Applied Soft Computing 144 (2023) 110469

Contents lists available at ScienceDirect

Applied Soft Computing

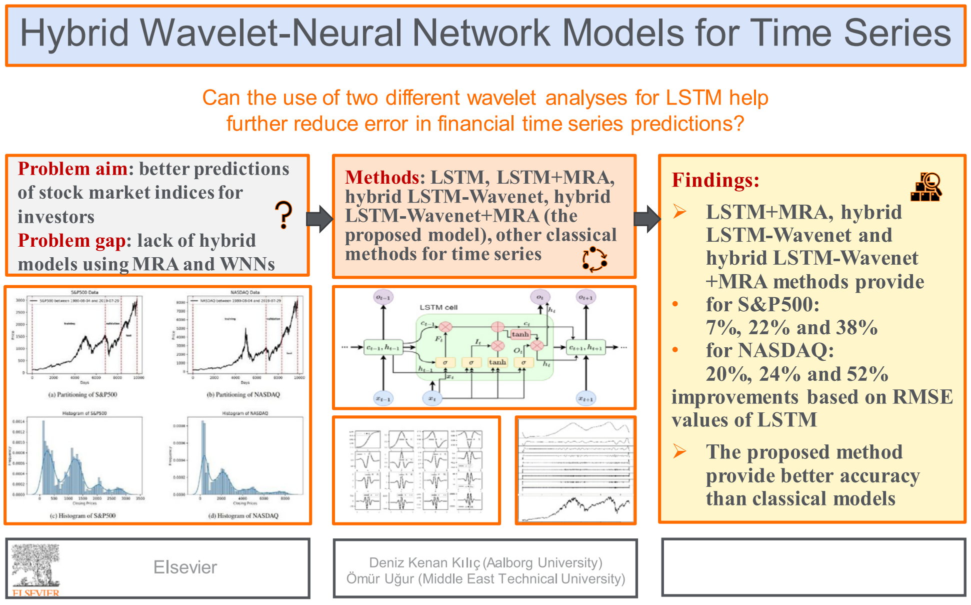
journal homepage: www.elsevier.com/locate/asoc

Hybrid wavelet-neural network models for time series

Deniz Kenan Kılıç[a,*], Ömür Uğur[b]

[a] Department of Materials and Production, Aalborg University, Fibigerstræde 16, 9220, Aalborg, Denmark
[b] Institute of Applied Mathematics, Middle East Technical University, 06800 Çankaya, Ankara, Turkey

GRAPHICAL ABSTRACT


Hybrid Wavelet-Neural Network Models for Time Series

Can the use of two different wavelet analyses for LSTM help further reduce error in financial time series predictions?

Problem aim: better predictions of stock market indices for investors
Problem gap: lack of hybrid models using MRA and WNNs

Methods: LSTM, LSTM+MRA, hybrid LSTM-Wavenet, hybrid LSTM-Wavenet+MRA (the proposed model), other classical methods for time series

Findings:
- LSTM+MRA, hybrid LSTM-Wavenet and hybrid LSTM-Wavenet +MRA methods provide
  - for S&P500: 7%, 22% and 38%
  - for NASDAQ: 20%, 24% and 52%

  improvements based on RMSE values of LSTM
- The proposed method provide better accuracy than classical models

S&P500 Data
NASDAQ Data
(a) Partitioning of S&P500
(b) Partitioning of NASDAQ
Histogram of S&P500
Histogram of NASDAQ
(c) Histogram of S&P500
(d) Histogram of NASDAQ
LSTM cell
Elsevier
Deniz Kenan Kılıç (Aalborg University)
Ömür Uğur (Middle East Technical University)

ARTICLE INFO

Article history:
Received 15 November 2022
Received in revised form 13 April 2023
Accepted 20 May 2023
Available online 7 June 2023

Keywords:
Nonlinear models
Long short-term memory (LSTM)
Recurrent neural network (RNN)
Time series analysis
Multiresolution analysis (MRA)
Wavenet
Wavelet neural network (WNN)

ABSTRACT

The use of wavelet analysis contributes to better modeling for financial time series in the sense of both frequency and time. In this study, S&P500 and NASDAQ data are separated into several components utilizing multiresolution analysis (MRA). Subsequently, using an appropriate neural network structure, each component is modeled. In addition, wavelets are used as an activation function in long short-term memory (LSTM) networks to form a hybrid model. The hybrid model is merged with MRA as a proposed method in this paper. Four distinct strategies are employed: LSTM, LSTM+MRA, hybrid LSTM-Wavenet, and hybrid LSTM-Wavenet+MRA. Results show that the use of MRA and wavelets as an activation function together reduces the error the most.

1. Introduction

Plenty of time series is non-stationary and chaotic in many fields. For such complex time series, nonlinear models are more

* Corresponding author.
E-mail addresses: denizkk@mp.aau.dk (D.K. Kılıç), ougur@metu.edu.tr (Ö. Uğur).

https://doi.org/10.1016/j.asoc.2023.110469

adequate than linear models (see [1]). Considering the wide variety and adaptability, neural networks are among the most used and widespread nonlinear models.

In the literature, several prediction and classification models are used for time series analysis [2–10] and specifically quantitative finance [11,12]. Examples of some statistical and machine learning models that are employed in quantitative finance are autoregressive moving average (ARMA) [13], autoregressive integrated moving average (ARIMA) [14–16], seasonal ARIMA (SARIMA) [17,18], exponential smoothing, linear regression, logistic regression [19], least absolute shrinkage and selection operator (LASSO) regression [20,21], Naïve Bayes [22,23], decision tree [24], random forest [22–25], support vector machine (SVM) [16,19,22,25], k-nearest neighbor (KNN) [23], gradient-boosted decision tree (such as extreme gradient boosting (XGBoost), light gradient-boosting machine (lightGBM)) [15,21,24,26], Prophet [27].

Furthermore, there are also studies concerning neural networks, such as artificial neural network (ANN) [22,24,28], extreme learning machine (ELM) [29], multi-layer perceptron (MLP) [23,30], recurrent neural network (RNN) [24,29], convolutional neural network (CNN) [31,32], long short-term memory (LSTM) [14–16,24,31–34], etc.

The performance rankings of the methods used in the related literature may vary according to the data type and parameter selection [4,5,14,15,18,27,35,36]. For this reason, instead of finding the best model between classical or state-of-the-art methods, the study focuses on the question of whether the LSTM model, which is one of the state-of-the-art methods, can be improved. While doing this, benchmarking is accomplished by using several other methods as well.

In a very large variety of fields, wavelets are used and their scopes are growing day by day. For complex time series, the frequency domain is crucial for most situations. As the data gets more complex, examining it in both time and frequency domains brings improvements in data analysis, modeling, and prediction [1,37,38]. Moreover, for both linear and nonlinear time series, multiresolution analysis (MRA) gives better modeling and forecasting results [1].

In this paper, making better predictions for stock market indexes is investigated to guide buying and selling attitudes of investors. Furthermore, several up-to-date applied studies and competitions on online platforms comprise traditional and advanced machine learning topics. Therefore, this study contributes to both literature and online studies by using MRA and wavenet with LSTM networks.

Wavenet is a type of wavelet neural network (WNN), where the translation and the dilation parameters are not trainable parameters. In other words, these parameters do not change in the activation functions. In this research, activation functions are generated using the polynomial powers of sigmoid (PPS) and model accuracy is increased by the use of MRA and LSTM.

Developing models by taking advantage of both neural networks and wavelets is a method used by many researchers. Jothimani et al. [39] synthesize wavelets and nonlinear models to predict the stock market data. They show that the accuracies of hybrid models are higher than the accuracy of classical models. Similarly, the study of Chandar et al. [40] also illustrates that utilizing wavelets with neural networks increases the accuracy of the stock market data.

Jin and Kim [41] make predictions about natural gas prices using the ARIMA model and ANN. They conclude that using wavelet with ANN and ARIMA gives better results than using only ANN and ARIMA models respectively.

A wavelet-based neural network structure for two deep learning models in time series classification and forecasting is studied by Wang et al. [42]. Results again show that hybrid techniques improve the outcomes of the models without wavelets.

Arévalo et al. [43] specify that using wavelet analysis for high-frequency financial data increases the accuracy of each ARIMA, deep neural network (DNN), gated recurrent unit (GRU), and LSTM methods.

Liu et al. [44] predict non-stationary wind power time series using discrete wavelet transform (DWT)-LSTM, DWT-RNN, DWT-back-propagation neural network (BP), LSTM, RNN, and BP. The proposed method, particularly DWT-LSTM, outperforms all the other models.

Some studies practice wavelets only for data decomposition to use sub-series as normal neural network inputs [45–50], while some works utilize wavelets just for activation functions of WNN [51–55]. Both MRA and WNN provide great benefits in many fields. Therefore, the motivation of this paper is to benefit from wavelet theory from multiple directions to increase the accuracy of the LSTM model for time series prediction. This paper aims to bridge the gap between these two different approaches by combining them.

