IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 27, 2020, accepted January 2, 2021, date of publication January 14, 2021, date of current version January 25, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3051872

Combining Deep Learning and Multiresolution Analysis for Stock Market Forecasting

KHALED A. ALTHELAYA, SALAHADIN A. MOHAMMED, AND EL-SAYED M. EL-ALFY, (Senior Member, IEEE)

Information and Computer Science Department, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia

Corresponding author: Salahadin A. Mohammed (adam@kfupm.edu.sa)

This work was supported by the King Fahd University of Petroleum and Minerals (KFUPM), Saudi Arabia.

ABSTRACT Due to its complexity, financial time-series forecasting is regarded as one of the most challenging problems. During the past two decades, nonlinear modeling techniques, such as artificial neural networks, were commonly employed to solve a variety of time-series problems. Recently, however, deep neural network has been found to be more efficient than those in many application domains. In this article, we propose a model based on deep neural networks that improves the forecasting of stock prices. We investigate the impact of combining deep learning techniques with multiresolution analysis to improve the forecasting accuracy. Our proposed model is based on an empirical wavelet transform which is shown to outperform traditional stationary wavelet transform in capturing price fluctuations at different time scales. The proposed model is demonstrated to be substantially more effective than other models when evaluated on the S&P500 stock index and Mackey-Glass time series.

INDEX TERMS Deep learning, multiresolution analysis, long-short term memory, financial time series, forecasting.

I. INTRODUCTION

A time-series is a sequence of data values taken at equally spaced successive points in time. Time-series analysis has many applications in areas such as speech recognition, electrical signal processing, traffic analysis, weather forecasting, unemployment rate analysis, inflation dynamics, seismic signal processing, economics, business and finance [1]–[3]. A branch of time-series analysis, called financial time-series analysis, is used extensively to predict future stock prices. This enables managers, traders, and investors to lay better plans, make better decisions, minimize risks, reduce costs, save resources, maximize profits, meet objectives, and achieve goals [4], [5]. In the context of this article, financial time-series data is a sequence of stock prices or index values taken at successive equally spaced points in time.

Financial time-series data is frequently used to predict future stock prices. However, predicting future stock prices using financial time-series data is very challenging [6]. This is because the price of a stock is influenced not only by the fundamentals of the company but also by the mood or sentiments of short-term and long-term investors [7]. Positive sentiments can drive prices up and vice versa. Some researchers have adopted techniques from natural language processing to study the impact of sentiments on stock prices [8]. Modeling investors' sentiments is a very challenging problem and predicting future prices based only on fundamental analysis is not sufficient. That is why many investors buy and sell stocks based on technical analysis. Decisions of buying and selling by technical analysts is based on reading charts which they believe help them to predict future stock prices. There are also other investors who base their decisions on several other factors, such as news, probability and statistics, pure fundamentals, and machine learning techniques. Depending on the time frame they use, market investors can be of different categories such as scalpers, day traders, swing traders, or positional traders. The rules to buy and sell shares are also different which adds to the complexity of predicting future stock prices.

Researchers have proposed notable models for financial time-series data using linear modeling techniques such as univariate Autoregressive (AR) and Autoregressive Integrated Moving Average (ARIMA) [9], [10]. However, due to the inherent complexity of financial time-series, design based on linear modeling can result in poor prediction. With the advance in computational power, other nonlinear models have

The associate editor coordinating the review of this manuscript and approving it for publication was Xinyu Du.

been proposed [11]–[13]. Recently, new nonlinear modeling techniques, such as deep learning, have been used for prediction using complex training data. Due to the unprecedented performance achieved by deep learning networks (DNNs), they have attracted considerable attention to build more efficient solutions for a variety of application domains, such as speech recognition, signal processing, image classification and financial time series forecasting. Some researchers proposed using deep recurrent neural network (RNN) along with data preprocessing techniques to reduce and denoise the input data [14], [15]. RNN has the ability to remember its previous states. A variation of RNN, called Long-Short Term Memory (LSTM), has memory units that enables very deep RNNs to remember previous states and use them to perform more accurate forecasting operations.

Data preprocessing is a very important step before performing data modeling. It can increase performance and improve prediction accuracy. Multiresolution Analysis (MRA) is a data preprocessing technique used for time-series data decomposition and denoising. It decomposes time-series data into low-frequency and high-frequency sub-series in the wavelet domain. Each data sub-series can then be used separately to improve forecasting.

In this article we are proposing a more accurate stock prices forecasting model that combines MRA with deep RNN. To demonstrate the performance of the proposed approach, we implemented, compared and evaluated two types of multiresolution analysis methods, namely stationary wavelet decomposition (SWT) and empirical wavelet transforms (EWT). SWT has been previously used in other studies, e.g. [16]. To the best of our knowledge, the proposed approach is the first to combine EWT multiresolution analysis with deep learning for both short- and long-term stock market forecasting. The proposed methodology performs data processing in two phases using several LSTM networks based on data decomposition levels. For our experiments, we used the bench mark datasets, namely S&P500 stock index dataset and Mackey-Glass time series.

The remaining sections of this article are as follows. Relevant literature is reviewed in Section II. The proposed methodology is explained in Section III. Experimental results and analysis of the proposed solution are discussed in Section IV. Conclusions and future works are stated in Section V.

II. LITERATURE REVIEW

Time-series data is used not only to predict future stock prices but also to predict long- and/or short-term trends of weather, wind speed, water demand, electricity consumption and many other application. Time-series data processing techniques developed for one application are frequently used for other applications. For instance, techniques used to predict weather forecasting can also be used to predict wind speed or water demand or electricity consumption with slight modification [1]–[3].

Time-series forecasting involves different linear modeling techniques such as AR, moving-average (MA), ARMA, and ARIMA. Many researchers [17]–[20] used ARMA modeling technique for time-series forecasting. Chen-Xu and Jie-Sheng [18] built a model based on ARMA to predict bank cash flow. Kim [19] discussed the symmetric maximum likelihood (ML) loss function and proposed asymmetric loss function to build ARMA model to forecast stock returns. Chen *et al.* [17] applied an adaptive approach to build ARMA model by deriving the error based on the theory of minimum mean square error (MMSE). Similarly, ARIMA modeling was used by many studies [21]–[23] for time-series forecasting. The designed models were evaluated on different application domains including stock market forecasting. In time-series forecasting, many studies show that non-linear modeling techniques when compared with linear techniques show superior performance and better accuracy [24]–[26]. Thus, many non-linear modeling techniques were employed for time-series forecasting. For example, Santos and dos Santos Coelho [25] investigated the advantages of integrating non-linear Multilayer Perceptron (MLP) neural network with Radial Basis Function Neural Network (RBFNN) and the Takagi-Sugeno fuzzy system to exchange-rate forecasting.

Artificial Neural Networks (ANNs) are widely used for time-series analysis and forecasting due to their ability to model non-linearity in time-series data. Many researchers proposed the use of ANNs in time-series forecasting. Xiao-Ming and Cheng-Zhang [27] combined ten ANNs models together to learn a highly accurate stock price forecasting model. They used AdaBoost technique to train the combined models by selecting several technical indicators of Shanghai Stock Exchange and international stock markets. Another study conducted by Pao and Chih [28] compares the performance of ANN using linear and non-linear models. They compared two nonlinear ANN models and three linear time-series cross-sectional models and showed that the ANN models exhibit higher forecasting accuracy. Guo *et al.* [29] also used ANN to build a hybrid dimensional reduction technique. The model combined two-directional and two-dimensional principal component analysis (2DPCA) with RBFNN to forecast daily stock closing prices. The proposed model input features consisted of many stock market technical variables and used a sliding window to shape the input data. The evaluation used the Shanghai Stock Market index. Some other modeling techniques such as Adaboost and Hidden-Markov Model (HMM) were also employed to time-series forecasting. Hassan and Nath [30] used HMM to predict next day stock price through searching for patterns in the dataset that matches specific query. Huang *et al.* [31] used Support Vector Machine (SVM) to train a model to forecast the direction of the weekly movement of the NIKKEI 225 index. Moreover, Majhi and Anish [32] introduced a non-dominated sorting genetic algorithm version-II (NSGA-II) and multi-objective particle swarm optimization (MOPSO) to efficiently design models for stock price prediction to adjust four performance constrains.

