detailed coefficients calculated by wavelet transformation have different structures, and each of them was a time series in itself. Separate N-BEATS models were trained for forecasting each of the

obtained approximation coefficients and detailed coefficients. Finally, all the forecast values obtained for different coefficients from different N-BEATS models were additively combined to get the final forecast of the stock market index price. The overview of this implementation is shown in Fig. 3.

This approach of individually forecasting the detailed coefficients and approximation coefficients using separate N-BEATS models and then summing them up for the final forecast provided a better result than using only N-BEATS.

Table 4 provides the detailed hyper parameter analysis of proposed model to reproduce the result.

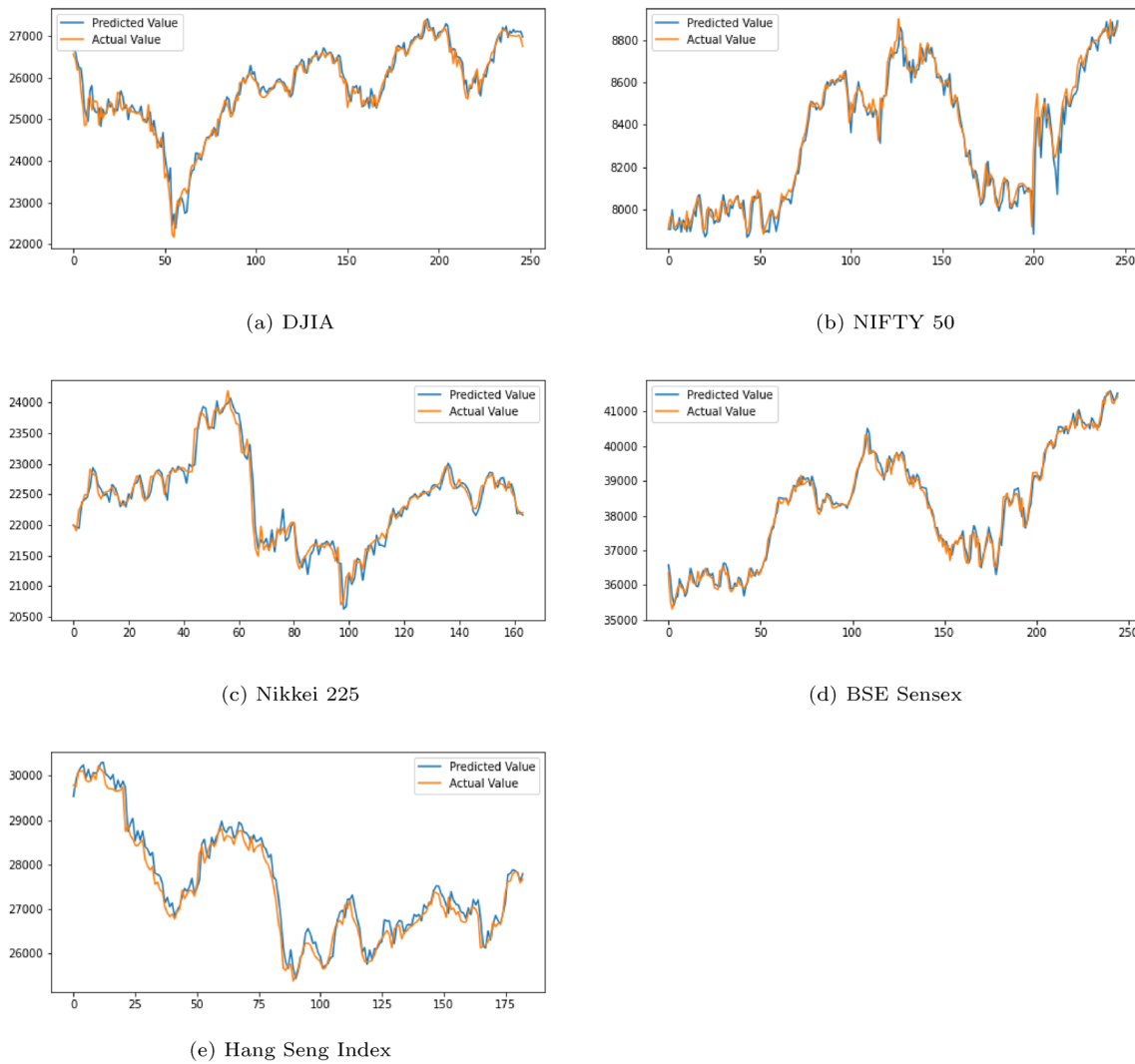


Fig. 6 Prediction results of N-BEATS model

Experimental Results

We have applied the WT + N-BEATS architecture proposed in this paper on five different stock market indices, namely Nifty 50, DJIA, Nikkei 225, BSE SENSEX, and HSI. For each index, 90% of the data is used as training data and 5% of the data is used for validation and test set each. Mean Absolute Percentage Error (MAPE), Root Mean Square Error (RMSE), and Mean Absolute Error (MAE) are used as primary metrics for comparing the proposed WT+N-BEATS architecture with some traditional deep learning approaches. Mathematical representations for MAE, RMSE and MAPE are given by Eqs. 15, 16 and 17 respectively. In these equations y_t refers to true value of the t th sample, \hat{y}_t refers to true value of the t th sample and n refers to the number of samples. In this paper, the proposed WT + N-BEATS architecture is compared to

the standalone N-BEATS model, LSTM model, and the CNN model. The experimental results obtained for each of the models are discussed in this section.

$$MAE = \frac{\sum_{t=1}^n |y_t - \hat{y}_t|}{n} \tag{15}$$

$$RMSE = \sqrt{\frac{\sum_{t=1}^n (y_t - \hat{y}_t)^2}{n}} \tag{16}$$

$$MAPE = \frac{100\%}{n} \sum_{t=1}^n \frac{y_t - \hat{y}_t}{y_t} \tag{17}$$

In the realm of stock market prediction, Long Short-Term Memory (LSTM) serves as a powerful tool, delving into historical data to discern intricate patterns and trends. This