The paper is organized as follows. As preliminary knowledge and the fundamental concepts used in this paper and the essentials of the LSTM are provided in Section 2.1. Further in Section 2.2, the significant technical points in using wavelets are addressed; maximal overlap discrete wavelet transform (MODWT) and MRA are briefly explained. Finally, in Section 2.3, the essential details of PPS are listed, as the wavelets used in the empirical analysis are derived from PPS. In Section 3, the LSTM and wavelets are used to model S&P500 and NASDAQ financial time series using four different methods: LSTM model without MRA, LSTM model with MRA, hybrid LSTM-Wavenet model without MRA, and hybrid LSTM-Wavenet model with MRA. The results obtained by these methods are fully discussed in Section 4. In addition, the results of classical statistical and machine learning models are compared with the proposed approach. A brief conclusion as well as a possible extension of the methodology is given in Section 5. Finally, Appendix contains the configurations used by the models used in the paper.

2. Preliminaries and the fundamental concepts

In this section, a short preliminary knowledge of LSTMs as well as wavelets is given; however, for the methodology used in the paper to be understood well enough the fundamentals of MRA as well as the WNN are also recalled briefly.

2.1. LSTM

LSTM, which is a specific type of RNN, is used in this study for financial time series analysis. Because financial data has fluctuations for both short and long time intervals, the network structure needs to own various memories for different time gaps.

Hochreiter and Schmidhuber [56] list several advantages of LSTM, while Viswanath [57] describes differences between ANN, RNN, and LSTM. The mathematical formulation of RNN can be described by

$$h_t = f_H\left(W_{IH}x_t + W_{HH}h_{t-1}\right), \tag{1}$$

$$y_t = f_O\left(W_{HO}h_t\right), \tag{2}$$

where h_t, x_t and y_t are hidden state, input, and output vectors; W_{IH}, W_{HH} and W_{HO} are the weight matrices; f_H and f_O are the activation functions for hidden and output parts, respectively. RNN would have vanishing or exploding gradient problems; however, LSTM could solve these obstacles by appending additional parts like the input gate, the forget gate, and the output gate. Therefore,

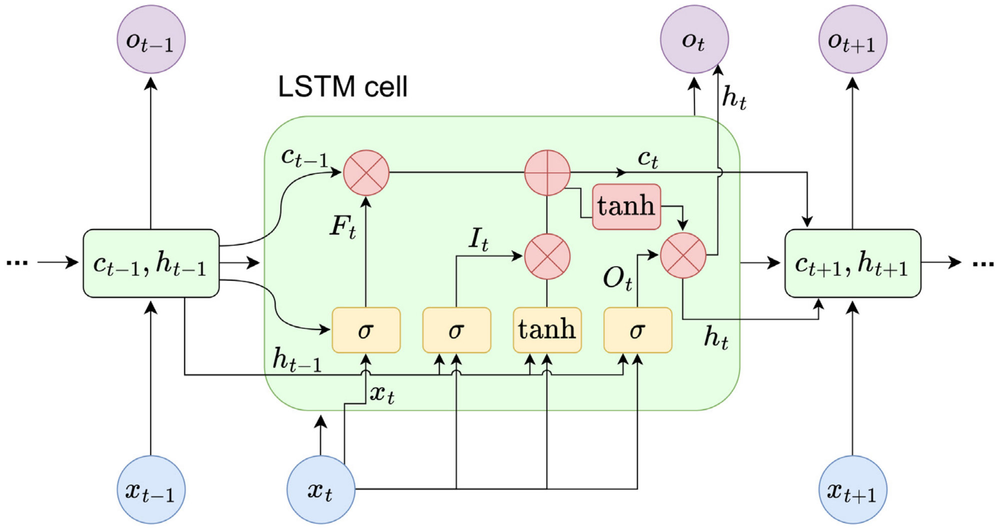


Fig. 1. Structure of a LSTM cell.

LSTM would be a more suitable choice for time series modeling and prediction.

In Fig. 1 one unit LSTM is illustrated; particularly equations for estimating the output h_t of the memory cell at time t are given as follows:

$$F_t = \sigma(W_F x_t + U_F h_{t-1} + b_F), \tag{3}$$

$$I_t = \sigma(W_I x_t + U_I h_{t-1} + b_I), \tag{4}$$

$$\tilde{C}_t = \tanh(W_C x_t + U_C h_{t-1} + b_C), \tag{5}$$

$$C_t = F_t \odot C_{t-1} + I_t \odot \tilde{C}_t, \tag{6}$$

$$O_t = \sigma(W_O x_t + U_F h_{t-1} + b_O), \tag{7}$$

$$h_t = O_t \odot \tanh(C_t), \tag{8}$$

where x_t is input vector to the LSTM at time t, the W's and U's are weight matrices of the input and recurrent connections, the b's are bias vectors, h_t in (8) is the output vector of LSTM cell, C_t in (6) and $\tilde{C}_t$ in (5) are state and candidate state vectors respectively, F_t in (3) contains the forget gate values, I_t in (4) includes input gate values and O_t in (7) covers output gate values. For further details of LSTM and how it works, one could refer to [58].

2.2. Wavelets

A signal can be split into high-frequency and low-frequency components in appropriate time intervals with the aid of the wavelet transform. Since financial time series have a high-frequency for short time lengths and a low-frequency for lengthy time periods; apparently, the use of wavelets should be appropriate and of great help.

2.2.1. MODWT

In theory, it is more common to apply continuous wavelet transform (CWT) for continuous functions but not for discrete signals as Masset remarks in [37]. With this, MRA with sampled wavelets is described by using MODWT.

MODWT has some advantages as opposed to DWT. MODWT does not demand dyadic length time series where DWT does. In other words, DWT restricts the data for owning a length of $N = 2^J$ where J is the scale level. The sizes of wavelet and scaling coefficients are equal to the original time series's length at every step of the transform for MODWT. Besides, MODWT is time-shift invariant while DWT is influenced by time-shifting as stated in [59]. Therefore, the variance analysis (i.e., scale-based analysis of variance) of MODWT is more productive than the variance analysis of DWT as mentioned in [60].

Let $\mathbf{w}$ hold wavelet and scaling coefficients of the MODWT,

$$\mathbf{w} = [w_1, w_2, \ldots, w_J, v_J]^T, \tag{9}$$

where the length of w_j (wavelet coefficients) is $N/2^j$ and the length of v_J (scaling coefficient) is $N/2^J$; these are consistent with scale sizes $\lambda_j = 2^{j-1}$ and $\lambda_J = 2^{J-1}$, respectively for $j = 1, 2, \ldots, J$. The vector $\mathbf{w}$ is obtained by using high-pass and low-pass filters as mentioned in [59], and

$$\mathbf{w} = \mathbf{W}\xi, \tag{10}$$

where $\mathbf{W} = [W_1, W_2, \ldots, W_J, V_J]^T$ is $(J+1)N \times N$ matrix in which each W_j and V_J are $N \times N$ matrices and ξ is the underlying time series (signal).

High-pass and low-pass filters are convolved with the time series to obtain wavelet and scaling coefficients of the first level as follows:

$$w_1(t) = \sum_{l=0}^{L-1} \widetilde{h}_l \xi(\dot{t}) \quad \text{and} \quad v_1(t) = \sum_{l=0}^{L-1} \widetilde{g}_l \xi(\dot{t}), \tag{11}$$

where $t = 0, 1, \ldots, N-1$ and $\dot{t} = t - l \pmod N$. Wavelet and scaling coefficients of the second level are captured by convolving $v_1(t)$ with the high-pass filter $\widetilde{h}_l$ and low-pass filter $\widetilde{g}_l$. After $J = \log_2 N$ convolution iterations, wavelet and scaling coefficients become:

$$w_J(t) = \sum_{l=0}^{L-1} \widetilde{h}_l v_{J-1}(\dot{t}) \quad \text{and} \quad v_J(t) = \sum_{l=0}^{L-1} \widetilde{g}_l v_{J-1}(\dot{t}), \tag{12}$$

where $\dot{t} = t - 2^{J-1} l \pmod N$.

As (11) and (12) are convolved with high-pass filter and low-pass filter respectively, the scaling coefficient of the previous level is received by summing the two convolving parts [59]:

$$v_{J-1}(t) = \sum_{l=0}^{L-1} \widetilde{h}_l w_J(\dot{t}) + \sum_{l=0}^{L-1} \widetilde{g}_l v_J(\dot{t}). \tag{13}$$

This scheme is iterated up to the first level of wavelet and scaling coefficients to get the original time series, written in the form,

$$\xi(t) = \sum_{l=0}^{L-1} \widetilde{h}_l w_1(\dot{t}) + \sum_{l=0}^{L-1} \widetilde{g}_l v_1(\dot{t}), \tag{14}$$

where $\dot{t} = t + l \pmod N$.

2.2.2. MRA

MRA is a sequence of closed nested subspaces $\left\{V_j : j \in \mathbb{Z}\right\}$ in $\mathrm{L}^2(\mathbb{R})$ with the conditions [61–63]:

1. $\{\phi(x-k) : k \in \mathbb{Z}\}$ is an orthonormal basis for V_0, where ϕ is scaling function,
2. $\{0\} \subset \cdots \subset V_j \subset V_{j+1} \subset \cdots \subset \mathrm{L}^2(\mathbb{R})$,
3. Closure of $\left(\cup_{j\in\mathbb{Z}} V_j\right) = \mathrm{L}^2(\mathbb{R})$,
4. $\cap_{j\in\mathbb{Z}} V_j = \{0\}$,
5. $f(t) \in V_j \Leftrightarrow f(2t) \in V_{j+1}$; i.e., the spaces V's are self-similar,
6. $V_{j+1} = V_j \oplus W_j$ where the W_j is the jth resolution level and $V_j \cap W_j = \{0\}$.