In general, ANN based techniques outperform linear techniques and produce better time-series forecasting models. However, according to Singh and Srivastava [33], the performance of ANN based models is not high enough due to the complexity of financial time-series data. Deep Neural Networks (DNNs), on the other hand, showed superior performance in many applications, such as signal processing, speech recognition, and image classification. Therefore, adopting DNNs techniques to financial time-series forecasting seems a promising approach. As a result, many researchers studied the use of DNN for time-series forecasting. Singh and Srivastava [33] compared Recurrent Neural Network (RNN) and ANN based models and concluded that DNNs models are better than ANNs. Chong *et al.* [13] explored the potential advantages and limitations of integrating DNNs into stock price forecasting by extracting features from high frequency raw data collected from intra-day stock prices. They conducted experiments to study the impact of three unsupervised feature extraction techniques on the networks to predict the market direction. The results showed that DNNs can improve the results of the autoregressive model and enhance its prediction ability. Shen *et al.* [34] proposed a model using an improved Deep Belief Network (DBN) to forecast exchange rates. They used continuous restricted Boltzmann machines (CRBMs) to construct and improve the DBN. Tsantekidis *et al.* [35] applied Convolutional Neural Network (CNN) on a special kind of high frequency data collected by limit-order-book (LOB). The proposed model outperformed both SVM and MLP models. Li *et al.* [36] developed a prediction model using Long Short-Term Memory (LSTM) neural network. The proposed model was improved by integrating a Naïve Bayes (NB) modeling technique to include and extract market factors and investor sentiment from forum posts. Other researchers used different kinds of data representation to train deep learning models. For example, Chen *et al.* [37] transferred time-series data into 2D images and fed them to CNN for training.

Characterizing, modeling, and extracting features of time-series data was better achieved by integrating signal processing methodologies into time-series analysis and decomposition. Time-series data can be represented using different kinds of signal transforms such as Fourier and wavelet transforms. Wavelet transform has been found to outperform Fourier transform in analyzing non-stationary data; thus making it a good candidate for time-series decomposition [38]. Wavelet methodology was first introduced by Grossmann and Morlet [39]. This pioneering work was followed by presenting MRA by Mallat [40]. Combining multiresolution wavelet methodology showed a significant improvement in data analysis and decomposition in many scientific areas including time series. Ismail and Dghais [41], for example, attempted to characterize financial time series using MRA. The study used Linear ARIMA with wavelets to address forecasting results using a multi-resolution fitting approach. Kılıç and Uğur [38] conducted experiments to analyze S&P500 dataset using MRA with some other descriptive statistical modeling. They also conducted experiments to investigate and compare the integration of MRA with linear as well as nonlinear forecasting methods. They concluded that using nonlinear models with MRA can produce better results. Bekiros *et al.* [42] applied MRA to linear and nonlinear models using neural networks and concluded that nonlinear modeling techniques and ANNs result in better performance. Zhang *et al.* [43] found out that the integration of wavelet transforms with ANN produces better forecasting results. They used shift invariant scale-related wavelet transform representation. The transformation established was based on Auto-correlation Shell Representation (ASR). They used two layers of MLPs for prediction and classification. The proposed approach transformed the financial time-series and extracted wavelet coefficients by ASR and applied Bayesian method of automatic relevance determination (ARD) to select best features for the first layer that is composed of multiple MLP predictors. The output of this layer is input to the second layer that consists of one MLP predictor. The proposed model proved to be efficient when compared to another MLP model without wavelet transform. Zhang *et al.* [44] used wavelet neural networks along with other forecasting techniques such as, LSTM, GRU, and MLP to perform prediction of multi-step ahead hydrologic time series acquired by means of an IoT network. The study concluded that LSTM and GRU models can achieve better performance. Reis and Da Silva [45] applied multiresolution analysis to do feature extraction. The study aimed at forecasting electrical short-term load. Multiple forecasting models developed to perform forecasting on different time scales. The multiresolution approach is used to forecast certain scales of the decomposed time series-and builds a separate model for each scale and combines the predicted sub-values to form the final predicted target. Aussem and Murtagh [46] proposed a simple time-series forecasting strategy based on multiresolution analysis. They decomposed time-series data into multiple series based on multiple resolution levels. Multiple prediction models were trained for every resolution scale. The predictions of all subseries were combined to perform final prediction step. A dynamic RNN was used to predict subseries at each resolution. The proposed model was compared to two other forecasting models, namely the MLP network and a simple autoregressive. The dynamic RNN was found to outperform both models.

Few researchers proposed combining wavelet transforms with deep learning techniques to learn financial time-series forecasting model. Persio and Honchar [47] investigated the merits of applying wavelet analysis with deep learning techniques in financial time-series forecasting. They conducted many experiments comparing the forecasting performance of LSTM and CNN. The results show that the combination of CNN and MRA outperforms the compared DNNs. Other types of DNNs such as Stacked Autoencoders (SAEs) were used by Bao *et al.* [15] to develop a forecasting model using LSTM combined with wavelet transform to denoise input features. The combined model outperformed the other three separated single models. Yan and Ouyang [14] combined wavelet analysis with LSTM network to forecast the daily closing prices of the Shanghai Composite Index. The applied methodology is based on using the reconstructed data from the wavelet analysis. They compared the proposed wavelet LSTM network with many other models, trained on the same data, and they found it to be superior.

III. METHODOLOGY

The proposed methodology is a combination of multiresolution analysis for data decomposition and deep learning architectures for data modeling and training. The financial time-series data is decomposed using multi-scale wavelet analysis. A set of deep learning networks is trained for each scale to perform the forecasting per each level. We applied empirical multiresolution wavelet analysis method and compared the results to stationary wavelet analysis. The training process includes two stages. In the first stage, the learning and forecasting of each sub-level of the time-series is done in isolation. In the second stage, forecasts from the previous stage are combined to predict the final target value. The proposed methodology was evaluated by conducting experiments for both short- and long-term forecasting using two benchmark datasets, namely S&P500 dataset and Mackey-glass time series.

A. WAVELET DECOMPOSITION

Data preprocessing is an important step in learning forecasting models. It plays a significant role in determining the most relevant features and models. Multiresolution analysis is a data preprocessing step used to decompose time-series data on different scales to model the data according to several variations of representation. Multiple representations of data are generated depending on the scaling parameters. Using multiple data representations enables more information to be captured and can thus produce better forecasting results. The proposed methodology uses multiresolution analysis for data decomposition. However, finding the best configuration of parameters, that gives the highest possible performance, relies on conducting several experiments.

Two types of wavelet decomposition are explored. During this process, the decomposition of the data is accomplished by defining basis functions called mother wavelet and the data multi-scale resolutions extracted by the projection of the given signal onto the basis function. The main advantage of wavelet decomposition is that it extracts the trend from the data and separates the spurious short fluctuations [48], [49]. The wavelet basis function translation and dilation are parameterized by b and a, respectively [48]. Such a basis function is given by,

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right) \tag{1}$$

Equation 1 conveys a basis for a continuous wavelet transformation. In order to get a multiresolution representation for the data, all the input data are decomposed based on translation and dilation parameters. In contrast, a Discrete Wavelet Transform (DWT) basis function at time location n and dyadic level m is given by [49],

$$\psi_{m,n}(\mathrm{t}) = 2^{\frac{m}{2}}\psi(2^{-m}\cdot t - n) \tag{2}$$

The wavelets of the DWT generated by the dyadic grid sampled wavelets are orthonormal. Based on the basis function in Eq. 2, the inner product between the time-series data denoted by $x(t)$ and the basis function $\psi_{m,n}$ expressed in Eq. 2 is given by,

$$\tau_{m,n} = \int_{-\infty}^{\infty} x(t)\psi_{m,n}(t)dt \tag{3}$$

The wavelet coefficient, $T_{m,n}$, is defined by Eq. 3 and parameterized by dilation m and translation n so that it returns detailed information of the time series. Eq. 4 defines the scaling function based on level and shift parameters denoted by m and n, respectively.

$$\phi_{m,n}(T) = 2^{\frac{m}{2}}\phi(2^{-m}\cdot t - n) \tag{4}$$

where $\phi_{m,n}$ has the property $\int_{-\infty}^{\infty}\phi_{m,n}(t)dt = 1$. For the translations, the scaling function is orthogonal to itself but not to its dilations. The smoothing of the time-series data is produced by the inner product of the time series with the scaling function. The obtained samples are called approximation coefficients and are defined by,

$$s_{m,n} = \int_{-\infty}^{\infty} x(t)\phi_{m,n}(t)dt \tag{5}$$

The following equation is used to obtain a smooth, scaling function-dependent, continuous approximation of the data,

$$x_m(t) = \sum_{n=-\infty}^{\infty} S_{m,n}\phi_{m,n}(t) \tag{6}$$

Knowing the approximation coefficients $S_{m,n}$ generated at level m_0 and chosen arbitrarily, and the wavelet detailed coefficients $T_{m,n}$ at levels $1, 2, .., m_0$, the final multiresolution representation of the data can be obtained as,

$$x(t) = \sum_{n=-\infty}^{\infty} S_{m_0,n}\phi_{m_0,n}(t) + \sum_{m=1}^{m_0}\sum_{n=-\infty}^{\infty} T_{m,n}\psi_{m,n}(t) \tag{7}$$

There are many wavelet families, such as Haar, Daubechies, Symlets, etc. Daubechies wavelets are used by many time-series forecasting applications [43], [45], [46]. The Daubechies wavelets are a family of orthogonal wavelets whose vanishing moments number is maximal for some given support. There are scaling and wavelet functions for each wavelet family of this class. The scaling function generates an orthogonal multiresolution analysis and is called the father wavelet. We use Daubechies-20 wavelet to perform the multiresolution analysis using stationary wavelet transform. The index number 20 refers to the number of coefficients. Figure 1 illustrates the scaling and wavelet functions [50]. DWT may suffer from noise. Alternatively, the stationary wavelet transform is an extension of the typical discrete wavelet transform which is commonly used for exploratory statistical and signal