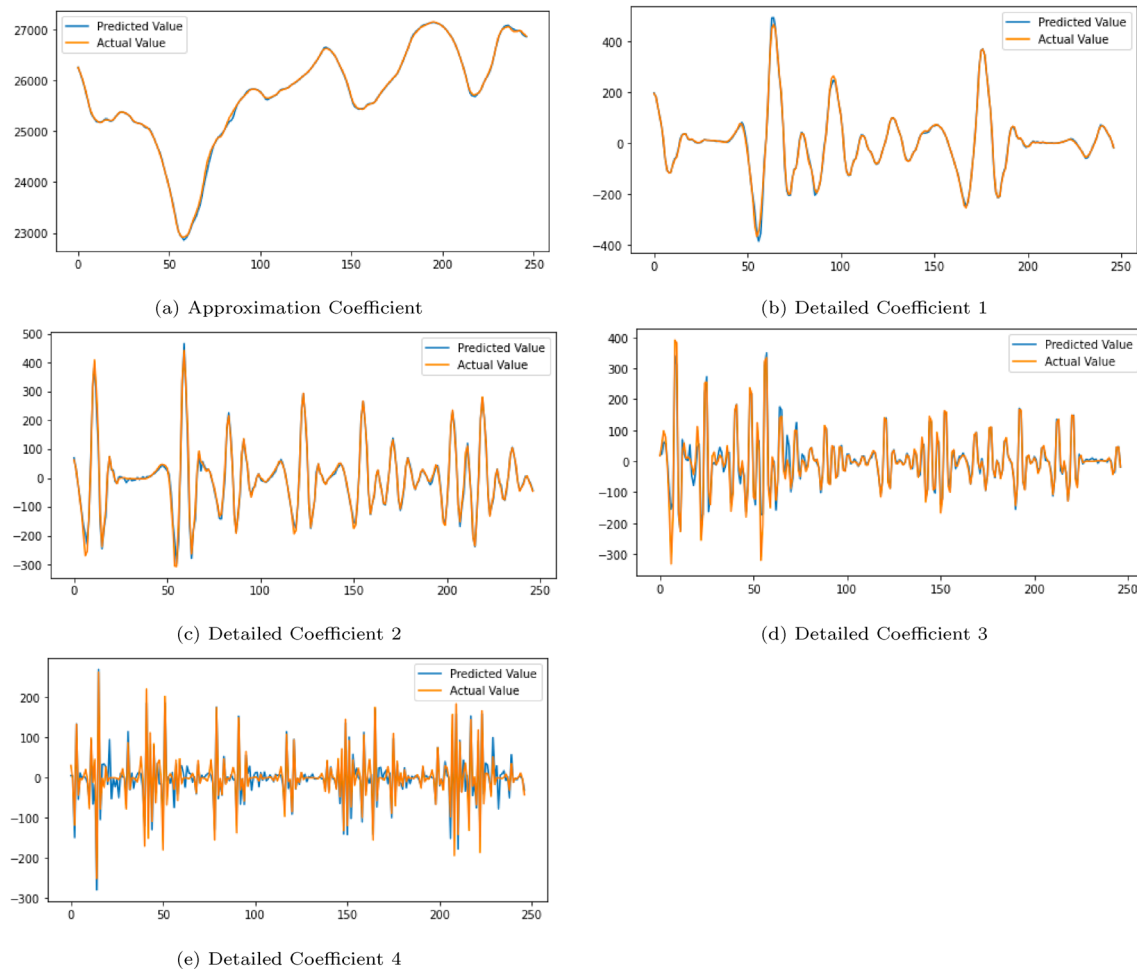


Fig. 7 Prediction results for each coefficient of DJIA

acquired knowledge is then harnessed to make informed forecasts about future stock prices. However, it is imperative to recognize the inherent complexity and dynamism of stock market prediction. While LSTM stands as a valuable asset in this context, it should be viewed as only one element within a larger and more encompassing approach to stock market analysis and prediction. Successful prediction in the financial domain demands a holistic strategy that incorporates various methodologies and factors to account for the multifaceted nature of market behavior.

Figure 4 shows the results of the predictions made by a LSTM model on the used datasets.

The network learns to recognize patterns and correlations in historical stock data, which may impact future stock prices, making CNNs a valuable tool for analyzing such data. This approach can help identify complex relationships within the data that may not be apparent through traditional statistical analysis. However, it's important to note that using CNNs for stock market prediction requires careful consideration of the data preprocessing, model

architecture, and evaluation metrics. While CNNs can be a powerful tool for feature extraction and pattern recognition, they are just one part of a larger toolkit for stock market prediction, and their effectiveness depends on the quality and relevance of the input data.

Figure 5 shows the results of the predictions made by a CNN model on the used datasets.

N-BEATS has garnered attention in the evolving landscape of stock market prediction owing to its capacity for precise forecasts. Deep neural networks are used by N-BEATS to understand the complex changes in stock prices. They do this by breaking down time series data into basic building blocks called basis functions. While traditional forecasting methods have historically facilitated strategic decision-making, the advent of deep learning, epitomized by N-BEATS, has revolutionized the domain by presenting unprecedented potential for accurate predictions of stock market movements. This innovative approach marks a significant stride in utilizing deep learning for stock market