Scaling $\varphi_{j,k}$ and wavelet $\psi_{j,k}$ functions generate bases for V_j and W_j subspaces by applying scaling and translation parameters respectively:

$$V_j = \mathrm{span}\left\{\varphi_{j,k}(x)\right\}, \tag{15}$$

$$W_j = \mathrm{span}\left\{\psi_{j,k}(x)\right\}. \tag{16}$$

After defining MRA by utilizing subspaces in (15) and (16), $\mathrm{L}^2(\mathbb{R})$ space or any function in it can be written as a direct sum of V_0 and W_j subspaces, since

$$\mathrm{L}^2(\mathbb{R}) = V_0 \oplus W_0 \oplus W_1 \oplus W_2 \oplus \cdots \oplus W_j \quad \text{for } j = 0, 1, 2, \ldots \tag{17}$$

Now, it should be clear that any function f can be given in the form of wavelet series expansion:

$$f(x) = \sum_k a_{j_0}(k)\varphi_{j_0,k}(x) + \sum_{j=j_0}^{\infty}\sum_k d_j(k)\psi_{j,k}(x) \quad \text{for } j > j_0, \tag{18}$$

in which the first sum with the scaling function represents the smooth part and the second sum represents the detailed parts of the function. The following integrals are used to find the coefficients of the expansion of f in (18):

$$a_{j_0}(k) = \int f(x)\varphi_{j_0,k}\, dx \quad \text{and} \quad d_j(k) = \int f(x)\psi_{j,k}\, dx. \tag{19}$$

In practice, the scale level J is chosen to be finite, and hence the function in (18) can be written as a time series as follows:

$$f(t) = \sum_k a_{J,k}\varphi_{J,k}(t) + \sum_{j=j_0}^{J}\sum_k d_{j,k}\psi_{j,k}(t) \quad \text{for } j = 1, 2, \ldots, J. \tag{20}$$

Using the celebrated Mallat's pyramid algorithm [64], MODWT divides (20) into smooth and detailed parts

$$f(t) = A_{J,k} + \sum_{j=j_0}^{J} D_{j,k}, \tag{21}$$

where the component $A_{J,k}$ keeps the average information (or trend) of the original data at the largest scale and is associated with the scaling coefficients. Components $D_{j,k}$'s, from the first scale to the last, are linked with wavelet coefficients. They are implemented for accumulating higher frequency information [37].

2.3. WNN

The goal of this part is to benefit both wavelets and neural networks for better modeling of the time series. For complex data, such as financial time series, the WNN is functional.

Traditional activation functions such as logistic, hyperbolic tangent, rectified linear unit (ReLU), softmax, etc., or some custom linear/nonlinear activation functions are used in neural networks. On the other hand, WNNs' activation functions consist of wavelets in the hidden layer neurons. The hidden layer neurons in WNN are named wavelons [65–67]. Wavelons have two parameters, which are termed translation and dilation. The single wavelon is given by

$$\psi_{u,v}(x) = \psi\left(\frac{x-u}{v}\right), \tag{22}$$

where u is the translation (or the location) and v is the dilation (or the scale) parameter.

Wen et al. [68] state that since wavelets are quickly vanishing functions, it is important not to select too small dilation parameters. Moreover, Radhwane and Bereksi [69] point out that random initialization of the translation and dilation parameters may result in too local wavelets.

2.3.1. Wavenets

Veitch [70] states that if the translation and the dilation parameters given in (22) are fixed for the learning process, then the network is called wavenet. In this study, wavenets are utilized.

Parameters specified in (22) may alter for each node in the hidden layer. These parameters need to be initialized according to the data (see Section 3). In this study, the wavelet functions ψ's are acquired from PPS. Marar and Bordin [71] state that there are many limitations in the traditional backpropagation algorithm, and hence, a family of polynomial wavelets generated from powers of sigmoid functions is used to eliminate these limitations.

2.3.2. PPS

Fernando et al. [72] declare that a group of polynomial wavelets created from the powers of sigmoid provides robust neural networks, particularly WNNs. Consecutive powers of sigmoid functions are used to produce polynomial types of wavelet functions so that the square integrability and admissibility conditions,

$$\int_{\mathbb{R}} \|f(x)\|^2\, dx < \infty \quad \text{and} \quad \int_0^{\infty} \frac{\|\mathfrak{C}(\omega)\|^2}{\omega}\, d\omega < \infty, \tag{23}$$

are satisfied, where ω is frequency and $\mathfrak{C}$ is the Fourier transform of $\psi(x)$.

Particularly, consider the following sigmoid function

$$f(x) = \frac{1}{1+\exp(-\alpha x)}, \tag{24}$$

where α is the smoothness constant. To create functions of wavelet family from the sigmoid function in (24), the nth power of the sigmoid function and the set of all powers of the sigmoid function are used.

The polynomial wavelet function is given as (see [73])

$$\psi_n(x) = \sum_k^{n}\sum_{j=0}^{k}(-1)^j(j+1)^n\binom{k}{j}f^{(k+1)}(x), \tag{25}$$

where n is the order of the derivative of the sigmoid function and $f^{(k+1)}(x)$ is the $(k+1)$th derivative of the sigmoid function. Particularly, the first, the second, and the third polynomial types of wavelet functions in (25), which use consecutive powers of sigmoid functions, are given respectively, as

$$\psi_1(x) = -f^{(2)}(x) + f(x), \tag{26}$$

$$\psi_2(x) = 2f^{(3)}(x) - 3f^{(2)}(x) + f(x), \tag{27}$$

$$\psi_3(x) = -6f^{(4)}(x) + 12f^{(3)}(x) - 7f^{(2)}(x) + f(x). \tag{28}$$

Note that the observed family of polynomial wavelets $\psi_i \in \mathrm{L}^2(\mathbb{R})$. Consequently, the conditions (23) hold not only for this family but also for shifted and dilated versions of this family.

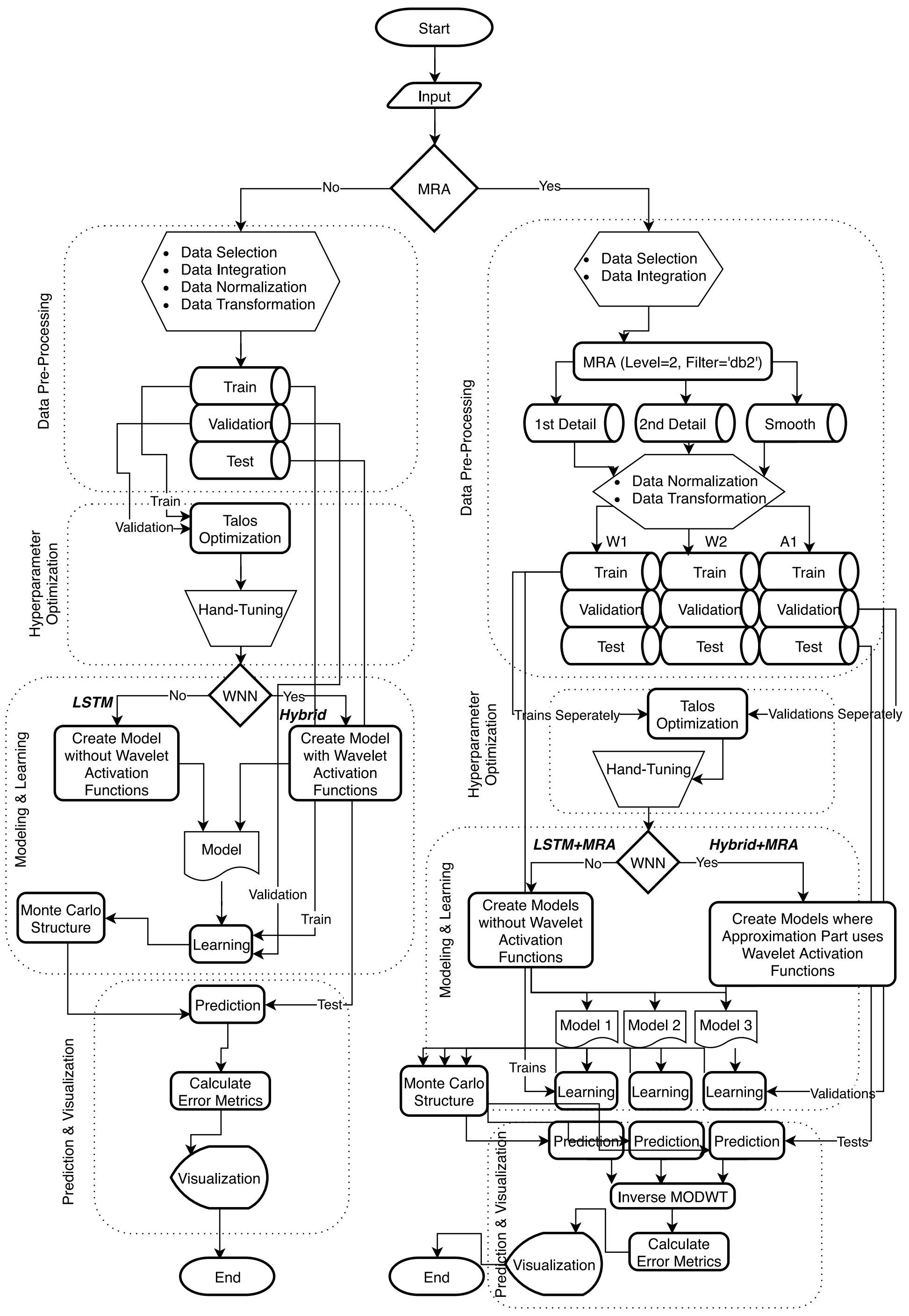


Fig. 2. The flowchart of four different methods employed in the study.