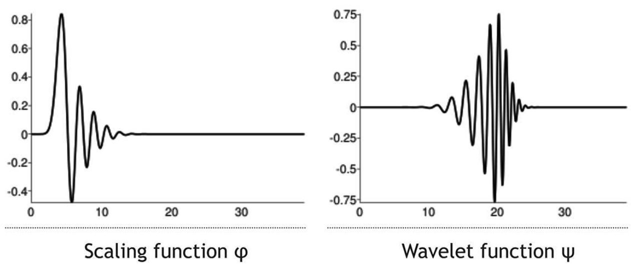

FIGURE 1. Scaling and wavelet functions. [50].

analysis [49]. We noticed that this type of wavelet transformation is commonly used in time-series analysis and decomposition due to its shift-invariant property [43], [46]. The number of resolution levels used for data decomposition in many previous studies is up to four levels [43], [45], [46]. The block diagram of the undecimated wavelet decomposition representation is shown in Figure 2, where D_j and A_j represent the detail coefficients and approximation at level j, respectively. The high frequency components, D_j, are generated from the high-pass filter H_j, and the low frequency components, A_j, are generated from the low-pass filter L_j. Figure 3 shows the data before applying wavelet decomposition and Figure 4 shows the decomposition of the data into four detailed coefficients and approximation coefficients at level four which are used for multiresolution analysis. It can be noticed that the higher the level of decomposition, the smoother the approximation coefficients, and the lower the level of detailed coefficients, the higher the captured frequency.

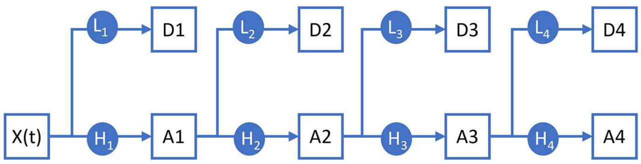

FIGURE 2. Decomposition block diagram of stationary wavelet transform.

Applying different types of wavelet transforms and multiresolution analysis methods aims at investigating, exploring, and deducing better forecasting results based on several potential developed forecasting models. Thus, we apply empirical wavelet transform (EWT) to perform multiresolution analysis of data to build and compare multiple forecasting models. EWT is an adaptive wavelet transform developed by Gilles [51]. The main advantage of this transform is that the wavelet and scaling functions adaptively analyze the data according to the information contained in the time-series without any prior information about the data. It facilitates time-series processing and forecasting by generating higher time-frequency resolution. EWT approach performs data analysis to define a set of adaptive filter banks extracted from data based on its prominent frequency components. It identifies a set of maximum Fourier spectrum ($X(\omega)$) of the signal with a set of corresponding frequency indices (ω_i) by defining frequency and magnitude thresholds. The frequency axis, which is in the range $[0 - \pi]$, is divided into N segments defined as $\Lambda_i = [\omega_{i-1}, \omega_i]$ using boundary values defined by a selected number (N) of maxima [52]. The boundaries ω_i are obtained by setting $\omega_0 = 0$ and $\omega_N = \pi$. The Fourier segments will be $[0, \omega_1], [\omega_1, \omega_2], \ldots [\omega_{N-1}, \pi]$. The filter bank represented by $N-1$ band pass filters and one low pass filter is constructed depending on boundaries. The supports (τ_n) for the filters can be calculated using the following equation [52],

$$\tau_n = 2\gamma\omega_n \tag{8}$$

where γ parameter is defined to ensure that no two consecutive transition bands are overlapping. The value of γ can be computed using the following equation,

$$\gamma \leq \min_i \left[\frac{\omega_{i+1} - \omega_i}{\omega_{i+1} + \omega_i}\right] \tag{9}$$

The wavelet functions are defined using the β function which is calculated as follows,

$$\beta(\gamma, \omega_i) = \beta\left(\frac{1}{2\gamma\omega_i}(|\omega| - (1-\gamma)\omega_i)\right) \tag{10}$$

where $\beta(x)$ is defined by,

$$\beta(x) = x^4(35 - 84x + 70x^2 - 20x^3) \tag{11}$$

Many functions can be defined by satisfying the following conditions,

$$\beta(x) = \begin{cases} 0, & if\ x\ \leq 0 \\ 1, & if\ x\ \geq 0 \\ \beta(x) + \beta(1-x) = 1 & if\ x\ \in [0, 1] \end{cases} \tag{12}$$

The empirical scaling function $\varphi_1(\omega)$ and the wavelets functions $\psi_i(\omega)$ are given by,

$$\varphi_1(\omega) = \begin{cases} 1, & \text{if } |\omega|(1-\gamma)\omega_1 \\ \cos\left[\frac{\pi}{2}\beta(\gamma, \omega_1)\right], & \text{if } (1-\gamma)\omega_1 \leq |\omega| \\ & \leq (1+\gamma)\omega_1 \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

$$\psi_i(\omega) = \begin{cases} 1, & \text{if } (1+\gamma)\omega_{\mathrm{i}} \leq |\omega| \\ & \leq (1-\gamma)\omega_{\mathrm{i+1}} \\ \cos\left[\frac{\pi}{2}\beta(\gamma, \omega_{\mathrm{i+1}})\right], & \text{if } (1-\gamma)\omega_{\mathrm{i+1}} \leq |\omega| \\ & \leq (1+\gamma)\omega_{\mathrm{i+1}} \\ \sin\left[\frac{\pi}{2}\beta(\gamma, \omega_{\mathrm{i}})\right], & \text{if } (1-\gamma)\omega_{\mathrm{i}} \leq |\omega| \\ & \leq (1+\gamma)\omega_{\mathrm{i}} \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

Based on the defined set of band filters, the EWT can be defined in a similar way as the normal wavelet transform. The approximation coefficients are obtained by the inner product of applied signal $X(\omega)$ with the empirical scaling function as given by [52],

$$W_x(1,\ t) = \langle x, \varphi_1 \rangle = IFFT(X(\omega) * \varphi_1(\omega)) \tag{15}$$

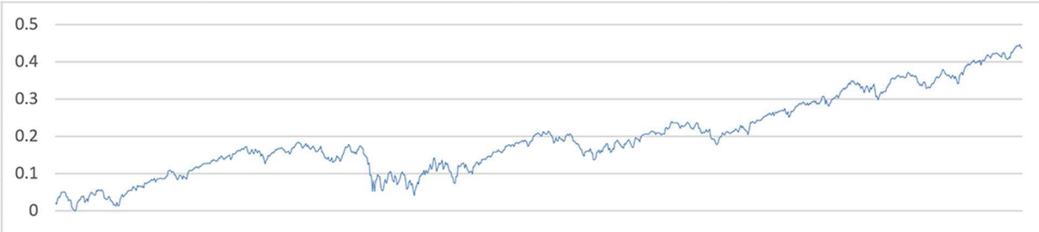

FIGURE 3. Curve of part of the actual S&P500 data before performing wavelet analysis.

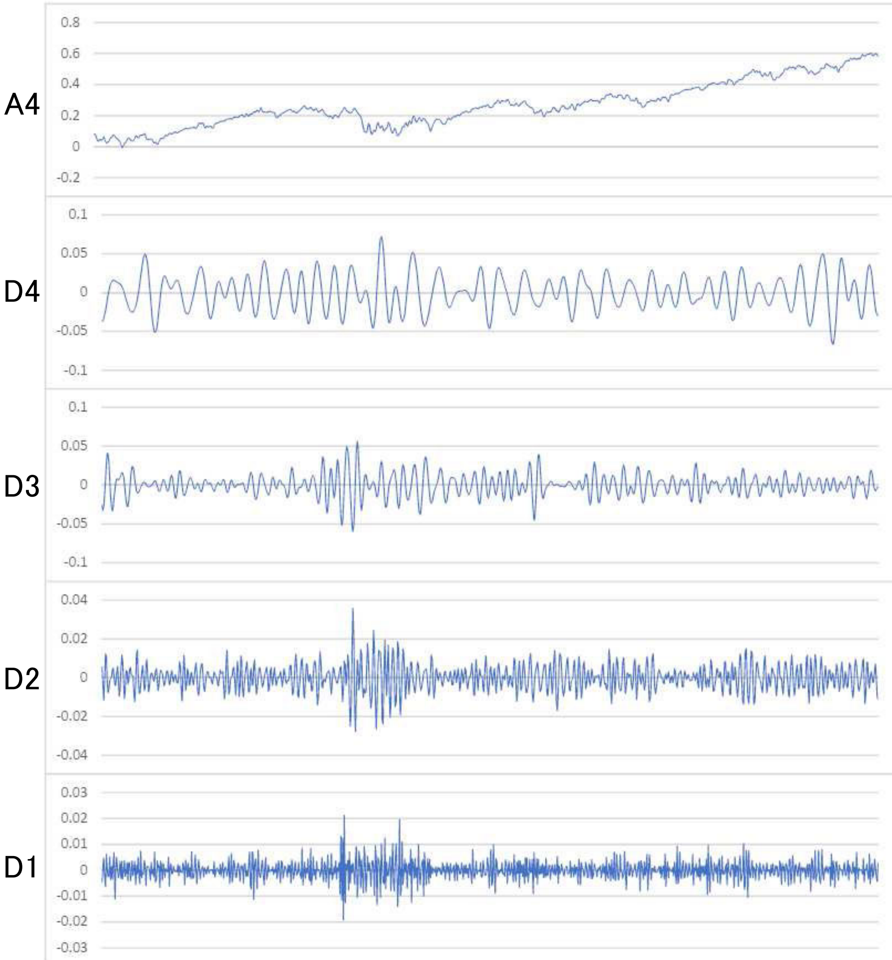

FIGURE 4. Stationary wavelet decomposition for part of the data.