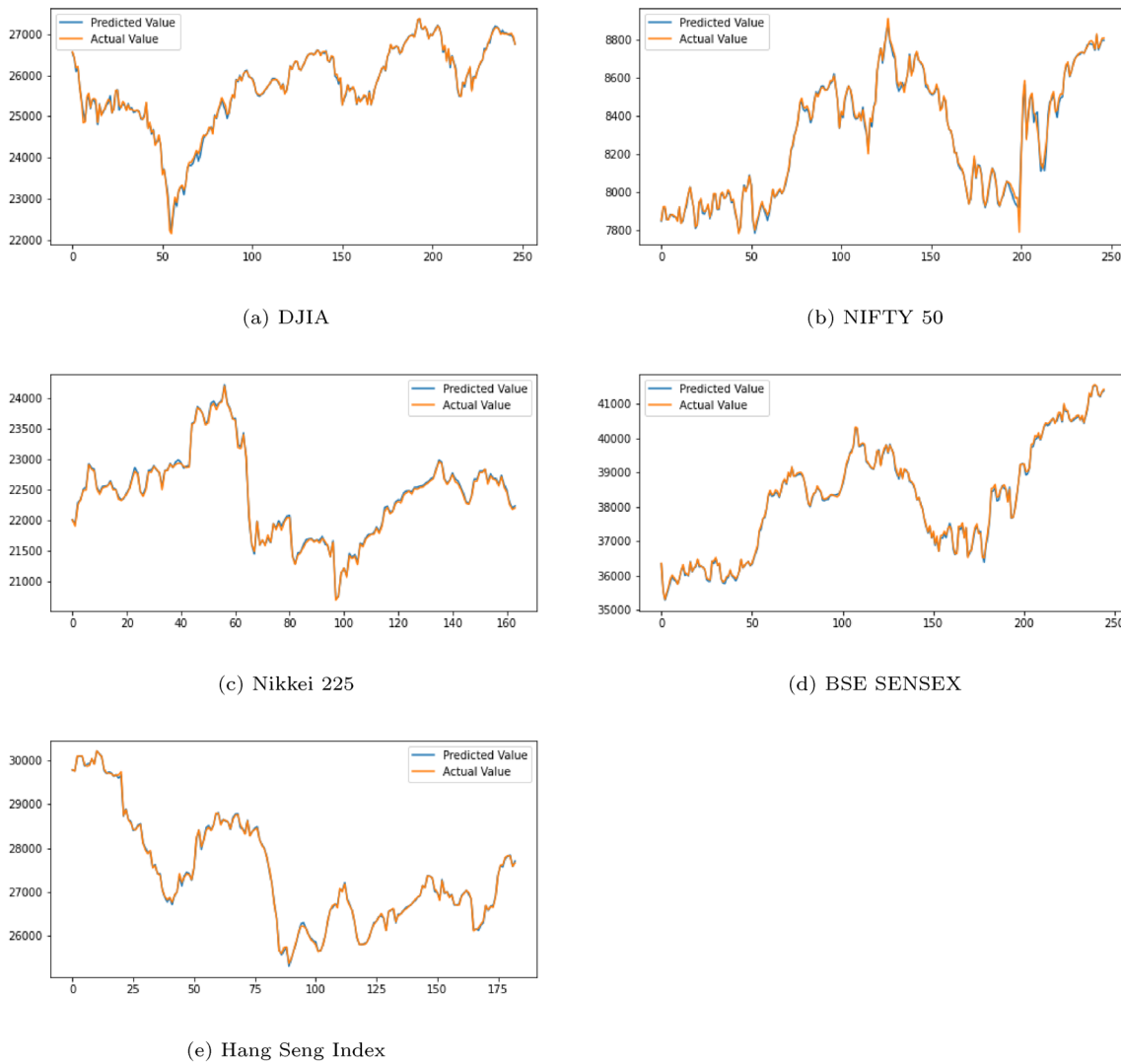


Fig. 8 Prediction results of WT + N-BEATS model

prediction, contributing to a transformative impact on the accuracy and reliability of stock price forecasts.

Figure 6 shows the result of predictions made by the standalone N-BEATS models which were trained on closing price of the indices directly and not on the coefficients of wavelet transformations.

Figure 7 shows the actual values and the prediction made by the N-BEATS model for the approximation coefficients and detailed coefficients for the DJIA dataset. From these graphs it can be seen that N-BEATS is able to predict the approximation coefficients with high accuracy. For detailed coefficients as the structure of these coefficients become complex in each, the predictive power of N-BEATS decreases.

Figure 8 shows the result of predictions made by WT + N-BEATS models which were trained on the coefficients of

wavelet transformations for all the five stock market indices under consideration.

Figure 9 shows the forecasts made by the proposed WT + N-BEATS model and other models for the DJIA dataset. Forecasting results produced by different models for the NIFTY 50 dataset are shown in Fig 10. Figure 11 shows the forecasts made by the proposed WT+N-BEATS model and other traditional models for the Nikkei 225 dataset. Similarly, Figs. 12 and 13 show forecasts made for BSE SENSEX and HSI dataset respectively. Figures 9, 10, 11, 12 and 13 summarises results obtained in the above analysis and help us get some useful insights. Various models exhibit their performance in a comprehensive overview, enhanced by the lucidity of our findings through Fig. 14.

Table 5 summarises the results obtained from different models for different datasets. Results for the proposed