3. Configurations and empirical results

The flowchart of LSTM, LSTM+MRA, hybrid LSTM-Wavenet, and hybrid LSTM-Wavenet+MRA methods are illustrated in Fig. 2. There is a decision point that determines whether MRA will be used or not. It is represented to show whether the algorithm is working with or without an MRA. The general flow for both cases includes data preprocessing, hyperparameter optimization, modeling and learning, prediction and visualization. Main parts of the flowchart are described below.

Data preprocessing. In this step, it is decided which parts of the data will be used. Then selected financial data (S&P500 and NASDAQ) is combined, and this fused data is normalized between 0 and 1. Normalization for the ith input is done via $x_i \leftarrow \frac{x_i - x_{\min}}{x_{\max} - x_{\min}}$, where x_i is the input value, $x_{\max}$ and $x_{\min}$ are the maximum and the minimum values of the data. The data is then converted to the structure of a supervised learning type by using certain time-steps.[1] The modified input is eventually divided into the train set, the validation set, and the test set.

[1] Several time-steps conditions are tried for the LSTM structure practiced to S&P500 data in [74]. The value 10 for the time-steps gives the most favorable outcome. Accordingly, 10 is used as a time-steps in the study.

Table 1
Descriptive statistics of S&P500 data.

	All data	Training data	Validation data	Test data
Start date	1980–08–04	1980–08–04	2007–10–10	2013–09–04
End date	2019–07–29	2007–10–09	2013–09–03	2019–07–29
# of observations	9830	6860	1485	1485
Minimum	102.42	102.42	676.53	1653.08
Maximum	3025.86	1565.15	1709.67	3025.86
Range	2923.44	1462.73	1033.14	1372.78
Mean	990.26	654.61	1247.78	2283.34
Median	993.52	459.59	1278.18	2143.16
Variance (ddof = 0)	511585.07	205914.77	46644.40	129766.45
Std (ddof = 0)	715.25	453.78	215.97	360.23
Variance (ddof = 1)	511637.11	205944.79	46675.83	129853.90
Std (ddof = 1)	715.29	453.81	216.05	360.35
Skewness	0.74	0.44	−0.24	0.37
Kurtosis	−0.10	−1.34	−0.38	−1.16

Hyperparameter optimization. The train set and the validation set are used in the Talos optimization process [75]. Automatically selected hyperparameters are again optimized manually.

Modeling and learning. If wavenets are not used, then the created model is called the LSTM model. On the other hand, if wavelet activation functions are utilized, the model is called a hybrid LSTM-Wavenet model. The train set and the validation set are used to trigger the learning process for the selected model.

Prediction and visualization. After the learning process, the models are used for the prediction using the test set. Further, the model training set is also predicted. All predictions are denormalized via $x_i \leftarrow x_i(x_{\max} - x_{\min}) + x_{\min}$. Monte Carlo estimates are used to get the mean error metric scores for the train set and the test set. In addition to calculating different error metric results, visuals of the analysis are acquired.

On the other hand, one should also notice that when MRA is used, the data is decomposed after data selection and integration parts in data preprocessing. All decomposed data levels are normalized and transformed into a supervised learning problem. Each level is subdivided into the train set, the validation set, and the test set. Later on, hyperparameter optimization is done for each subseries by Talos optimization and hand-tuning[2] methods. If wavelets are used as an activation function, then the model is named a hybrid LSTM-Wavenet model with MRA. Otherwise, the model is called an LSTM model with MRA. After predicting train and test parts for each level by using Monte Carlo estimates, inverse MODWT is applied to the predictions to reconstruct the original time series.

The whole analysis is done using Python language in Anaconda. To perform ANN, the Keras library, which uses the TensorFlow backend, is used [76].

Batch normalization is practiced in each analysis between LSTM and dense layers to produce smaller pieces of data with a mean of zero and a standard deviation of one. In [77,78] it is claimed that batch normalization has some advantages like reducing overfitting problems, speeding up the training process, and improving accuracy results/decreasing loss values.

The list of the specifications of the computing environment codes run on as well as the error metrics used are given in Appendix.

3.1. Data sets

S&P500 (^GSPC) and NASDAQ (^IXIC) stock data between 1980-08-04 and 2019-07-29 are downloaded from https://finance.yahoo.com/. Data has 9830 observations with "Open, High, Low, Close, Adj Close, Volume" columns. Only closing prices are used in experimental parts. Data summary tables for S&P500 and NASDAQ are given in Table 1 and Table 2, respectively.

S&P500 is a USA-based stock market index that carries 500 large corporations registered on stock exchanges. S&P500 includes finance, health care, industry, energy, information technology, and many other sectors. NASDAQ Composite is also another important USA-based stock market index that embraces the information technology sector. Because S&P500 keeps almost every sector, it is less volatile than NASDAQ. NASDAQ is supposed risky, while S&P500 is considered risk-free. Portfolio diversification by investing in both riskless and risky markets is crucial for market players and investors.

In Figs. 3(a) and 3(b), almost 70% of the data is in the train set (length of 6860), about 15% is in the validation set (length of 1485), and nearly 15% of all data is in the test set (length of 1485). It is noticed that there are large changes in values between the training set, the validation set, and the test set. Also, there are fluctuations: the minimum, maximum, mean, and standard deviation of both S&P500 and NASDAQ are quite different for the training set, the validation set, and the test set; thence classical methods may not be appropriate choices to be expected to work efficiently.

3.2. LSTM model without MRA

Firstly, closing prices of S&P500 and NASDAQ stocks are concatenated. Each sample is related to index 0 for closing prices of S&P500 and index 1 for closing prices of NASDAQ, respectively. By using ten days window set, the next day's closing price is focused on being predicted.

Two hidden layers are used, where the first one consists of LSTM nodes, and the second one is formed of a regular densely-connected neural network. The kernel, recurrent, and bias regularizers are used in the LSTM layer. Batch normalization and ReLU activation function are joined between dense and LSTM layers, respectively. In the dense part, kernel and bias regularizers are utilized.

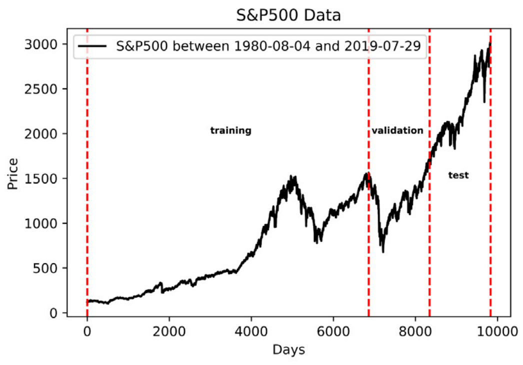
After modeling the problem according to the selected configuration using Talos optimization and hand-tuning, the model is fit to the data and predicts train/test data multiple times within a loop. Subsequently, train/test predictions and the means of the error metrics are estimated.

Results are affected by tuning L_1 regularization and L_2 regularization in the kernel, recurrent and bias regularizers for a fixed epoch value. The optimized Talos configuration is given in Table A.13. Training and test results which are obtained using the optimized parameters are given in Table 3 for S&P500 and NASDAQ.

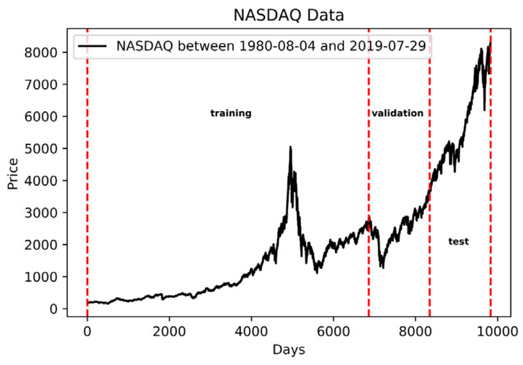
[2] At this stage, kernel initializer, kernel regularizer, recurrent regularizer, and bias regularizer are optimized.

Table 2
Descriptive statistics of NASDAQ data.