The inner product of the empirical wavelets with applied signal $X(\omega)$ produces the detailed coefficients as given by,

$$W_x(i, t) = \langle x, \psi_i \rangle = IFFT(X(\omega) * \psi_i(\omega)) \quad (16)$$

The decomposition of the input signal to Intrinsic Mode Functions (IMF) is performed using equations 15 and 16. This decomposition approach is characterized by using basis functions generated according to the information contained in the signal [52]. Figure 3 shows the data before applying wavelet decomposition and Figure 5 shows part of the decomposition of S&P500 data into 18 resolution levels using empirical wavelet transform which are used for multiresolution analysis.

B. DEEP NETWORK ARCHITECTURE

Deep learning networks are built using several stacked hidden layers to train on huge amount of data. Each intermediate layer extracts certain kind of patterns from the data and redirects the learned patterns to the following layers to perform other types of pattern extraction. A hierarchy of patterns is

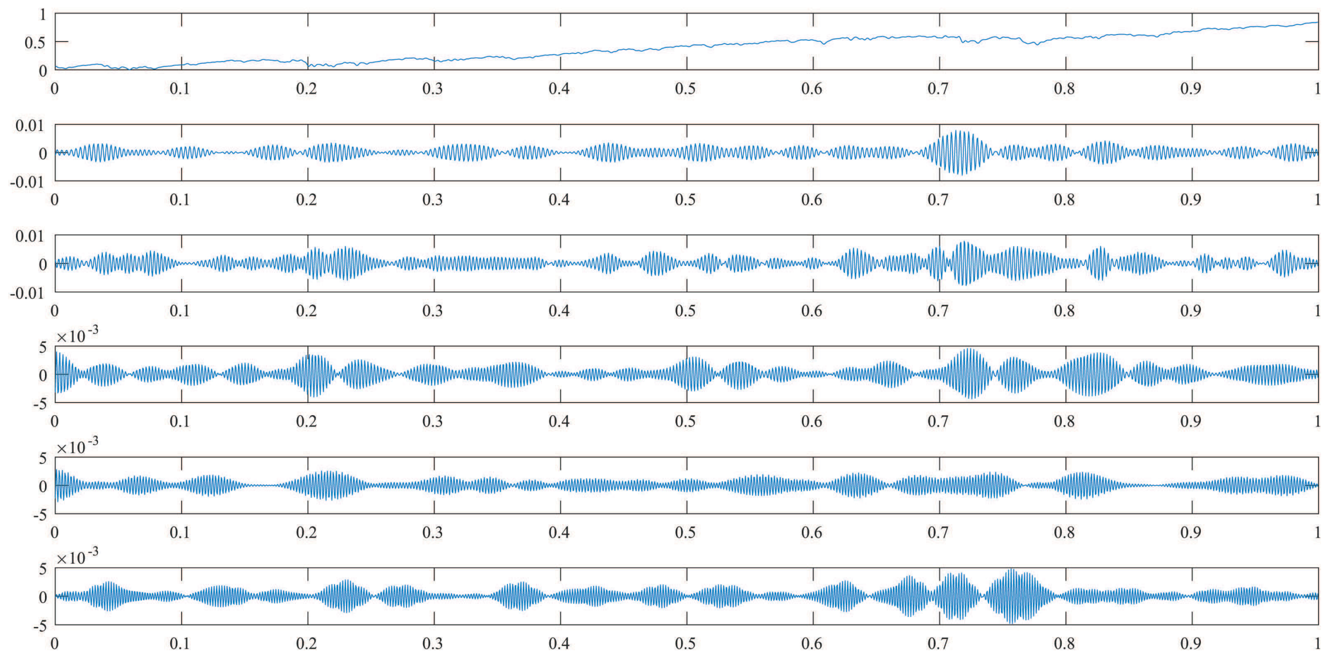

FIGURE 5. Adaptive wavelet decomposition for part of the data.

extracted from the training data and employed to perform forecasting. The decomposition of data into sub-patterns simplifies learning more complicated hierarchies. The input features are fed into the deep neural network to learn network weights. Selecting the best network design topology depends on the performance results of the conducted experiments.

Many deep learning solutions have been developed to process and model patterns from data sequences. Deep RNN is a commonly used deep learning technique to process sequential data. It is widely employed to map input sequences to target values or output sequences. It has been used in image captioning, sentiment classification, and language translation. It has also been used to develop various forecasting models to different kinds of time-series data, such as power consumption, traffic analysis, and wind speed as well as financial time series. It is characterized by its ability to preserve network preceding states and retain information in its hidden network cells while cascading forward through data sequences. It uses a training algorithm mainly based on backpropagation through time (BPTT). Training operations align with the order and sequence of time series by linking time steps sequentially. Capturing long-term dependencies is infeasible using RNN due to the vanishing gradient problem. This problem is solved by enhancing the hidden states of the RNN to remember longer sequences of data. The enhanced version of RNN is called LSTM, which is devised by Hochreiter and Schmidhuber. It adds input, forget, and output gates that control information to or out of the memory cells using pointwise multiplication and sigmoid neural net layer. The gated cells act on the current input data by allowing or blocking its information to pass based on its importance to the target value. Unlike RNN, LSTM uses truncated BPTT (TBPTT) that allows faster and more stable results. The LSTM transition equations are as follows:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + V_i c_{t-1}) \tag{17}$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + V_f c_{t-1}) \tag{18}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + V_o c_t) \tag{19}$$

$$\tilde{c_t} = \tanh(W_c x_t + U_c h_{t-1}) \tag{20}$$

$$c_t = f_t^i \odot c_{t-1} + i_t \odot \tilde{c_t} \tag{21}$$

$$h_t = o_t \odot \tanh(c_t) \tag{22}$$

where i_t denotes the input gate and o_t denotes the output gate, f_t, c_t, and h_t denote the forget gate, memory cell, and hidden state, respectively [53].

C. TRAINING THE NETWORK

The methodology followed to construct the network architecture is illustrated in Figure 6. The proposed architecture consists of three stages. The first stage performs data analysis and decomposes the data into many sub-series. The second stage learns multiple intermediate networks based on the number of resolution levels generated from the previous stage. Each intermediate network is constructed using two layers of LSTM and one dense layer to learn each sub-series. To choose the best network configuration, we varied the number of memory units of the first LSTM layer between 8, 16, and 32. The number of memory units in the second layer of the LSTM was half of the first layer. The forecast of each resolution level is produced by a dense layer which uses a linear activation function. The third stage of the proposed architecture receives forecasts produced by the second stage as input features. It is constructed using two layers of LSTM

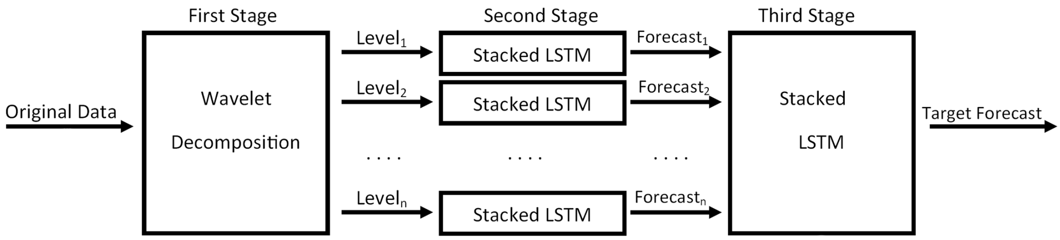

FIGURE 6. **Outline of the proposed methodology.**

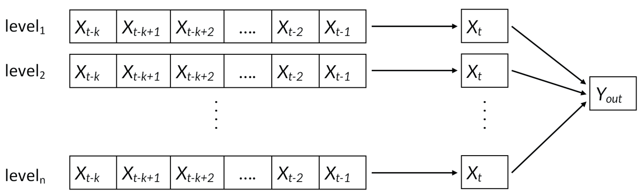

FIGURE 7. **Sliding window and features reshaping to perform forecasting.**

and one dense layer to produce the final output using a linear activation function. The LSTM output layer uses tanh transfer function, while the hard sigmoid transfer function is used for the recurrent activation. Input features are formed according to time lag methodology adopted by this work. We use a sliding window of size k constructed from t_1 to t_n time steps to forecast the t_{n+1} time step in the future. Each data sample formed by sliding window consists of k time steps denoted as $(x_{t-k}, x_{t-k+1}, \ldots, x_{t-2}, x_{t-1})$ to forecast x_t in the future. We utilize k time steps of each resolution level to perform the forecast as illustrated in Figure 7. The third stage of the proposed architecture combines the forecasts of each resolution level to produce the final forecast value. The values of the time steps used are five and ten working days, i.e. working days of one and two weeks, respectively.