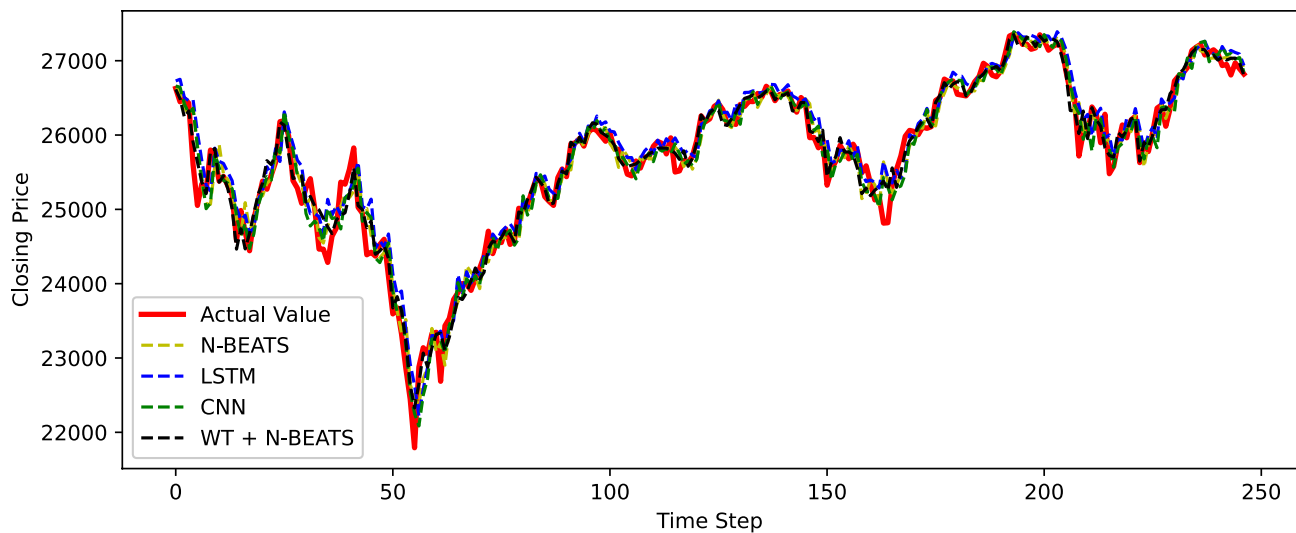


Fig. 9 Combined results for DJIA

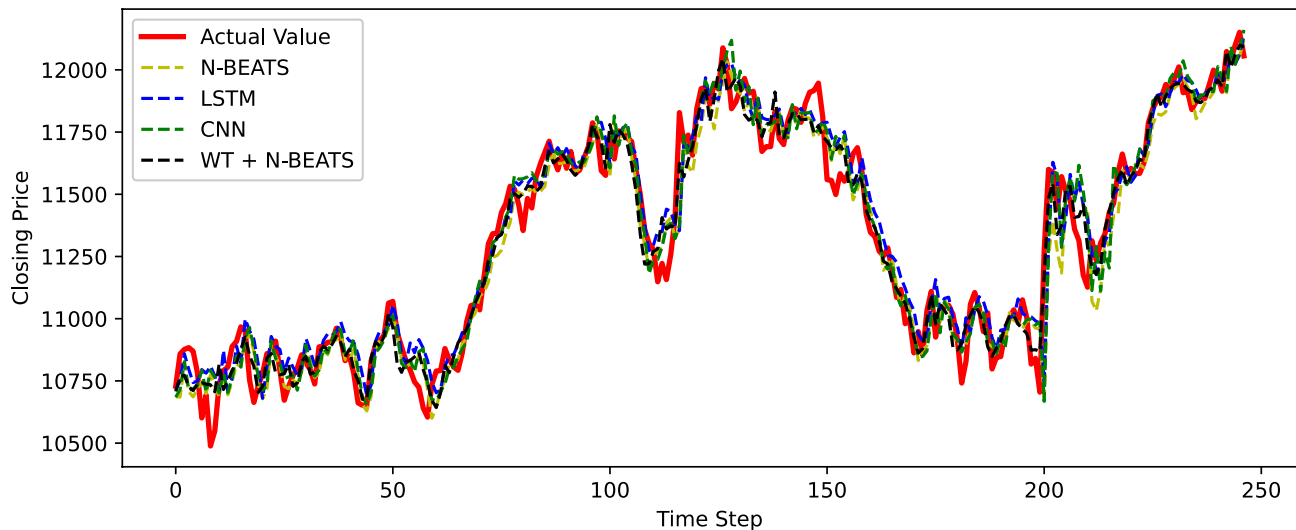


Fig. 10 Combined results for NIFTY 50

architecture are highlighted in Table 5. Figure 15 draws a visual comparison between different models used in this work. Table 5 and Fig. 15 suggests that the proposed WT+N-BEATS model outperforms all other models used in this study. Hence through this empirical study, it can be concluded that the proposed WT+N-BEATS architecture can improve stock market forecasting. To ascertain the significance of the results obtained in this work, we conducted Wilcoxon signed-rank test and Friedman test, the results of which are summarised in Table 6.

The Wilcoxon signed-rank test is a non-parametric statistical test used to compare two paired groups of data when the assumption of normality is violated or when the data is ordinal. It assesses whether the median of the differences between paired observations is significantly different from zero. The steps for conducting the test involve ranking the absolute differences between pairs, disregarding their signs, and then summing the ranks of the positive or negative differences. The test statistic is based on the sum of the ranks, with larger values indicating greater evidence against the null hypothesis of no difference. The critical value of the test