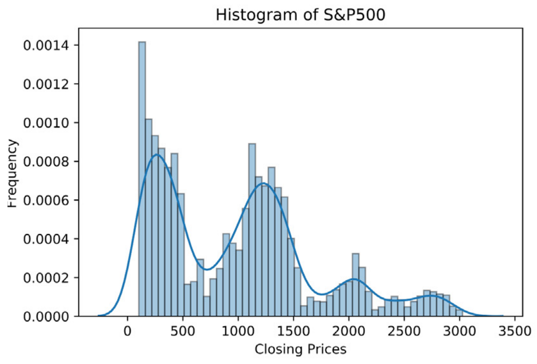
	All data	Training data	Validation data	Test data
Start date	1980–08–04	1980–08–04	2007–10–10	2013–09–04
End date	2019–07–29	2007–10–09	2013–09–03	2019–07–29
# of observations	9830	6860	1485	1485
Minimum	159.14	159.14	1268.64	3649.04
Maximum	8330.21	5048.62	3692.95	8330.21
Range	8171.07	4889.48	2424.31	4681.17
Mean	2042.74	1147.35	2519.96	5701.76
Median	1697.72	743.45	2516.69	5190.10
Variance (ddof = 0)	3547004.08	909358.80	268338.30	1690667.54
Std (ddof = 0)	1883.35	953.60	518.01	1300.26
Variance (ddof = 1)	3547364.95	909491.37	268519.12	1691806.81
Std (ddof = 1)	1883.45	953.67	518.19	1300.69
Skewness	1.33	1.09	−0.14	0.44
Kurtosis	1.29	0.78	−0.34	−1.18



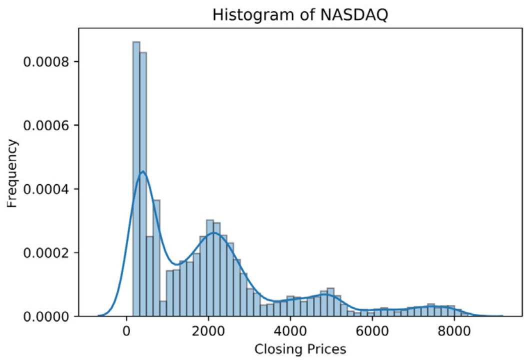
(a) Partitioning of S&P500



(b) Partitioning of NASDAQ



(c) Histogram of S&P500



(d) Histogram of NASDAQ

Fig. 3. Partitioning graphs and histograms.

Table 3
LSTM model, configuration (S&P500/NASDAQ): mean scores for the train set and the test set by running 1000 experiments.

Configuration		
Experiment size	1000	
Time taken by process	228 m 29 s/234 m 49 s	
Monte Carlo Scores	Reconstructed Train Scores	Reconstructed Test Scores
RMSE	11.93/36.65	29.06/85.31
Scaled RMSE	0.00/0.00	0.01/0.01
R^2	1.00/1.00	0.99/1.00
MAE	8.06/21.44	23.30/66.58
EVS	1.00/1.00	1.00/1.00
ME	107.80/492.62	139.80/324.16
MdAE	4.86/11.35	19.93/53.22

Table 4
LSTM model+MRA, configuration (S&P500/NASDAQ): mean scores for the synthesized train set and the synthesized test set by running 1000 experiments.

Configuration			
Experiment size	1000	wavelet filter	db2 (D4)
Time taken by process	1981 m 47 s/2008 m 34 s	wavelet level	2
Monte Carlo Scores	Reconstructed Train Scores	Reconstructed Test Scores	
RMSE	15.29/33.12	26.91/65.41	
Scaled RMSE	0.01/0.00	0.01/0.01	
R^2	1.00/1.00	0.99/1.00	
MAE	12.29/20.74	22.66/52.67	
EVS	1.00/1.00	1.00/1.00	
ME	239.30/612.95	104.68/244.27	
MdAE	11.11/11.67	20.36/44.05	

3.3. LSTM model with MRA

In this section, the data is decomposed into two detail parts and one approximation part by applying MRA. Daubechies wavelets, particularly the "db2=D4" filter and MODWT, are used to decompose the time series with a level of two. Then the first detail, second detail, and approximation parts are obtained as subseries. Each subseries (length of 9830) is split into the train, validation, and test sets. Almost 70% of each level is used in the train set (length of 6860), about 15% of each level is used in the validation set (length of 1485), and nearly 15% is used in the test set (length of 1485).

The same network form is used in the approximation level as built in Section 3.2. On the other hand, a single LSTM hidden layer is used for the first and the second detail levels independently. Each level is modeled by using Talos optimization and hand-tuning methods. The model is then fit for the data to predict train and test datasets for 1000 experiments. After denormalizing predictions, inverse MODWT is applied to these predictions of the train and the test separately. Lastly, the means of the error metrics are calculated and train/test predictions are synthesized.

In Table A.14 configurations of the first detail, second detail, and approximation are presented for S&P500 and NASDAQ. In Table 4 means of 1000 reconstructed scores are given for S&P500 and NASDAQ. It is seen that using MRA improves the test results based on RMSE, MAE, and MdAE metrics except for the MdAE scores of S&P500.

3.4. Hybrid LSTM-wavenet model without MRA

PPS is used to generate an internal activation function for the time series in this part. Details of polynomial wavelets are given in Section 2.3.2. Data preparation, time-steps, and network structure are the same as used in Section 3.2.

When the polynomial wavelet function generated by the nth derivative of the sigmoid function is used for the nth node of the LSTM layer, results are much worse than those when a single polynomial wavelet function in each cell is used. The polynomial wavelet function generated by the 6th derivative of the sigmoid function is used to create a wavelet activation function in each cell. Each activation function with index j is generated as

$$\psi_6^j(x) = \psi\left(\frac{x-u_j}{v_j}\right), \tag{29}$$

where u is the translation (or location) and v is the dilation (or scale) parameters for $j = 1, 2, \ldots, 16$. The subscript 6 denotes the order of the derivative. Translation and dilation parameters are initialized as

$$u_1 \approx \frac{1}{2}\left(\frac{\beta-\alpha}{n}\right) \quad \text{and} \quad v_1 \approx \frac{1}{2}\left(\frac{\beta+\alpha}{n}\right), \tag{30}$$

where α and β are, respectively, the minimum and the maximum values in the training set; n is the number of LSTM nodes.

As a result, initial translation and dilation parameters in (30) for S&P500 data are $u_1 = 40$ and $v_1 = 50$. On the other hand, the same values are used for the initial parameters of NASDAQ data for the sake of the same configurations. Thus,

$$\psi_6^j(x) = \psi\left(\frac{x-40j}{50j}\right), \tag{31}$$

where j is the index of the activation functions.

Two different approaches are used for the hybrid LSTM-wavenet model without MRA. In the first method, the same wavelet activation function is used for all 16 LSTM nodes, where j is fixed to 6 in (31). In the second strategy, different activation functions are created using (31) for $j = 1, 2, \ldots, 16$. The second approach is named configuration by API since the functional API of Keras is used to create models.

In both approaches, learning and prediction processes are carried out with the model created after selecting the model parameters using the Talos optimization and hand-tuning methods. Finally, the mean values of predictions and errors are calculated.

In Table A.15 configuration parameters of the first approach are given for S&P500 and NASDAQ. Results obtained are presented in Table 5 for S&P500 and NASDAQ. It is clear that the results are better than the non-hybrid methods.

Configuration parameters of the second approach (configuration by API) are given in Table A.16 for S&P500 and NASDAQ. Outcomes are shown in Table 6 and for S&P500 and NASDAQ. Once again, outcomes are superior to the methods LSTM and LSTM+MRA according to RMSE, MAE, and MdAE metrics.

3.5. Hybrid LSTM-wavenet model with MRA

MRA and the hybrid LSTM-Wavenet model are combined in this section in order to obtain the ultimate possible benefit from MRA. The model structure of the proposed hybrid LSTM-Wavenet with MRA model is demonstrated in Fig. 4. In substance, MRA is used to decompose time series, and wavenet form is applied to the approximation part. Finally, all outputs derived from different decomposition levels are merged by inverse wavelet transform.

MRA, data preparation, and selection of the time-steps processes are carried out exactly in the same way as in Section 3.3. However, in this case, two different approaches, mentioned in Section 3.4, are used to create activation functions by utilizing PPS.

Daubechies wavelets, particularly the "db2=D4" filter and MODWT, are employed to decompose the time series with a level of two as selected in Section 3.3.

Each level is modeled by taking advantage of both Talos optimization and hand-tuning again. Following that, a thousand experiments are conducted for the model fitting and predictions of train/test sets. The means of the error metrics and train/test predictions are then averaged for each wavelet level. Then, the mean error values of the reconstructed data are calculated.

Table 5
Hybrid LSTM-Wavenet model, configuration 1 (S&P500 /NASDAQ): mean scores for the train set and the test set by running 1000 experiments.

Configuration 1		
Experiment size	1000	
Time taken by process	459 m 45 s/391 m 24 s	
Monte Carlo Scores	Reconstructed Train Scores	Reconstructed Test Scores
RMSE	9.71/30.15	22.82/64.57
Scaled RMSE	0.00/0.00	0.01/0.01
R^2	1.00/1.00	1.00/1.00
MAE	6.37/16.18	17.33/46.77
EVS	1.00/1.00	1.00/1.00
ME	86.70/355.23	126.88/373.57
MdAE	3.76/7.54	13.74/34.45

Table 6
Hybrid LSTM-Wavenet model by API structure, configuration 2 (S&P500/NASDAQ): mean scores for the train set and the test set by running 1000 experiments.