IV. EXPERIMENTS

To evaluate the proposed approach, we conducted several experiments using two time series datasets (Mackey-Glass and S&P500 datasets). We implemented the proposed forecasting approach in Python using Keras open-source package for deep learning with TensorFlow backend. The stationary wavelet analysis was performed using PyWavelets open-source library [50]. The empirical adaptive wavelet analysis was accomplished using the MATLAB toolbox for empirical wavelet transforms [51]. Data was normalized using the mix-max normalization (Eq. 23) and framed into five and ten days sliding windows with overlapping.

$$norm(x) = \frac{x - min_i}{max_i - min_i}.(nmax_i - nmin_i) + nmin_i \quad (23)$$

where min_i and max_i denote the original interval, $nmin_i$ and $nmax_i$ represent the new interval, x represents the value, $norm(x)$ denotes the normalized value. The applications of EWT converts the time series into many intrinsic mode functions (IMFs) and all returned functions were used in the construction of the forecasting model. The evaluation process used four performance measures to assess how well the forecasts represent the actual data. These measures are the Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and coefficient of determination denoted as R^2. The lower the error measures (MAE, RMSE and MAPE) are and the closer the value of R^2 to one is, the better the forecasting performance is. These measures are defined mathematically as follows:

$$MAE = \frac{1}{n}\sum_{t=0}^{n-1} |y_t - \hat{y}_t| \quad (24)$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=0}^{n-1} (y_t - \hat{y}_t)^2} \quad (25)$$

$$MAPE = \frac{1}{n}\sum_{t=0}^{n-1} \frac{|y_t - \hat{y}_t|}{y_t + \epsilon} \quad (26)$$

$$R^2 = 1 - \frac{\sum_{t=0}^{n-1} (y_t - \hat{y}_t)^2}{\sum_{t=0}^{n-1} (y_t - \bar{y})^2} \quad (27)$$

where y_t and $\hat{y}_t$ represent the actual and predicted values at step t for $0 \leq t < n$, respectively, $\bar{y} = \sum_{t=0}^{n-1} y_t/n$, and ϵ is a infinitesimal value to avoid dividing by zero. Several experiments were carried out using stationary and empirical wavelet transforms for the data analysis and decomposition phase. The proposed network architecture was tuned using different numbers of cells (8, 16, and 32) for both intermediate and final stage deep networks. Each resolution level data was reshaped into supervised learning and fed into the intermediate deep network. The training process was performed using training data and the best network weights were determined based on the results generated from the validation data. The number of epochs used to train all developed networks was 800.

First, the proposed approach was evaluated and compared to other methods using Mackey-Glass (MG) chaotic time series [54], which was first introduced to model the production of white blood cells. It is defined by the following time-delay ordinary differential equation,

$$\frac{dx(t)}{dt} = \beta \frac{x(t-\tau)}{1 + x^{10}(t-\tau)} - \gamma x(t) \quad (28)$$

TABLE 1. Results of short-term forecasting using deep learning with empirical wavelet transform (EWT) or stationary wavelet transform (SWT) for Mackey-Glass time series normalized and original test data.

Window	Network		SWT					EWT							
			MAE		$RMSE$		R^2	MAE			$RMSE$			R^2	
	1st stage	2nd stage	Norm.	Origin.	Norm.	Origin.	Value	Norm	Origin	Imp(%)	Norm	Origin	Imp(%)	Value	Imp(%)
5	8	8	0.11434	0.10015	0.15808	0.13846	0.67061	0.00255	0.00223	97.77	0.00326	0.00286	97.94	0.99987	49.10
		16	0.09990	0.08750	0.15038	0.13171	0.74139	0.00252	0.00221	97.48	0.00322	0.00282	97.86	0.99987	34.86
		32	0.06293	0.05512	0.10755	0.09421	0.87956	0.00252	0.00221	96.00	0.00322	0.00282	97.01	0.99987	13.68
	16	8	0.11190	0.09801	0.15973	0.13990	0.69960	0.00256	0.00224	97.71	0.00331	0.00290	97.93	0.99986	42.92
		16	0.08177	0.07162	0.12885	0.11286	0.81233	0.00255	0.00223	96.89	0.00327	0.00286	97.46	0.99987	23.09
		32	0.07829	0.06857	0.12525	0.10971	0.82800	0.00252	0.00221	96.78	0.00322	0.00282	97.43	0.99987	20.76
	32	8	0.14229	0.12463	0.19145	0.16769	0.50372	0.00255	0.00223	98.21	0.00327	0.00287	98.29	0.99987	98.50
		16	0.08492	0.07438	0.13353	0.11695	0.79384	0.00253	0.00222	97.02	0.00324	0.00284	97.57	0.99987	25.95
		32	0.06203	0.05433	0.12037	0.10543	0.89556	0.00253	0.00222	95.92	0.00323	0.00283	97.32	0.99987	11.65
10	8	8	0.11973	0.10487	0.16464	0.14421	0.63091	0.00287	0.00252	97.60	0.00378	0.00331	97.70	0.99984	58.48
		16	0.07883	0.06905	0.12676	0.11103	0.82431	0.00284	0.00249	96.40	0.00366	0.00320	97.12	0.99985	21.30
		32	0.07797	0.06830	0.12619	0.11053	0.82679	0.00285	0.00250	96.34	0.00364	0.00319	97.11	0.99985	20.93
	16	8	0.11374	0.09963	0.16935	0.14833	0.68524	0.00258	0.00226	97.73	0.00340	0.00298	97.99	0.99986	45.91
		16	0.08951	0.07840	0.13492	0.11817	0.78426	0.00257	0.00225	97.13	0.00333	0.00292	97.53	0.99986	27.49
		32	0.06875	0.06022	0.11509	0.10081	0.86713	0.00253	0.00221	96.32	0.00323	0.00283	97.19	0.99987	15.31
	32	8	0.12701	0.11125	0.17897	0.15675	0.61429	0.00263	0.00230	97.93	0.00345	0.00302	98.07	0.99986	62.77
		16	0.08898	0.07793	0.13534	0.11854	0.79356	0.00256	0.00225	97.12	0.00330	0.00289	97.56	0.99987	26.00
		32	0.06592	0.05774	0.11319	0.09914	0.86528	0.00259	0.00227	96.07	0.00333	0.00291	97.06	0.99987	15.55

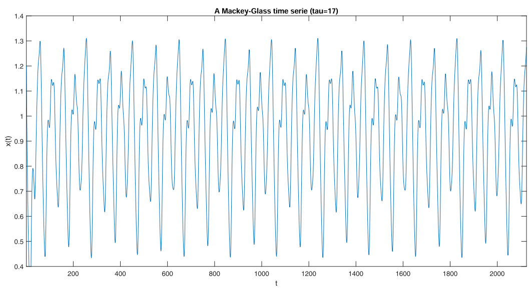

FIGURE 8. Plot of data generated by the Mackey-Glass time series.

where τ is a delay parameter which must be greater than 16.8 for generating a chaotic time series. The parameters selected to generate this time series for our experiment were $\beta = 0.2$, $\gamma = 0.1$, $\tau = 17$, and $x(0) = 1.2$. The adopted time series was composed of the last 2000 of 2123 samples generated from Eq. 28. Figure 8 shows a plot of this MG time series, where the first 75% was used for training and validation, and the remaining 25% for testing. The short-term forecasting results of the proposed approach are as shown in Table 1 for both EWT and SWT analysis methods. The results are grouped based on different number of neurons for the intermediate and final layers and on different window sizes (5 or 10 time steps). They are also grouped based on normalized (Norm) and original (Origin) data and on the multiresolution analysis method EWT or SWT. Figure 9 illustrates the difference between actual data and forecasts of the two best short-term forecasting models produced using EWT and SWT based approaches.

The following experimental work was then conducted for a real dataset. The S&P500 dataset was used to evaluate the proposed approach. We downloaded and used the historical data for the period from 01/01/2010 to 20/02/2018 from Yahoo finance. This dataset is one of the important benchmark datasets and contains the stock prices of 500 companies in the US market. Figure 3 shows the curve of the data used to evaluate the model forecasting of future closing prices. The decomposed data generated from the multiresolution analysis phase were reshaped using sliding windows of size five and ten working days, respectively. The data was split into two parts: the first 80% for training and the remaining 20% for testing. A number of experiments were conducted on this dataset for short-term and long-term forecasting. The short-term forecasting aims at predicting the closing price of the next day whereas the long-term forecasting aims at predicting the closing price 30 days ahead.