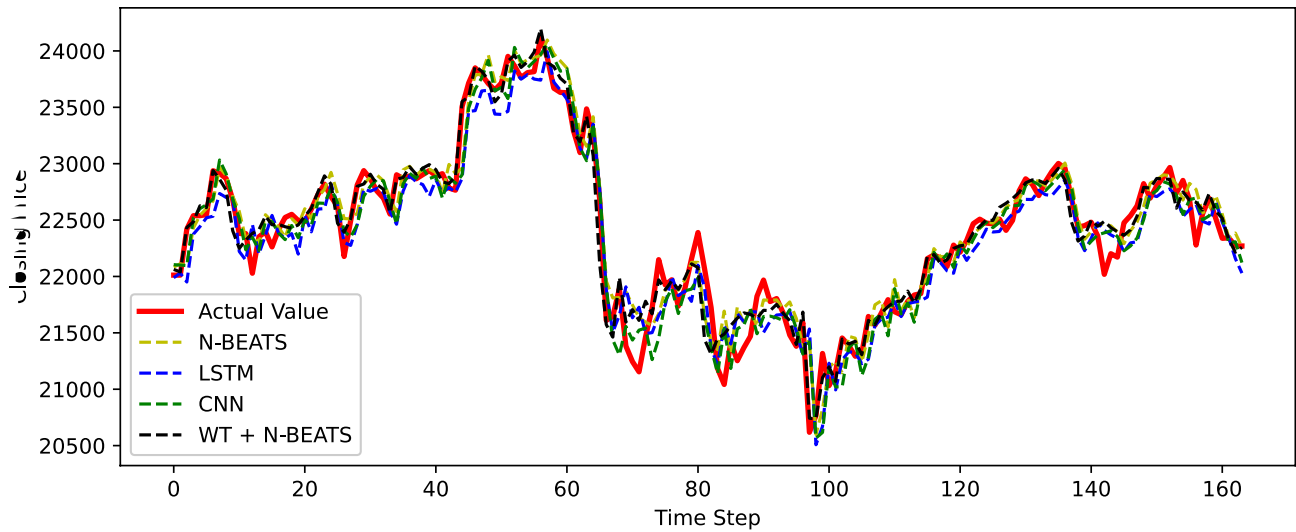


Fig. 11 Combined results for Nikkei 225

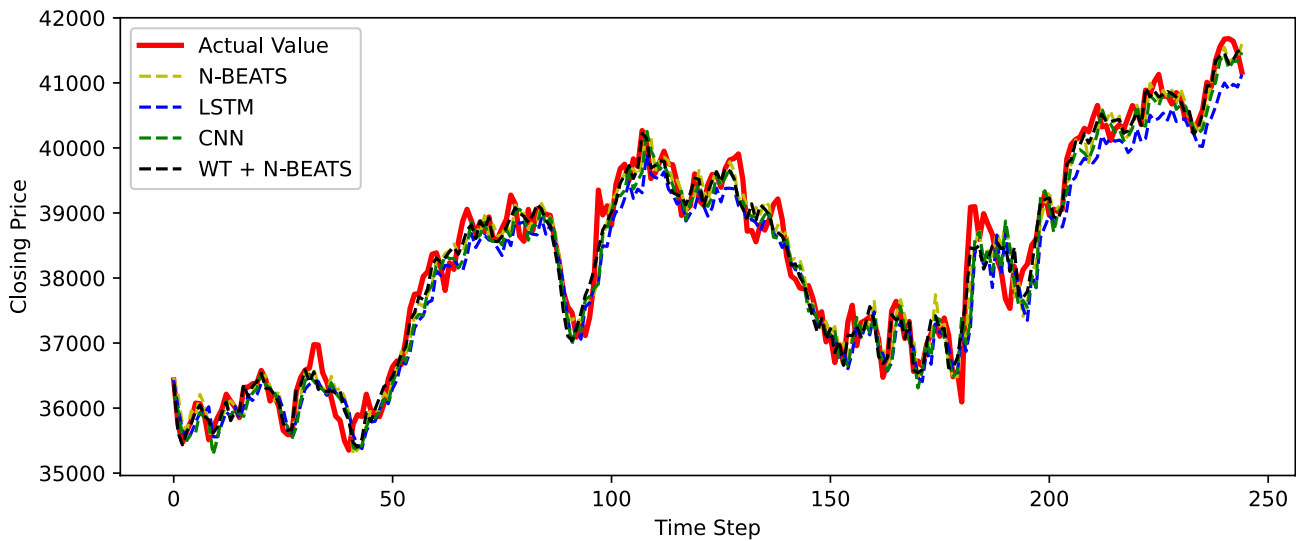


Fig. 12 Combined results for BSE SENSEX

statistic is compared to a table of critical values or obtained from software to determine statistical significance. It's commonly used in research settings where data doesn't meet the assumptions of parametric tests like the paired t-test, such as in psychology or biology. The Wilcoxon signed-rank test provides a robust alternative for analyzing paired data without requiring assumptions about the underlying distribution.

The Friedman test is a non-parametric statistical test used to compare the means of three or more paired groups

when the data violates assumptions of normality and homogeneity of variances. It assesses whether there are statistically significant differences among the groups. The procedure involves ranking the data within each group and calculating the average rank for each observation. Then, a test statistic is computed based on the differences between the average ranks. The null hypothesis is that there are no differences among the groups. If the test statistic is significant, it suggests that at least one group differs from the