Configuration 2		
Experiment size	1000	
Time taken by process	2464 m 15 s/2379 m 44 s	
Monte Carlo Scores	Reconstructed Train Scores	Reconstructed Test Scores
RMSE	9.58/30.70	24.04/67.30
Scaled RMSE	0.00/0.00	0.01/0.01
R^2	1.00/1.00	0.99/1.00
MAE	6.29/16.94	18.40/49.41
EVS	1.00/1.00	1.00/1.00
ME	84.81/355.76	128.34/ 378.39
MdAE	3.77/8.53	14.48/36.80

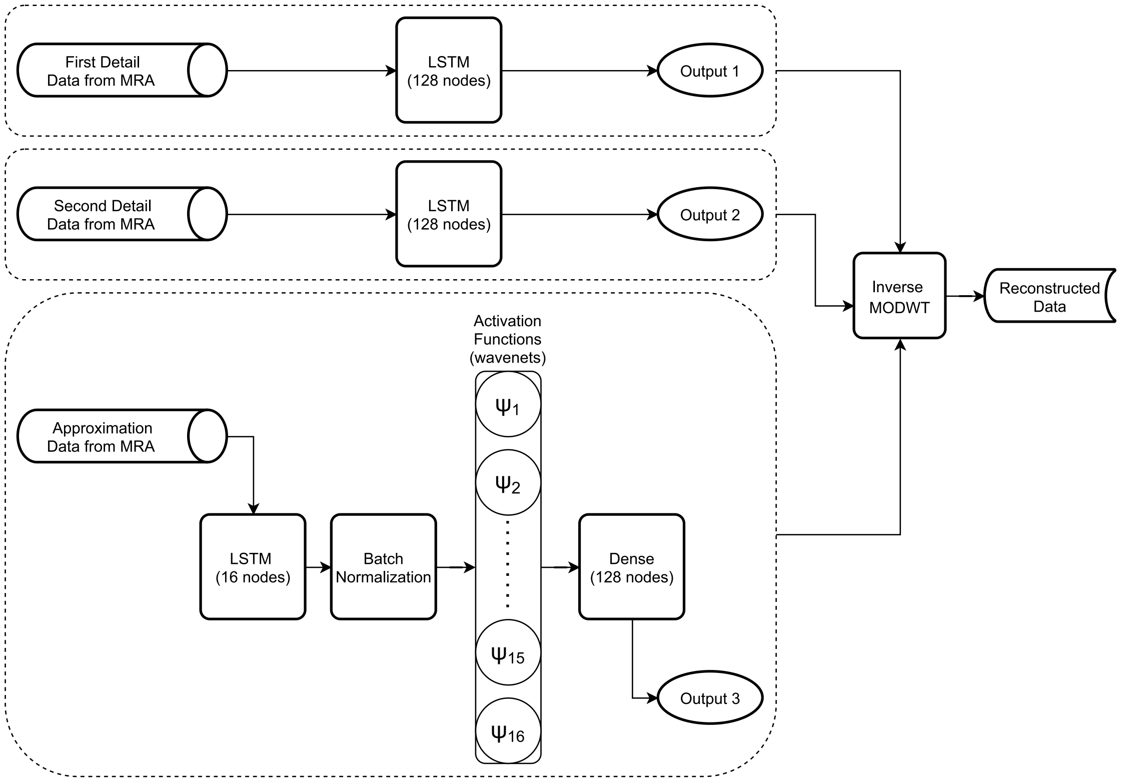


Fig. 4. The structure of the proposed hybrid LSTM-Wavenet with MRA model.

In Table A.17 picked Talos configuration parameters of the first detail, the second detail, and the approximation parts are given for S&P500 and NASDAQ. In this configuration, j is set to 6 in (31) for the approximation part. Wavenet structure is not used in the first and the second detail parts since the mentioned levels may be considered to be *noise*. The noise structure can be captured without the need for wavenets. In Table 7 average error scores of reconstructed train and test data are given for S&P500 and NASDAQ concerning this configuration 1.

In Table A.18 the second Talos configuration parameters of the first detail, the second detail, and the approximation parts are given for S&P500 and NASDAQ time series. At this point, the

Table 7
Hybrid LSTM-Wavenet model+MRA, configuration 1 (S&P500/NASDAQ): mean scores for the synthesized train set and the synthesized test set by running 1000 experiments.

Configuration 1			
Experiment size	1000	wavelet filter	db2 (D4)
Time taken by process	2103 m 30 s/2257 m 23 s	wavelet level	2
Monte Carlo Scores	Reconstructed Train Scores	Reconstructed Test Scores	
RMSE	8.04/18.41	17.93/40.62	
Scaled RMSE	0.00/0.00	0.01/0.00	
R^2	1.00/1.00	1.00/1.00	
MAE	6.56/12.02	15.17/32.21	
EVS	1.00/1.00	1.00/1.00	
ME	50.05/214.12	77.81/184.95	
MdAE	5.46/7.86	13.47/26.91	

Table 8
Hybrid LSTM-Wavenet model+MRA by API structure, configuration 2 (S&P500/NASDAQ): mean scores for the synthesized train set and the synthesized test set by running 1000 experiments.

Configuration 2			
Experiment size	1000	wavelet filter	db2 (D4)
Time taken by process	4318 m 36 s/4368 m 23 s	wavelet level	2
Monte Carlo Scores	Reconstructed Train Scores	Reconstructed Test Scores	
RMSE	7.45/18.59	17.96/41.97	
Scaled RMSE	0.00/0.00	0.01/0.01	
R^2	1.00/1.00	0.99/1.00	
MAE	5.81/11.84	15.18/33.24	
EVS	1.00/1.00	1.00/1.00	
ME	49.43/214.91	77.41/188.37	
MdAE	4.56/7.23	13.50/27.59	

second approach (configuration by API) mentioned in Section 3.4 is used to create activation functions. In Table 8 results are given for S&P500 and NASDAQ with respect to configuration 2.

Both configurations, which describe the hybrid LSTM-Wavenet model with MRA, outperform LSTM, LSTM+MRA, and hybrid LSTM-Wavenet methods. It is seen that configuration 1 is slightly better than configuration 2 when their respective RMSE, MAE, and MdAE metrics for both S&P500 and NASDAQ are compared.

4. Discussion of the results and comparison with classical methods

Table 9 summarizes the results obtained for S&P500 and NASDAQ. It should not be a surprise that using MRA improves the capability of both LSTM and hybrid LSTM-Wavenet models. Using different dilation and translation parameters for each activation function does not change the results significantly when compared to the case of using constant dilation and translation parameters for all nodes in LSTM.

The proposed method (hybrid LSTM-Wavenet+MRA) outperforms all the methods, LSTM, LSTM+MRA, and hybrid LSTM-Wavenet, for financial time series in terms of the error metrics. Hence, it is obvious that using wavelets in both MRA and activation functions improves the train and test performances.

In Section 3.1, it is declared that all training, validation, and test sets display diverse characteristics. The differences between the training sets and the test sets are relatively notable. Since such a difference occurs in the financial time series dynamics, it is understandable to view an absolute distinction between the RMSE values of the training set and the test set for both S&P500 and NASDAQ: RMSE error rates are relatively low compared to time series values. For instance, the mean of the training data is 654.61, and the mean of the test data is 2283.34 for S&P500. On the other hand, in the results obtained from the four methods, the maximum RMSE value for the training set is 15.29, and the minimum RMSE value is 7.45 for S&P500. Furthermore, in all results, the maximum RMSE is 29.06, and the minimum RMSE is 17.93 for the test set for S&P500. The mean of training data is 1147.35, and the mean of the test data is 5701.76 for NASDAQ data. The results belonging to the four methods show that the maximum RMSE value for the training set is 36.65, and the minimum RMSE value is 18.41 for NASDAQ. Additionally, the maximum RMSE is 85.31, and the minimum RMSE is 40.62, in the test set for NASDAQ.

If the SRMSE values are examined, it is seen that the error difference between training and test sets is small for both S&P500 and NASDAQ. The reason is that the time series observations are relatively high compared to the calculated RMSE values.

The error metrics R^2 and EVS show how well the model fits. RMSE, SRMSE, and MAE metrics generate average errors by using residuals. RMSE and SRMSE penalize large error values rather than other error metrics due to taking the square of residuals. RMSE and SRMSE are mostly used for model comparison. MdAE is quite robust against outliers. Nevertheless, it is not an advantage in our case, as it brings a drawback since large errors occur at big jumps in financial time series and the results of these large errors, i.e., outliers, might be significant.

In conclusion, nearly all error metrics generate robustness: each model trial with thousand repetitions gives very similar outcomes.

To sum up, LSTM+MRA, hybrid LSTM-Wavenet (configuration 1), and hybrid LSTM-Wavenet+MRA (configuration 1) methods provide, respectively, 7%, 22%, and 38% improvement in predictions based on RMSE values of LSTM for S&P500 as given in Table 9. If the MAE scores of S&P500 are considered, the respective improvements are 3%, 26%, and 35% based on LSTM. When the same comparisons for NASDAQ are made, the corresponding improvement values for RMSE are 20%, 24%, and 52% while the improvements for MAE are 21%, 30%, and 52%.

RMSE and MAE test error values of LSTM and hybrid LSTM-Wavenet methods decrease when MRA is used (for both S&P500 and NASDAQ). On the other hand, if wavelets are used as an activation function, then RMSE and MAE test error values for both LSTM and LSTM+MRA methods decrease for both time series. Consequently, the best test scores are reached by the hybrid LSTM-Wavenet+MRA method with fixed dilation and translation parameters for both S&P500 and NASDAQ data. The use of

Table 9
Summary table for results (S&P500/NASDAQ): mean scores for the train set and the test set by running 1000 experiments where 'conf.' stands for configuration.