The short-term forecasting results generated from the proposed architecture for both EWT and SWT are shown in Table 2. Considering all trained network architectures using S&P500 dataset for short-term forecasting, the best performance (based on minimum RMSE) was achieved using EWT analysis method. As shown in Table 2, the best achieved performance reaches RMSE of 10.089 (above 97% improvement), MAE of 7.319, MAPE of 0.305, and R^2 of 1.0. These results were produced using ten time steps input features to LSTM network architecture designed of 32 neurons in the intermediate network and 8 in the final network. For the SWT analysis method, the best performance for short-term forecasting was achieved using ten time steps input features to a network architecture composed of 16 neurons in both intermediate and final LSTM networks. We can see in Table 2 that the best SWT results are 342.018 for RMSE, 174.695 for MAE, 7.058 for MAPE and 0.981 for R^2. Figure 10 illustrates the difference between the actual data and forecasts of the two best short-term forecasting models produced using EWT and SWT analysis approaches.

For the long-term forecasting using S&P500 dataset, experiments were conducted to predict the closing price

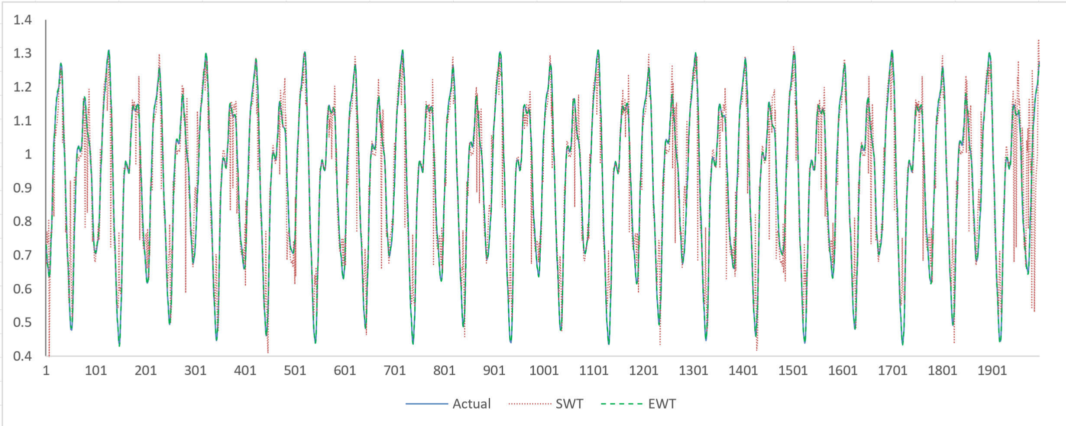

FIGURE 9. Mackey-Glass data curves of actual prices versus forecasts using best selected models of SWT and EWT analysis approaches.

TABLE 2. Results of short-term forecasting using deep learning with empirical wavelet transform (EWT) or stationary wavelet transform (SWT) for S&P500 normalized and original test data.

Window	Model		$RMSE$			MAE			$MAPE$			R^2		
	1st stage	2nd stage	SWT	EWT	Imp(%)	SWT	EWT	Imp(%)	SWT	EWT	Imp(%)	SWT	EWT	Imp(%)
5	8	8	456.703	13.334	97.080	267.291	9.609	96.405	10.791	0.394	96.350	0.981	1.000	1.928
		16	426.633	17.523	95.893	236.141	12.517	94.699	9.527	0.506	94.688	0.976	1.000	2.456
		32	435.366	18.537	95.742	250.916	12.813	94.893	10.117	0.515	94.907	0.968	1.000	3.301
	16	8	456.061	11.205	97.543	276.999	8.288	97.008	11.208	0.348	96.891	0.979	1.000	2.127
		16	432.719	12.023	97.222	256.497	8.860	96.546	10.345	0.370	96.423	0.973	1.000	2.759
		32	418.183	14.889	96.440	245.544	10.553	95.702	9.924	0.433	95.638	0.962	1.000	3.944
	32	8	434.323	13.501	96.892	263.248	10.505	96.010	10.684	0.442	95.867	0.971	0.999	2.985
		16	422.749	18.055	95.729	232.361	13.676	94.114	9.365	0.562	93.995	0.976	0.999	2.439
		32	440.815	21.172	95.197	267.856	15.667	94.151	10.860	0.638	94.128	0.969	1.000	3.127
10	8	8	373.177	13.892	96.277	203.866	9.954	95.117	8.249	0.406	95.079	0.981	1.000	1.863
		16	348.361	15.565	95.532	174.695	11.039	93.681	7.058	0.447	93.663	0.978	1.000	2.183
		32	349.418	18.784	94.624	181.577	12.964	92.860	7.332	0.520	92.910	0.975	1.000	2.563
	16	8	358.026	11.427	96.808	177.828	8.323	95.320	7.202	0.343	95.237	0.979	1.000	2.150
		16	359.632	14.652	95.926	203.648	10.572	94.808	8.246	0.429	94.792	0.977	1.000	2.318
		32	348.006	22.200	93.621	190.864	15.611	91.821	7.717	0.623	91.923	0.975	1.000	2.498
	32	8	372.524	10.087	97.292	208.612	7.319	96.492	8.433	0.305	96.384	0.981	1.000	1.957
		16	342.996	10.297	96.998	181.917	7.453	95.903	7.369	0.310	95.795	0.979	1.000	2.152
		32	342.018	16.598	95.147	185.445	11.212	93.954	7.512	0.451	93.999	0.975	1.000	2.526

30-days ahead. We reserved the last 30% of the whole data for testing and the remaining 70% was used for training and validation. The results of both stationary and empirical wavelet transforms are shown in Table 3. These results show that the highest performance of the EWT analysis method is achieved by the model using 10 time steps window size as input features and consisting of 32 neurons in both intermediate and final LSTM networks. The best achieved performance (minimum RMSE) has RMSE = 169.951 (approx. 73% improvement), MAE = 134.16, MAPE = 5.481, and R^2 = 0.977. On the other hand, the best results of the SWT analysis method were achieved using 10 time steps features as input to a network architecture consisting of 16 neurons in both intermediate and final LSTM networks. This produced RMSE = 550.97, MAE = 356.71, and R^2 = 0.976.

After comparing the results of the two approaches, we can notice that significant forecasting improvement was achieved when using the empirical wavelet transform for data decomposition. The results produced from the evaluation experiments conducted using Mackey-Glass and S&P500 datasets demonstrate that using empirical wavelet analysis has significantly enhanced the performance compared to stationary wavelet analysis. EWT segments the Fourier spectrum of the data and constructs adaptive filter bank to perform data decomposition and separate each mode. Consequently, it helps finding sparse representations of the data based on information included in the dataset. On the contrary, SWT uses basis functions designed independently of the data representation which may result in a weak multiresolution representation of the data. The wavelet coefficients include very large redundancy which increases the computational cost. It also lacks directionality and has persistent oscillation. The optimal number of resolution levels of SWT is not easy to determine. On the other hand, EWT has the ability to design

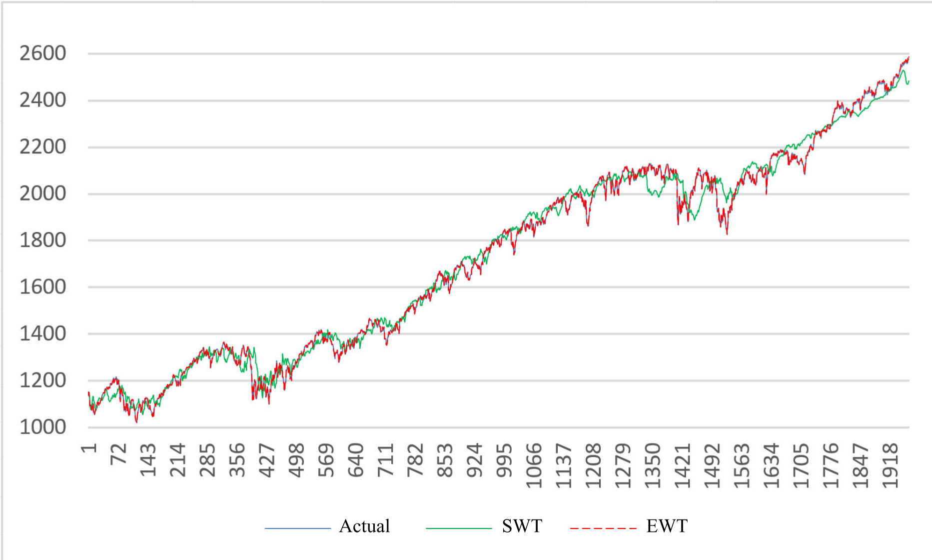

FIGURE 10. S&P500 data curves of actual prices versus forecasts using best selected models of SWT and EWT analysis approaches (Actual and EWT curves mostly overlap).

TABLE 3. Results of long-term forecasting using deep learning with empirical wavelet transform (EWT) or stationary wavelet transform (SWT) for S&P500 normalized and original test data.