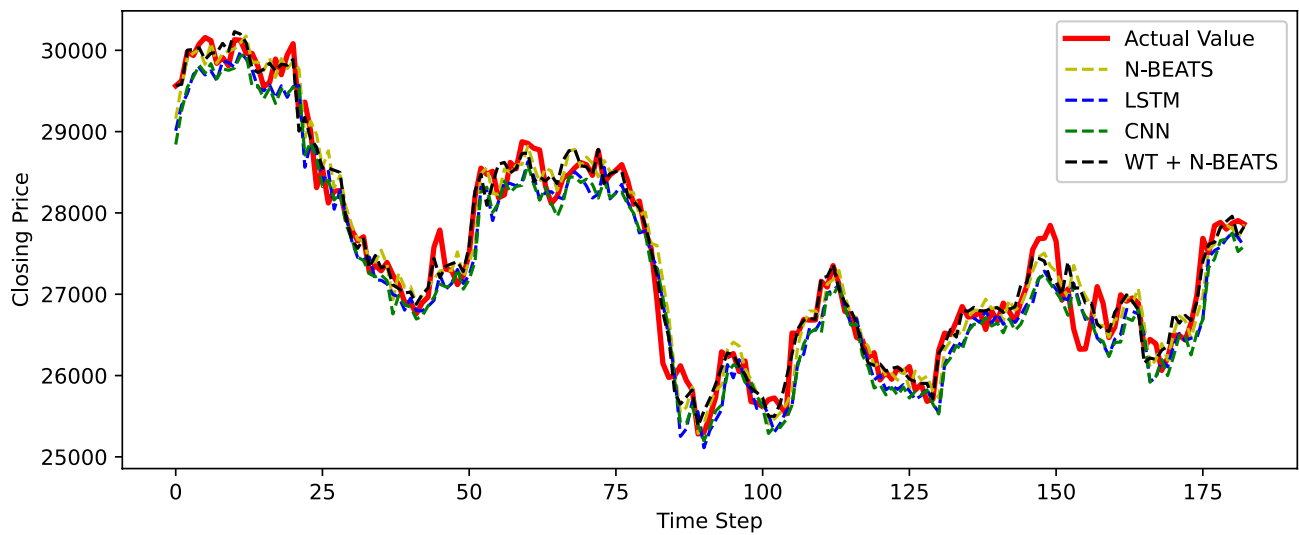


Fig. 13 Combined results for HSI

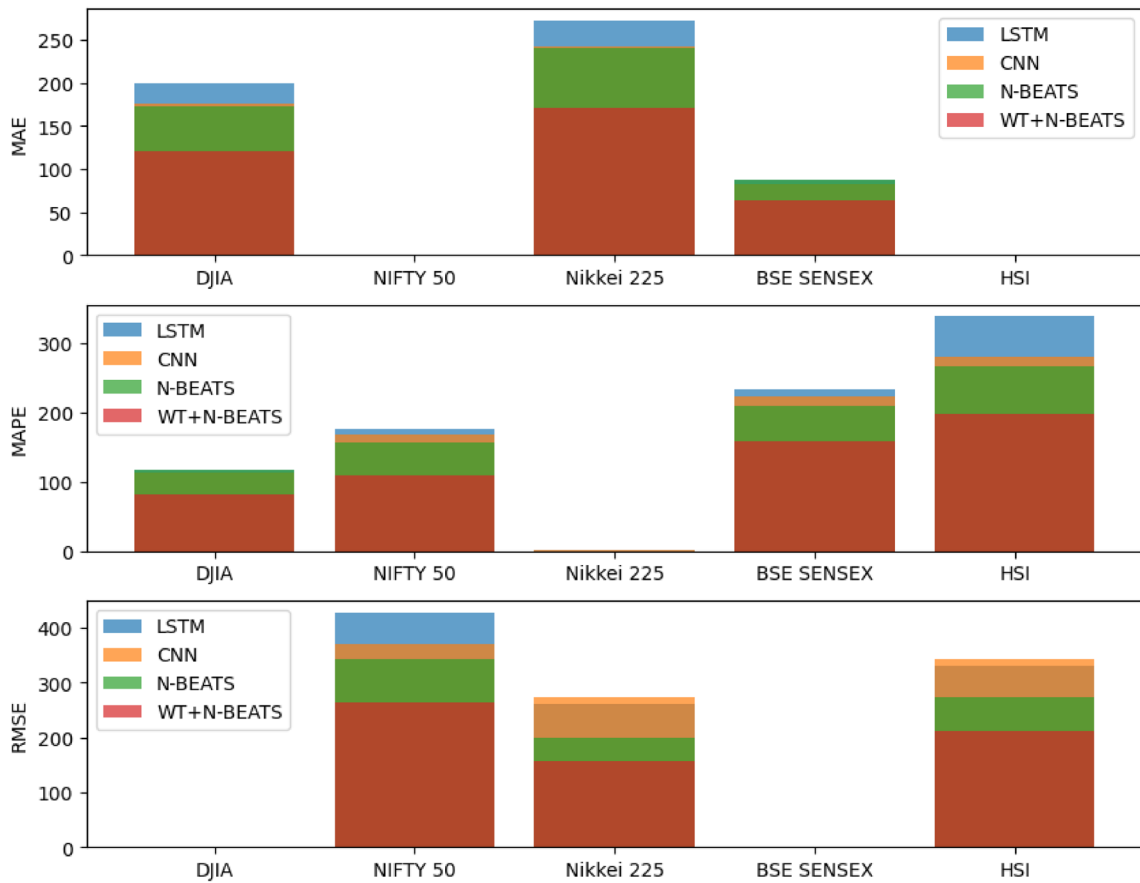


Fig. 14 Comparison of models for different indices and metrics

Table 5 Final results for all models

| Index | Metric | LSTM | CNN | N-BEATS | WT + N-BEATS |
|------------|--------|----------|----------|----------|-----------------|
| DJIA | MAE | 199.6460 | 176.0111 | 172.4302 | 121.1623 |
| | MAPE | 0.7909 | 0.6955 | 0.6828 | 0.4810 |
| | RMSE | 272.5661 | 242.7572 | 240.0762 | 170.8828 |
| NIFTY 50 | MAE | 87.7324 | 83.5523 | 87.7324 | 63.1464 |
| | MAPE | 0.7763 | 0.7412 | 0.7764 | 0.5612 |
| | RMSE | 117.6041 | 114.3441 | 117.6041 | 81.3019 |
| Nikkei 225 | MAE | 175.4711 | 168.4006 | 157.3350 | 109.7702 |
| | MAPE | 0.7875 | 0.7571 | 0.7085 | 0.4947 |
| | RMSE | 233.2632 | 223.7306 | 208.9863 | 158.1120 |
| BSE SENSEX | MAE | 338.3028 | 280.8321 | 265.8805 | 196.8925 |
| | MAPE | 0.8778 | 0.7345 | 0.6973 | 0.5162 |
| | RMSE | 428.4140 | 370.4772 | 343.7019 | 263.2523 |
| HSI | MAE | 260.3581 | 274.4688 | 200.2134 | 157.0981 |
| | MAPE | 0.9471 | 0.9983 | 0.7336 | 0.5751 |
| | RMSE | 330.7833 | 342.1693 | 273.5654 | 211.5542 |

The best outcomes in the comparative analysis is marked in bold

Table 6 Results for Wilcoxon signed-rank test and Friedman Test

| Models | Wilcoxon signed-rank test ($\alpha = 0.05$) | Friedman test ($\alpha = 0.05$) |
|----------------------------------------------------|-----------------------------------------------|-----------------------------------|
| WT + N-BEATS versus LSTM | 0.0000** | NA |
| WT + N-BEATS versus CNN | 0.0000** | NA |
| WT + N-BEATS versus N-BEATS | 0.0000** | NA |
| WT + N-BEATS versus N-BEATS versus CNN versus LSTM | NA | 0.0000** |

**p value is less than 0.05

others. The critical value of the test statistic is compared to a chi-squared distribution to determine statistical significance. The Friedman test is commonly used in fields such as psychology, medicine, and environmental sciences when dealing with repeated measures designs and non-normally distributed data. It provides a robust alternative to parametric tests like repeated measures ANOVA when assumptions are violated.

Table 6 shows that both the tests reject the null hypothesis and clearly conclude that the results obtained by the proposed method are statistically different as compared to the results obtained from other methods considered in this paper.

Implications of this Study Stock market predictions play a vital role in overall financial planning and corporate financial decisions. The proposed ML models can help the end users in various capacities like: helping them make smart decisions, enhance their portfolio management, analyse risk and returns, contribute to market efficiency by incorporating new information into stock prices etc. Overtime with more data availability, the accuracy of the ML models is expected to increase. This provides a win-win situation both for the end users as well as business firms for formulating future strategies. Stock market projections shape economic expectations and individual and corporate financial decisions. Financial markets are unpredictable, so investors must be careful and diversified. Predictions can drive investment strategies and economic policies. Planners can trust long-term forecasts, but investors and end-users must respond to shifting economic conditions and market signals. Although the usage of ML based approaches can be beneficial, but there are various external factors like government policies, international relations, sudden natural disasters, wars, acquisitions etc., that do have an impact the stock market forecasting. So, one should always be a little cautious while adopting these models.

Limitations of this Study: We have used an ensemble of Wavelet Decomposition and N-BEATS for improved stock