	Time	RMSE	SRMSE	MAE	MdAE
LSTM	228 m 29 s/ 286 m 52 s				
Train scores		11.93/36.65	0.00/0.00	8.06/21.44	4.86/11.35
Test scores		29.06/85.31	0.01/0.01	23.30/66.58	19.93/53.22
LSTM+MRA	1981 m 47 s/ 2008 m 34 s				
Train scores		15.29/33.12	0.01/0.00	12.29/20.74	11.11/11.67
Test scores		26.91/68.41	0.01/0.01	22.66/52.67	20.36/44.05
Hybrid (conf. 2, by API)	2464 m 15 s/ 2379 m 44 s				
Train scores		9.58/30.70	0.00/0.00	6.29/16.94	3.77/8.53
Test scores		24.04/67.30	0.01/0.01	18.40/49.41	14.48/36.80
Hybrid (conf. 1)	459 m 45 s/ 391 m 24 s				
Train scores		9.71/30.15	0.00/0.00	6.37/16.18	3.76/7.54
Test scores		22.82/64.57	0.01/0.01	17.33/46.77	13.74/34.45
Hybrid+MRA (conf. 2, by API)	4318 m 36 s/ 4368 m 23 s				
Train scores		7.45/18.59	0.00/0.00	5.81/11.84	4.56/7.23
Test scores		17.96/41.97	0.01/0.01	15.18/33.24	13.50/27.59
Hybrid+MRA (conf. 1)	2103 m 30 s/ 2257 m 23 s				
Train scores		8.04/18.41	0.00/0.00	6.56/12.02	5.46/7.86
Test scores		17.93/40.62	0.01/0.00	15.17/32.21	13.47/26.91

wavelets in MRA and activation function effectively increases the performance for time series prediction. Even further, the use of wavelets for both MRA and activation functions improves the performance the most.

The proposed method contributes to the modeling of (financial) time series which are non-stationary, non-normal, noisy, and chaotic. Therefore, the recommended method is likely to be applied to other time series efficiently.

4.1. Comparison with other classical benchmark methods

Although the main aim of this study is to combine MRA and wavelet activation functions for LSTM, the usability of the proposed method is examined by comparing it with various classical state-of-the-art methods used in quantitative finance.

For comparison, the test data stated in Section 3.1 is used for selected methods. In addition, a one-step ahead prediction is considered, and no model is updated after a prediction. Because these conditions are the same for all LSTM experiments.

The methods used for comparison are Prophet (additive regression model released by Facebook[3]) for univariate (uni.) and multivariate (multi.) data, KNN, lightGBM, random forest, support vector regression (SVR) inference with radial basis function (RBF), Bayesian ridge regression, LASSO, XGBoost, and SARIMA. The SARIMA model takes into account NASDAQ data as an exogenous variable for S&P500 and vice versa. Depending on the size of the number of parameter combinations, model parameters were chosen using either a randomized search on hyperparameters or an exhaustive search across the estimator's given parameter values based on the metrics used, such as, R^2, negated mean square error, Akaike information criterion (AIC).

The proposed method has lower error than those the state-of-the-art methods provide for both S&P500 and NASDAQ data for the predictions made for test data on evaluation with metrics including RMSE and MAE. In addition, all methods in Table 9 have lower RMSE values than those of the state-of-the-art models in Table 10. Only the MAE values of the SARIMA model are lower than the corresponding values of the LSTM and LSTM+MRA models. These results demonstrate the strength and the superiority of the LSTM and its hybrid versions. For instance, considering SARIMA, which yields the best result in Table 10, the proposed method has 39% and 27% better results for S&P500 in terms of RMSE and MAE, respectively. Besides, more strikingly for NASDAQ, improvements in RMSE and MAE values are, respectively, 56% and 50%.

Table 10
Performance results of the classical state-of-the-art models on the test sets for S&P500/NASDAQ.

Models	RMSE	MAE	Models	RMSE	MAE
Prophet (uni.)	944.86/ 2488.17	888.83/ 2257.06	Prophet (multi.)	184.13/ 203.09	171.24/ 151.90
KNN	824.34/ 1450.49	707.92/ 980.72	Bayesian ridge	35.06/ 110.03	24.01/ 75.83
LightGBM	699.76/ 1598.99	600.37/ 1111.27	LASSO	35.06/ 110.03	24.01/ 75.83
Random forest	682.87/ 1446.07	580.80/ 960.10	XGBoost	35.04/ 109.87	23.91/ 75.26
SVR-RBF	411.50/ 675.93	402.87/ 501.43	SARIMA	29.27/ 91.32	20.70/ 64.64

5. Conclusion

Since there is a shortage of merging MRA and WNNs in the literature; the hybrid LSTM-Wavenet+MRA approach is a recommended method in this paper to overcome this shortage. The proposed hybridization is compared with the performances of LSTM, LSTM+MRA, and hybrid LSTM-Wavenet methods to make a one-step ahead prediction of S&P500 and NASDAQ. It is seen that the two different wavelet methodologies used increase the performances and the best performance is obtained when the

[3] https://facebook.github.io/prophet/

two wavelet techniques are used together. Moreover, comparison with other classical state-of-the-art methods also shows that our proposed method provides a significant improvement in predictions. According to the results obtained, it is noticed that practicing wavelets in modeling financial time series is essential and promising; furthermore, such modeling may be very appreciated by practitioners and investors in financial markets.

The proposed method (hybrid LSTM-Wavenet+MRA) provides an original contribution to the knowledge in time series analysis by combining wavenets and MRA. As a result, this study bridges the gaps between hybrid models using MRA and hybrid models with WNNs in the related literature.

Some future applications and extensions to the proposed hybridization methodology may be on *multi-step ahead predictions*, *dynamically updating the models after each prediction*, and *adaptive parameter selection in datasets with different characteristics*.

CRediT authorship contribution statement

Deniz Kenan Kılıç: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Ömür Uğur:** Conceptualization, Methodology, Validation, Formal analysis, Investigation, Data curation, Writing – review & editing, Visualization, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgment

All authors read and agreed to the published version of the manuscript.

Appendix. Configurations of the models

Components of the computing environment are listed in Table A.11 together with python and the packages used.

The error metrics used in this paper are given in Table A.12, where y_i is the ith observed value, $\hat{y}_i$ is the ith predicted value, $y_{\max}$ is the maximum observation, $y_{\min}$ is the minimum observation and $\bar{y}$ is the mean of all observations.

Table A.11
The list of the computing environment.

OS Platform	Windows 10
Processor	Intel(R) Core(TM) i5-4210H CPU 2.90 GHz
Memory (RAM)	8,00 GB
Conda version	4.7.11
Conda-build version	3.17.6
Python version	3.6.9
TensorFlow version	1.13.1
Keras version	2.2.4
Talos version	0.5.0

Table A.12
The list of the error metrics.

Root Mean Square Error (RMSE)	$\sqrt{\sum_{i=1}^{n} \frac{(y_i-\hat{y}_i)^2}{n}}$
Scaled Root Mean Square Error (SRMSE)	$\mathrm{RMSE}/(y_{\max} - y_{\min})$
R^2	$1 - \frac{\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}{\sum_{i=1}^{n}(y_i-\bar{y})^2}$
Mean Absolute Error (MAE)	$\frac{1}{n}\sum_{i=1}^{n}\lvert y_i - \hat{y}_i\rvert$
Explained Variance Score (EVS)	$1 - \frac{\mathrm{Var}(y-\hat{y})}{\mathrm{Var}(y)}$
Maximum Error (ME)	$\max\left(\lvert y_i - \hat{y}_i\rvert\right)$
Median Absolute Error (MdAE)	$\mathrm{median}\left(\lvert y_1 - \hat{y}_1\rvert, \ldots, \lvert y_n - \hat{y}_n\rvert\right)$

Table A.13
LSTM model, Talos configuration.

Configuration			
loss	mse		
optimizer	Adam		
time-steps	10		
batch size	1024		
epochs	100		
Layer parameters	LSTM layer	Between layers	Dense layer
# of nodes	16	–	128
kernel initializer	normal	–	–
batch normalization	–	yes	–
kernel regularizer	l1, l2 = 1e−6	–	l1, l2 = 1e−6
recurrent regularizer	l1, l2 = 1e−4	–	–
bias regularizer	l1, l2 = 1e−5	–	l1, l2 = 1e−5
activation	–	ReLU	ReLU

Table A.14
LSTM model+MRA, Talos configuration.

Configuration			
wavelet filter = db2 (D4)			
wavelet level = 2			
Detail 1		***Detail 2***	
loss	mse	loss	mse
optimizer	Adam	optimizer	Adam
time-steps	10	time-steps	10
batch size	1024	batch size	1024
epochs	50	epochs	50
Layer parameters	LSTM layer	***Layer parameters***	LSTM layer
# of nodes	128	# of nodes	128
kernel initializer	–	kernel initializer	–
batch normalization	–	batch normalization	–
kernel regularizer	–	kernel regularizer	–
recurrent regularizer	–	recurrent regularizer	–
bias regularizer	–	bias regularizer	–
activation	–	activation	–
Approximation			
loss	mse		
optimizer	Adam		
time-steps	10		
batch size	1024		
epochs	100		
Layer parameters	LSTM layer	Between Layers	Dense Layer
# of nodes	16	–	128
kernel initializer	normal	–	–
batch normalization	–	yes	–
kernel regularizer	l1, l2 = 1e−6	–	l1, l2 = 1e−6
recurrent regularizer	l1, l2 = 1e−4	–	–
bias regularizer	l1, l2 = 1e−5	–	l1, l2 = 1e−5
activation	–	ReLU	ReLU

Table A.15
Hybrid LSTM-Wavenet model, configuration 1: Talos configuration.