Window	Model		$RMSE$			MAE			$MAPE$			R^2		
	1st stage	2nd stage	SWT	EWT	Imp(%)	SWT	EWT	Imp(%)	SWT	EWT	Imp(%)	SWT	EWT	Imp(%)
5	8	8	716.905	208.624	70.899	471.053	160.762	65.872	19.111	6.512	65.926	0.970	0.974	0.385
		16	709.073	208.873	70.543	457.396	161.545	64.681	18.493	6.546	64.602	0.975	0.974	-0.045
		32	688.927	206.094	70.085	449.097	159.438	64.498	18.185	6.464	64.455	0.973	0.974	0.143
	16	8	686.966	189.364	72.435	457.108	148.137	67.593	18.574	6.028	67.549	0.974	0.975	0.090
		16	686.996	183.814	73.244	443.575	144.084	67.518	17.943	5.870	67.284	0.961	0.974	1.378
		32	695.538	181.342	73.928	465.966	142.232	69.476	18.953	5.798	69.408	0.964	0.974	1.098
	32	8	597.034	192.102	67.824	380.526	149.331	60.757	15.379	6.063	60.574	0.977	0.976	-0.085
		16	594.064	185.746	68.733	373.305	144.776	61.218	15.069	5.887	60.934	0.977	0.976	-0.155
		32	600.134	179.498	70.090	379.521	139.905	63.136	15.333	5.696	62.851	0.979	0.975	-0.318
10	8	8	710.582	186.033	73.820	481.636	146.519	69.579	19.624	5.971	69.574	0.970	0.976	0.548
		16	698.631	176.836	74.688	466.650	139.965	70.006	18.963	5.718	69.848	0.976	0.975	-0.096
		32	667.747	171.712	74.285	442.866	136.597	69.156	17.983	5.590	68.915	0.978	0.975	-0.275
	16	8	569.106	176.192	69.041	364.760	138.707	61.973	14.765	5.660	61.667	0.975	0.976	0.180
		16	550.968	175.009	68.236	356.706	138.553	61.158	14.459	5.659	60.863	0.976	0.976	-0.005
		32	469.655	170.531	63.690	316.080	135.633	57.089	12.888	5.548	56.953	0.944	0.976	3.477
	32	8	642.696	182.853	71.549	419.548	143.881	65.706	16.992	5.860	65.514	0.969	0.977	0.921
		16	635.656	171.887	72.959	411.050	135.893	66.940	16.631	5.550	66.628	0.977	0.977	0.076
		32	632.893	169.951	73.147	417.916	134.160	67.898	16.953	5.481	67.670	0.962	0.977	1.508

basis wavelet functions constructed based on the information in the data. The number of resolution levels depends on the Fourier spectrum.

V. CONCLUSION

This article proposed a novel approach for financial time-series forecasting using multiresolution analysis and deep recurrent learning techniques. We implemented, evaluated and compared two multiresolution analysis methods using stationary as well as empirical wavelet transforms. Applying deep learning for financial time-series forecasting produced higher accuracy for both short-term and long-term forecasting. In addition, combining multiresolution analysis and data decomposition further enhances the forecasting accuracy. The proposed learning approach uses two stages of stacked LSTM. The first stage performs learning for each resolution level whereas the second stage produces the final forecasts based on the components of each resolution level. The proposed approach was evaluated using two different benchmark datasets. It was found that using the empirical wavelet transform for data multiresolution decomposition outperformed stationary wavelet transform. Being adaptive methodology, EWT helps modeling techniques to extract more representative patterns from the data to construct more effective forecasting models. Though the proposed methodology has been evaluated on stock market index data, it did not consider the behavior of individual stocks. As future work, it is recommended to combine other trading data sources and technical analysis of stock markets.

ACKNOWLEDGMENT

The authors would like to thank King Fahd University of Petroleum and Minerals (KFUPM), Saudi Arabia, for support during this work.

REFERENCES

[1] A. Altunkaynak and T. A. Nigussie, "Monthly water consumption prediction using season algorithm and wavelet transform–based models," *J. Water Resour. Planning Manage.*, vol. 143, no. 6, Jun. 2017, Art. no. 04017011.

[2] P. Liang, H.-D. Yang, W.-S. Chen, S.-Y. Xiao, and Z.-Z. Lan, "Transfer learning for aluminium extrusion electricity consumption anomaly detection via deep neural networks," *Int. J. Comput. Integr. Manuf.*, vol. 31, nos. 4–5, pp. 396–405, Apr. 2018.

[3] H. Z. Wang, G. B. Wang, G. Q. Li, J. C. Peng, and Y. T. Liu, "Deep belief network based deterministic and probabilistic wind speed forecasting approach," *Appl. Energy*, vol. 182, pp. 80–93, Nov. 2016.

[4] X. Qiu, H. Zhu, P. Suganthan, and G. A. Amaratunga, "Stock price forecasting with empirical mode decomposition based ensemble ν-support vector regression model," in *Proc. Int. Conf. Comput. Intell., Commun., Bus. Anal.* Singapore: Springer, 2017, pp. 22–34.

[5] D. G. Gloubos, "Estimating corporate failure as an auditor's going concern evaluation factor," M.S. thesis, School Bus. Admin., Univ. Macedonia, Thessaloniki, Greece, 2016.

[6] K. A. Althelaya, E.-S.-M. El-Alfy, and S. Mohammed, "Evaluation of bidirectional LSTM for short-and long-term stock market prediction," in *Proc. 9th Int. Conf. Inf. Commun. Syst. (ICICS)*, Apr. 2018, pp. 151–156.

[7] E. Cambria, "Affective computing and sentiment analysis," *IEEE Intell. Syst.*, vol. 31, no. 2, pp. 102–107, Mar. 2016.

[8] J. Griffith, M. Najand, and J. Shen, "Emotions in the stock market," *J. Behav. Finance*, vol. 21, no. 1, pp. 42–56, 2020.

[9] M. Khashei, M. Bijari, and G. A. Raissi Ardali, "Improvement of auto-regressive integrated moving average models using fuzzy logic and artificial neural networks (ANNs)," *Neurocomputing*, vol. 72, nos. 4–6, pp. 956–967, Jan. 2009.

[10] M. Al Wadia and M. Tahir Ismail, "Selecting wavelet transforms model in forecasting financial time series data based on Arima model," *Appl. Math. Sci.*, vol. 5, no. 7, pp. 315–326, 2011.

[11] M. Bildirici and Ö. Ö. Ersin, "Nonlinearity, volatility and fractional integration in daily oil prices: Smooth transition autoregressive ST-FI (ap) GARCH models," *Romanian J. Econ. Forecasting*, vol. 3, pp. 108–135, 2014.

[12] A. Lendasse, E. de Bodt, V. Wertz, and M. Verleysen, "Non-linear financial time series forecasting–Application to the bel 20 stock market index," *Eur. J. Econ. Social Syst.*, vol. 14, no. 1, pp. 81–91, 2000.

[13] E. Chong, C. Han, and F. C. Park, "Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies," *Expert Syst. Appl.*, vol. 83, pp. 187–205, Oct. 2017.

[14] H. Yan and H. Ouyang, "Financial time series prediction based on deep learning," *Wireless Pers. Commun.*, vol. 102, no. 2, pp. 683–700, Sep. 2018.

[15] W. Bao, J. Yue, and Y. Rao, "A deep learning framework for financial time series using stacked autoencoders and long-short term memory," *PLoS ONE*, vol. 12, no. 7, Jul. 2017, Art. no. e0180944.

[16] K. Nagarathinam and R. S. Kathavarayan, "Moving shadow detection based on stationary wavelet transform," *EURASIP J. Image Video Process.*, vol. 2017, no. 1, p. 49, Dec. 2017.

[17] J.-F. Chen, W.-M. Wang, and C.-M. Huang, "Analysis of an adaptive time-series autoregressive moving-average (ARMA) model for short-term load forecasting," *Electr. Power Syst. Res.*, vol. 34, no. 3, pp. 187–196, Sep. 1995.

[18] N. Chen-xu and W. Jie-sheng, "Auto regressive moving average (ARMA) prediction method of bank cash flow time series," in *Proc. 34th Chin. Control Conf. (CCC)*, Jul. 2015, pp. 4928–4933.

[19] M. Kim, "Cost-sensitive estimation of ARMA models for financial asset return data," *Math. Problems Eng.*, vol. 2015, pp. 1–8, Jan. 2015.

[20] D. Marcek, "Forecasting high frequency data: An ARMA-soft RBF network model for time series," *Appl. Mech. Mater.*, vol. 596, pp. 160–163, Jul. 2014.

[21] D. Banerjee, "Forecasting of indian stock market using time-series ARIMA model," in *Proc. 2nd Int. Conf. Bus. Inf. Manage. (ICBIM)*, Jan. 2014, pp. 131–135.

[22] P. Newbold, "ARIMA model building and the time series analysis approach to forecasting," *J. Forecasting*, vol. 2, no. 1, pp. 23–35, Jan. 1983.

[23] X. Wang and Y. Liu, "ARIMA time series application to employment forecasting," in *Proc. 4th Int. Conf. Comput. Sci. Edu.*, Jul. 2009, pp. 1124–1127.

[24] D. Fay, J. Ringwood, M. Condon, and M. Kelly, "A comparison of linear and neural parallel time series models for short-term load forecasting in the republic of Ireland," in *Proc. 3rd Eur. IFS Workshop*. Herzogenrath, Germany: Shaker Verlag, 2000, pp. 1–5.