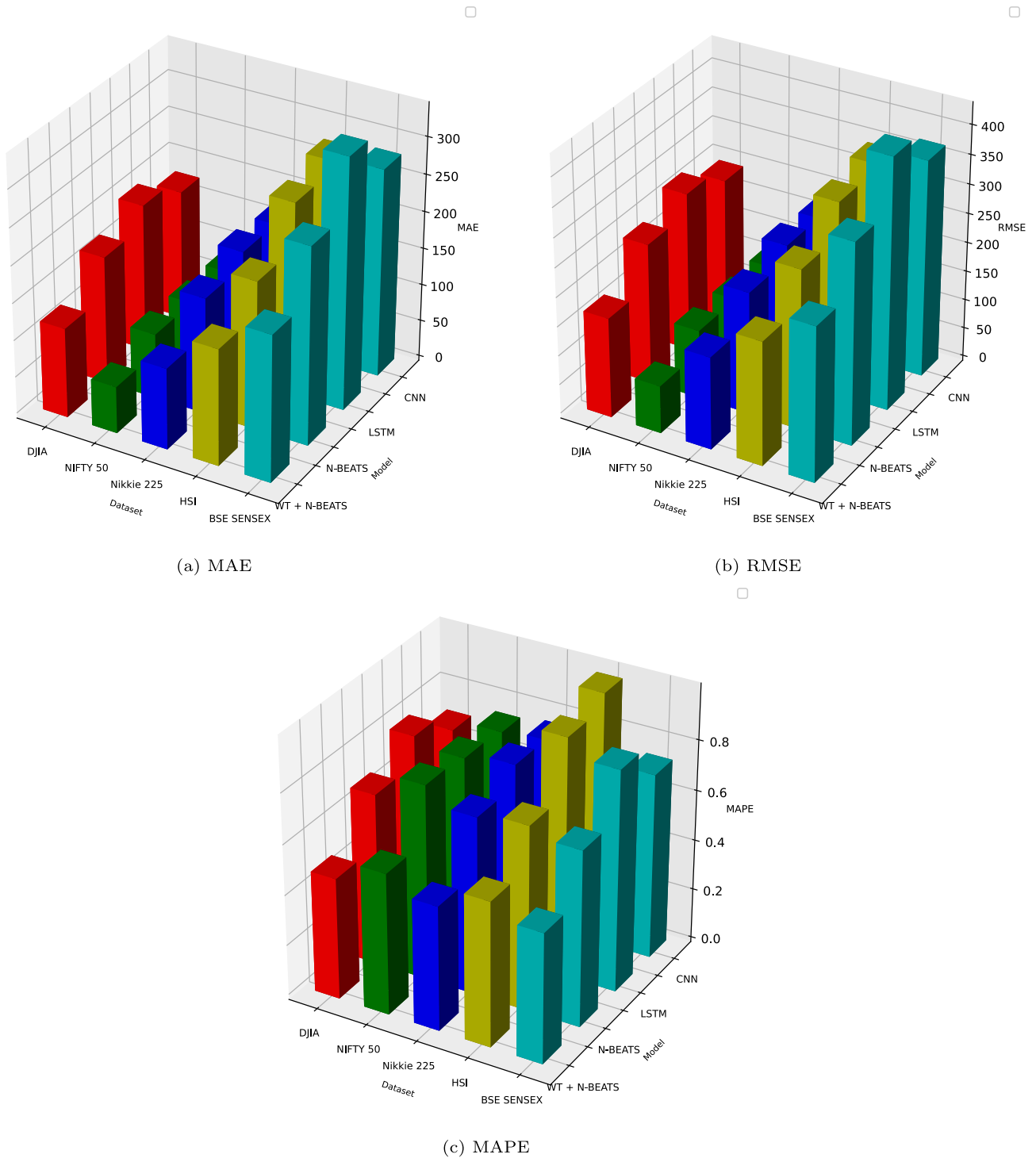


Fig. 15 Error metrics for various models

market forecasting. Wavelet transforms are highly effective tools for signal processing, especially when it comes to analyzing time-series data that is not steady. Nevertheless, these methods have various drawbacks, such as their inefficacy in handling specific types of data, strong correlation and lack of

orthogonality, restrictions in dealing with discrete time series, and difficulties in signal reconstruction. The issues in real-time systems are compounded by computational complexity and the requirement to strike a balance between compression ratio, quality, and latency. Stock forecasting data being time series in

nature and is prone to sudden changes overtime, it is possible that the data distribution of a given stock on a day/hour may result into a non-desired prediction due to inherent randomness of the data under consideration.

Conclusion

Precise forecasting models are crucial for informed decision-making in stock markets. This study highlights the effectiveness of N-BEATS in predicting future prices for prominent stock market indices, including DJIA, NIFTY 50, BSE SENSEX, HSI, and Nikkei 225. Experimental results reveal that a basic N-BEATS model performed competitively, occasionally outperforming traditional deep learning time series predictors like LSTM and CNN. Notwithstanding these successes, it is important to acknowledge the inherent limitations and challenges encountered during the research.

The analysis of forecasting models entails assessing key performance metrics such as mean absolute error (MAE), mean squared error (MSE), and root mean squared error (RMSE). We execute a comprehensive comparison with baseline models like LSTM and CNN, supported by p -values. We employ cross-validation techniques to gauge model stability and generalization. We explore the robustness of the N-BEATS model across diverse datasets, emphasizing variations and limitations. Statistical tests are used to see if the combined wavelet transformation and N-BEATS architecture is a good way to improve performance and make computations easier. We scrutinize external factors that affect stock market dynamics through correlation and regression methods. Visual representations, such as time series plots and prediction versus actual price charts, offer insights, while sensitivity analysis delves into the impact of key parameters on forecasting accuracy. In summary, the proposed WT + N-BEATS architecture consistently surpasses traditional models, marking a significant stride in enhancing stock price forecasting accuracy, despite acknowledged challenges and limitations.

Even though combining wavelet transformation and N-BEATS works, it might be hard to do on a computer, and the performance gains seen may be different for different datasets. The ongoing challenge of external factors influencing stock market dynamics further emphasizes the need for accurate predictions. Despite these challenges, our proposed WT + N-BEATS architecture consistently outperformed simple N-BEATS models and traditional architectures across all tested stock market indices, representing a significant advancement in enhancing stock price forecasting accuracy.

Author Contributions Neha Pramanik: Formal analysis, Data curation, Original draft preparation and Writing. Vatsal Singhal: Implementation and Writing. Neeraj: Writing—Conceptualization, Original draft preparation and Writing. Jimson Mathew: Research profile design, Writing, Reviewing, and Revising. Mayank Agarwal: Research profile design, Reviewing, and Revising.

Data Availability Some or all data, models, or code that support the findings of this study are available from the corresponding author upon reasonable request.

Declarations

Conflict of interest NA.