[25] A. A. P. Santos and L. dos Santos Coelho, "Neural networks, fuzzy system, and linear models in forecasting exchange rates: Comparison and case studies," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Jul. 2006, pp. 3094–3099.

[26] S. K. Sharma and S. Ghosh, "Short-term wind speed forecasting: Application of linear and non-linear time series models," *Int. J. Green Energy*, vol. 13, no. 14, pp. 1490–1500, Nov. 2016.

[27] X.-M. Bai and C.-Z. Wang, "AdaBoost artificial neural network for stock market predicting," in *Proc. Joint Int. Conf. Artif. Intell. Comput. Eng. (AICE), Int. Conf. Netw. Commun. Secur. (NCS)*, 2016.

[28] H.-T. Pao and Y.-Y. Chih, "Comparison of linear and nonlinear models for panel data forecasting: Debt policy in taiwan," *Rev. Pacific Basin Financial Markets Policies*, vol. 8, no. 3, pp. 525–541, Sep. 2005.

[29] Z. Guo, H. Wang, J. Yang, and D. J. Miller, "A stock market forecasting model combining two-directional two-dimensional principal component analysis and radial basis function neural network," *PLoS ONE*, vol. 10, no. 4, Apr. 2015, Art. no. e0122385.

[30] M. R. Hassan and B. Nath, "Stock market forecasting using hidden Markov model: A new approach," in *Proc. 5th Int. Conf. Intell. Syst. Design Appl. (ISDA)*, 2005, pp. 192–196.

[31] W. Huang, Y. Nakamori, and S.-Y. Wang, "Forecasting stock market movement direction with support vector machine," *Comput. Oper. Res.*, vol. 32, no. 10, pp. 2513–2522, Oct. 2005.

[32] B. Majhi and C. M. Anish, "Multiobjective optimization based adaptive models with fuzzy decision making for stock market forecasting," *Neurocomputing*, vol. 167, pp. 502–511, Nov. 2015.

[33] R. Singh and S. Srivastava, "Stock prediction using deep learning," *Multimedia Tools Appl.*, vol. 76, no. 18, pp. 18569–18584, Sep. 2017.

[34] F. Shen, J. Chao, and J. Zhao, "Forecasting exchange rate using deep belief networks and conjugate gradient method," *Neurocomputing*, vol. 167, pp. 243–253, Nov. 2015.

[35] A. Tsantekidis, N. Passalis, A. Tefas, J. Kanniainen, M. Gabbouj, and A. Iosifidis, "Forecasting stock prices from the limit order book using convolutional neural networks," in *Proc. IEEE 19th Conf. Bus. Informat. (CBI)*, Jul. 2017, pp. 7–12.

[36] J. Li, H. Bu, and J. Wu, "Sentiment-aware stock market prediction: A deep learning method," in *Proc. Int. Conf. Service Syst. Service Manage.*, Jun. 2017, pp. 1–6.

[37] J.-F. Chen, W.-L. Chen, C.-P. Huang, S.-H. Huang, and A.-P. Chen, "Financial time-series data analysis using deep convolutional neural networks," in *Proc. 7th Int. Conf. Cloud Comput. Big Data (CCBD)*, Nov. 2016, pp. 87–92.

[38] D. K. Kılıç and O. Uğur, "Multiresolution analysis of s&p500 time series," *Ann. Oper. Res.*, vol. 260, nos. 1–2, pp. 197–216, 2018.

[39] A. Grossmann and J. Morlet, "Decomposition of hardy functions into square integrable wavelets of constant shape," *SIAM J. Math. Anal.*, vol. 15, no. 4, pp. 723–736, Jul. 1984.

[40] S. G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 7, pp. 674–693, Jul. 1989.

[41] M. T. Ismail and A. A. A. Dghais, "Multiresolution analysis of bursa Malaysia KLCI time series," *AIP Conf. Proc.*, vol. 1847, no. 1, 2017, Art. no. 020020.

[42] S. Bekiros and M. Marcellino, "The multiscale causal dynamics of foreign exchange markets," *J. Int. Money Finance*, vol. 33, pp. 282–305, Mar. 2013.

[43] B.-L. Zhang, R. Coggins, M. A. Jabri, D. Dersch, and B. Flower, "Multiresolution forecasting for futures trading using wavelet decompositions," *IEEE Trans. Neural Netw.*, vol. 12, no. 4, pp. 765–775, Jul. 2001.

[44] D. Zhang, G. Lindholm, and H. Ratnaweera, "Use long short-term memory to enhance Internet of Things for combined sewer overflow monitoring," *J. Hydrol.*, vol. 556, pp. 409–418, Jan. 2018.

[45] A. J. Rochareis and A. P. AlvesdaSilva, "Feature extraction via multiresolution analysis for short-term load forecasting," *IEEE Trans. Power Syst.*, vol. 20, no. 1, pp. 189–198, Feb. 2005.

[46] A. Aussem and F. Murtagh, "Combining neural network forecasts on wavelet-transformed time series," *Connection Sci.*, vol. 9, no. 1, pp. 113–122, Mar. 1997.

[47] L. Di Persio and O. Honchar, "Artificial neural networks architectures for stock price prediction: Comparisons and applications," *Int. J. Circuits, Syst. Signal Process.*, vol. 10, pp. 403–413, 2016.

[48] S. Mallat, *A Wavelet Tour of Signal Processing*. New York, NY, USA: Academic, 1999.

[49] P.-H. Chiang, S. P. V. Chiluvuri, S. Dey, and T. Q. Nguyen, "Forecasting of solar photovoltaic system power generation using wavelet decomposition and bias-compensated random forest," in *Proc. 9th Annu. IEEE Green Technol. Conf. (GreenTech)*, Mar. 2017, pp. 260–266.

[50] G. Lee, R. Gommers, F. Waselewski, K. Wohlfahrt, and A. O'Leary, "PyWavelets: A Python package for wavelet analysis," *J. Open Source Softw.*, vol. 4, no. 36, p. 1237, Apr. 2019.

[51] J. Gilles, "Empirical wavelet transform," *IEEE Trans. Signal Process.*, vol. 61, no. 16, pp. 3999–4010, Aug. 2013.

[52] P. Keerthy, P. Maya, and M. R. Sindhu, "An adaptive transient tracking harmonic detection method for power quality improvement," in *Proc. IEEE Region Symp. (TENSYMP)*, Jul. 2017, pp. 1–6.

[53] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM networks," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Jul. 2005, pp. 2047–2052.

[54] L. Glass and M. Mackey, "Mackey-glass equation," *Scholarpedia*, vol. 5, no. 3, p. 6908, 2010.

KHALED A. ALTHELAYA received the master's degree in computer science from the King Fahd University of Petroleum and Minerals (KFUPM), Saudi Arabia, in 2018, where he is currently pursuing the Ph.D. degree with the Information and Computer Science Department. He has worked as a Teaching Assistant with KFUPM and Thamar University. His current research interests include deep learning, time series forecasting and analysis, scientific data visualization, the IoT security, and privacy.

SALAHADIN A. MOHAMMED received the B.S. and M.S. degrees in computer science from the King Fahd University of Petroleum and Minerals (KFUPM) and the Ph.D. degree in computer science from the Computer Science Department, Monash University, Melbourne, Australia. He is currently an Assistant Professor with the Department of Information and Computer Science, KFUPM. He has more than 30 years of experience in industry and academia, involving project management, software development, database administration, Unix administration, research, teaching, supervision, curriculum design, program assessment, and quality assurance in higher education. He is also a certified Oracle Database Administrator. He has more than 50 journal and conference publications. His research interests include machine learning, data mining and analytics, databases, data warehousing, and big data.

EL-SAYED M. EL-ALFY (Senior Member, IEEE) is currently a Professor with the Information and Computer Science Department, King Fahd University of Petroleum and Minerals (KFUPM), Saudi Arabia. He has more than 25 years of experience in industry and academia, involving research, teaching, supervision, curriculum design, program assessment, and quality assurance in higher education. He is an approved ABET/CSAB Program Evaluator (PEV), and a Reviewer and a Consultant for NCAAA and several universities and research agents in various countries. He is an active Researcher with interests in fields related to machine learning and nature-inspired computing and applications to data science and cybersecurity analytics, pattern recognition, multimedia forensics, and security systems. He has published numerously in peer-reviewed international journals and conferences, edited a number of books published by reputable international publishers, attended and contributed in the organization of many world-class international conferences, and supervised master and Ph.D. students. He was also a member of ACM, the IEEE Computational Intelligence Society, the IEEE Computer Society, the IEEE Communication Society, and the IEEE Vehicular Technology Society. His work has been internationally recognized and received a number of awards. He has served as a Guest Editor for a number of special issues in international journals, and been on the editorial board of a number of premium international journals, including IEEE/CAA Journal of Automatica Sinica, IEEE Transactions on Neural Networks and Learning Systems, *International Journal of Trust Management in Computing and Communications*, and *Journal of Emerging Technologies in Web Intelligence (JETWI)*.