References

- Oreshkin BN, Dudek G, Peřka P, Turkina E. N-beats neural network for mid-term electricity load forecasting. *Appl Energy*. 2021;293:116918.
- Fama EF. Random walks in stock market prices. *Financ Anal J*. 1995;51(1):75–80.
- Adebiji A, Adewumi A, Ayo C. Stock price prediction using the ARIMA model. In: 2014 UKSim-AMSS 16th international conference on computer modelling and simulation; 2014. IEEE. <https://doi.org/10.1109/UKSim.2014.67>.
- Adebayo FA, Sivasamy R, Shangodoyin DK. Forecasting stock market series with ARIMA model. *Stat Econ Methods*. 2014;3:65–77.
- Luo S, Yan F, Lai D, Wu W, Lu F. Using ARIMA model to fit and predict index of stock price based on wavelet de-noising. *Int J u- and e- Serv Sci Technol*. 2016;9:317–26. <https://doi.org/10.14257/ijunesst.2016.9.12.28>
- Milosevic N. Equity forecast: predicting long term stock price movement using machine learning. *J Econ Libr*. 2016;3(2):288–94. <https://doi.org/10.1453/jel.v3i2.750>.
- Zhang J, Cui S, Xu Y, Li Q, Li T. A novel data-driven stock price trend prediction system. *Expert Syst Appl*. 2018;97:60–9. <https://doi.org/10.1016/j.eswa.2017.12.026>.
- Patel J, Shah S, Thakkar P, Kotecha K. Predicting stock market index using fusion of machine learning techniques. *Expert Syst Appl*. 2015;42(4):2162–72. <https://doi.org/10.1016/j.eswa.2014.10.031>.
- El-Rashidy MA. A novel system for fast and accurate decisions of gold-stock markets in the short-term prediction. *Neural Comput Appl*. 2021;33(1):393–407.
- Jin Z, Yang Y, Liu Y. Stock closing price prediction based on sentiment analysis and LSTM. *Neural Comput Appl*. 2020;32(13):9713–29.
- Lin C-T, Wang Y-K, Huang P-L, Shi Y, Chang Y-C. Spatial-temporal attention-based convolutional network with text and numerical information for stock price prediction. *Neural Comput Appl*. 2022;34:14387–95.
- Lu W, Li J, Wang J, Qin L. A CNN-BILSTM-AM method for stock price prediction. *Neural Comput Appl*. 2021;33(10):4741–53.
- Wang X, Phua P, Lin W. Stock market prediction using neural networks: does trading volume help in short-term prediction? In: *Proceedings of the International Joint Conference on Neural Networks*, 2003; vol. 4. p. 2438–2442. <https://doi.org/10.1109/IJCNN.2003.1223946>.

14. Bernal A, Fok S, Pidaparathi R. Market time series prediction with recurrent neural networks. State College: Citeseer. [Google Scholar], Citeseer; 2012.
15. Moghar A, Hamiche M. Stock market prediction using LSTM recurrent neural network. *Proc Comput Sci.* 2020;170:1168–73. <https://doi.org/10.1016/j.procs.2020.03.049>.
16. Qiu J, Wang B, Zhou C. Forecasting stock prices with long-short term memory neural network based on attention mechanism. *PLoS ONE.* 2020;15(1):e0227222.
17. Eapen J, Bein D. Novel deep learning model with CNN and bi-directional LSTM for improved stock market index prediction. In: 2019 IEEE 9th annual computing and communication workshop and conference (CCWC); 2019. p. 0264–0270. <https://doi.org/10.1109/CCWC.2019.8666592>.
18. Fryzlewicz P, Van Bellegem S, Von Sachs R. Forecasting non-stationary time series by wavelet process modelling. *Ann Inst Stat Math.* 2003;55(4):737–64.
19. Renaud O, Starck J, Murtagh F. Wavelet-based combined signal filtering and prediction. *IEEE Trans Syst Man Cybern Part B (Cybern).* 2005;35(6):1241–51. <https://doi.org/10.1109/TSMCB.2005.850182>.
20. Wang C, Chen Y, Zhang S, Zhang Q. Stock market index prediction using deep transformer model. *Expert Syst Appl.* 2022;208:118128.
21. Ghotbi M, Zahedi M. Predicting price trends combining kinetic energy and deep reinforcement learning. *Expert Syst Appl.* 2023;244: <https://doi.org/10.1016/j.eswa.2023.122994>.
22. Abraham R, Samad ME, Bakhach AM, El-Charani H, Sardouk A, Nemar SE, Jaber D. Forecasting a stock trend using genetic algorithm and random forest. *J Risk Financ Manag.* 2022;15(5):188.
23. Amin MS, Ayon EH, Ghosh BP, Bhuiyan MS, Jewel RM, Linkon AA. Harmonizing macro-financial factors and Twitter sentiment analysis in forecasting stock market trends. *J Comput Sci Technol Stud.* 2024;6(1):58–67.
24. Behera J, Pasayat AK, Behera H, Kumar P. Prediction based mean-value-at-risk portfolio optimization using machine learning regression algorithms for multi-national stock markets. *Eng Appl Artif Intell.* 2023;120:105843.
25. Zhao Y, Yang G. Deep learning-based integrated framework for stock price movement prediction. *Appl Soft Comput.* 2023;133:109921.
26. Agrawal M, Shukla PK, Nair R, Nayyar A, Masud M. Stock prediction based on technical indicators using deep learning model. *Comput Mater Contin.* 2022;70(1):287–304. <https://doi.org/10.32604/cmc.2022.014637>.
27. Mehta P, Pandya S, Kotecha K. Harvesting social media sentiment analysis to enhance stock market prediction using deep learning. *PeerJ Comput Sci.* 2021;7:e476.
28. Prachyachuwong K, Vateekul P. Stock trend prediction using deep learning approach on technical indicator and industrial specific information. *Information.* 2021;12(6):250.
29. Ma Y, Han R, Wang W. Portfolio optimization with return prediction using deep learning and machine learning. *Expert Syst Appl.* 2021;165:113973.
30. Stock market indices dataset. <https://github.com/VatsalSin/Stock-Market-Indices-Dataset>
31. Paramanik RN, Singhal V. Sentiment analysis of Indian stock market volatility. *Proc Comput Sci.* 2020;176:330–8. <https://doi.org/10.1016/j.procs.2020.08.035>.
32. Verma R, Verma P. Noise trading and stock market volatility. *J Multinatl Financ Manag.* 2007;17:231–43. <https://doi.org/10.1016/j.mulfin.2006.10.003>.
33. Dremmin I. Wavelets: mathematics and applications. *Phys Atomic Nucl.* 2005;68:508–20. <https://doi.org/10.1134/1.1891202>.
34. Bunnoon P, Chalermyanont K, Limsakul C. Wavelet and neural network approach to demand forecasting based on whole and electric sub-control center area. *Int J Soft Comput Eng.* 2012;1:2231–307.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.