Configuration 1			
loss	mse		
optimizer	Adam		
time-steps	10		
batch size	1024		
epochs	100		
Layer parameters	LSTM layer	Between Layers	Dense Layer
# of nodes	16	–	128
kernel initializer	normal	–	–
batch normalization	–	yes	–
kernel regularizer	l1, l2 = 1e−6	–	l1, l2 = 1e−6
recurrent regularizer	l1, l2 = 1e−4	–	–
bias regularizer	l1, l2 = 1e−5	–	l1, l2 = 1e−5
activation	$\psi_6^6(x)$[a]	ReLU	ReLU

[a] $\psi_6^6(x) = \psi\left(\frac{x-240}{300}\right)$.

Table A.16
Hybrid LSTM-Wavenet model by API structure, configuration 2: Talos configuration.

Configuration 2			
loss	mse		
optimizer	Adam		
time-steps	10		
batch size	1024		
epochs	100		
Layer parameters	LSTM layer	Between Layers	Dense Layer
# of nodes	16 × 1	–	128
kernel initializer	normal	–	–
batch normalization	–	yes	–
kernel regularizer	l1, l2 = 1e−6	–	l1, l2 = 1e−6
recurrent regularizer	l1, l2 = 1e−4	–	–
bias regularizer	l1, l2 = 1e−5	–	l1, l2 = 1e−5
activation	$\psi_6^j(x)$[a]	ReLU	ReLU

[a] $\psi_6^j(x) = \psi\left(\frac{x-40j}{50j}\right)$ for $j = 1, \ldots, 16$.

The configuration parameters of the LSTM model are shown in Table A.13 for S&P500 and NASDAQ data. The network uses mean square error as a loss function, Adam as an optimizer for a batch size of 1024 and epochs of 100 where the length of the time-steps is 10. There are two hidden layers where the first one is the LSTM layer and the second one is the regular dense layer.

The configuration parameters of the LSTM+MRA model are given in Table A.14 for S&P500 and NASDAQ data. Data is decomposed into two detail and one approximation parts by applying the Daubechies wavelet filter db2. All network structures practices mean square error as a loss function and Adam as an optimizer. Loss functions, optimizers, lengths of time-steps, and batch sizes are mean square error, Adam, 10, and 1024, respectively. Differently, the number of epochs is halved and only one LSTM hidden layer is used in each detail part.

The configuration parameters of the hybrid LSTM-Wavenet model are shown in Table A.15 where the fixed wavelet activation function is used in all LSTM nodes. On the other hand, in Table A.16 different wavelet activation functions are applied for each LSTM node.

The hybrid LSTM-Wavenet+MRA approach uses the configuration parameters given in Table A.17 but has activation functions in the LSTM layer of the approximation part. Similar to the LSTM-Wavenet model, fixed wavelet activation function is used for each node in Table A.17. However, changing wavelet activation functions are used in Table A.18.

Table A.17
Hybrid LSTM-Wavenet model+MRA, configuration 1: Talos configuration.

Configuration 1			
wavelet filter = db2 (D4)			
wavelet level = 2			
Detail 1		***Detail 2***	
loss	mse	loss	mse
optimizer	Adam	optimizer	Adam
time-steps	10	time-steps	10
batch size	1024	batch size	1024
epochs	50	epochs	50
Layer parameters	LSTM layer	*Layer parameters*	LSTM layer
# of nodes	128	# of nodes	128
kernel initializer	–	kernel initializer	–
batch normalization	–	batch normalization	–
kernel regularizer	–	kernel regularizer	–
recurrent regularizer	–	recurrent regularizer	–
bias regularizer	–	bias regularizer	–
activation	–	activation	–
Approximation			
loss	mse		
optimizer	Adam		
time-steps	10		
batch size	1024		
epochs	100		
Layer parameters	LSTM layer	Between layers	Dense layer
# of nodes	16	–	128
kernel initializer	normal	–	–
batch normalization	–	yes	–
kernel regularizer	l1, l2 = 1e−6	–	l1, l2 = 1e−6
recurrent regularizer	l1, l2 = 1e−4	–	–
bias regularizer	l1, l2 = 1e−5	–	l1, l2 = 1e−5
activation	$\psi_6^6(x)$[a]	ReLU	ReLU

[a] $\psi_6^6(x) = \psi\left(\frac{x-240}{300}\right)$.

Table A.18
Hybrid LSTM-Wavenet model+MRA by API structure, configuration 2: Talos configuration.

Configuration 2			
wavelet filter = db2 (D4)			
wavelet level = 2			
Detail 1		***Detail 2***	
loss	mse	loss	mse
optimizer	Adam	optimizer	Adam
time-steps	10	time-steps	10
batch size	1024	batch size	1024
epochs	50	epochs	50
Layer parameters	LSTM layer	*Layer parameters*	LSTM layer
# of nodes	128	# of nodes	128
kernel initializer	–	kernel initializer	–
batch normalization	–	batch normalization	–
kernel regularizer	–	kernel regularizer	–
recurrent regularizer	–	recurrent regularizer	–
bias regularizer	–	bias regularizer	–
activation	–	activation	–
Approximation			
loss	mse		
optimizer	Adam		
time-steps	10		
batch size	1024		
epochs	100		
Layer parameters	LSTM layer	Between Layers	Dense Layer
# of nodes	16	–	128
kernel initializer	normal	–	–
batch normalization	–	yes	–
kernel regularizer	l1, l2 = 1e−6	–	l1, l2 = 1e−6
recurrent regularizer	l1, l2 = 1e−4	–	–
bias regularizer	l1, l2 = 1e−5	–	l1, l2 = 1e−5
activation	$\psi_6^j(x)$[a]	ReLU	ReLU

[a] $\psi_6^j(x) = \psi\left(\frac{x-40j}{50j}\right)$ for $j = 1, \ldots, 16$.

References

[1] D.K. Kılıç, Ö. Uğur, Multiresolution analysis of S&P500 time series, Ann. Oper. Res. 260 (1–2) (2018) 197–216, http://dx.doi.org/10.1007/s10479-016-2215-3.

[2] S.R. Karingula, N. Ramanan, R. Tahmasbi, M. Amjadi, D. Jung, R. Si, C. Thimmisetty, L.F. Polania, M. Sayer, J. Taylor, et al., Boosted embeddings for time-series forecasting, in: Machine Learning, Optimization, and Data Science: 7th International Conference, LOD 2021, Grasmere, UK, October 4–8, 2021, Revised Selected Papers, Part II, Springer, 2022, pp. 1–14, http://dx.doi.org/10.1007/978-3-030-95470-3_1.

[3] R. Wang, X. Pei, J. Zhu, Z. Zhang, X. Huang, J. Zhai, F. Zhang, Multivariable time series forecasting using model fusion, Inform. Sci. 585 (2022) 262–274, http://dx.doi.org/10.1016/j.ins.2021.11.025.

[4] Y. Ensafi, S.H. Amin, G. Zhang, B. Shah, Time-series forecasting of seasonal items sales using machine learning – A comparative analysis, Int. J. Inf. Manag. Data Insights 2 (1) (2022) 100058, http://dx.doi.org/10.1016/j.jjimei.2022.100058.

[5] R.V. Joseph, A. Mohanty, S. Tyagi, S. Mishra, S.K. Satapathy, S.N. Mohanty, A hybrid deep learning framework with CNN and Bi-directional LSTM for store item demand forecasting, Comput. Electr. Eng. 103 (2022) 108358, http://dx.doi.org/10.1016/j.compeleceng.2022.108358.

[6] Y. Li, J. Liu, Y. Teng, A decomposition-based memetic neural architecture search algorithm for univariate time series forecasting, Appl. Soft Comput. 130 (2022) 109714, http://dx.doi.org/10.1016/j.asoc.2022.109714.

[7] S.-X. Lv, L. Peng, H. Hu, L. Wang, Effective machine learning model combination based on selective ensemble strategy for time series forecasting, Inform. Sci. 612 (2022) 994–1023, http://dx.doi.org/10.1016/j.ins.2022.09.002.

[8] X. Wang, H. Liu, J. Du, X. Dong, Z. Yang, A long-term multivariate time series forecasting network combining series decomposition and convolutional neural networks, Appl. Soft Comput. 139 (2023) 110214, http://dx.doi.org/10.1016/j.asoc.2023.110214.

[9] B. Kumar, Sunil, N. Yadav, A novel hybrid model combining βSARMA and LSTM for time series forecasting, Appl. Soft Comput. 134 (2023) 110019, http://dx.doi.org/10.1016/j.asoc.2023.110019.

[10] H.V. Dudukcu, M. Taskiran, Z.G. Cam Taskiran, T. Yildirim, Temporal Convolutional Networks with RNN approach for chaotic time series prediction, Appl. Soft Comput. 133 (2023) 109945, http://dx.doi.org/10.1016/j.asoc.2022.109945.

[11] S.K. Sahu, A. Mokhade, N.D. Bokde, An overview of machine learning, deep learning, and reinforcement learning-based techniques in quantitative finance: recent progress and challenges, Appl. Sci. 13 (3) (2023) http://dx.doi.org/10.3390/app13031956.

[12] J.-S. Chou, N.-M. Nguyen, C.-P. Chang, Intelligent candlestick forecast system for financial time-series analysis using metaheuristics-optimized multi-output machine learning, Appl. Soft Comput. 130 (2022) 109642, http://dx.doi.org/10.1016/j.asoc.2022.109642.

[13] J. Sun, K. Xiao, C. Liu, W. Zhou, H. Xiong, Exploiting intra-day patterns for market shock prediction: A machine learning approach, Expert Syst. Appl. 127 (2019) 272–281, http://dx.doi.org/10.1016/j.eswa.2019.03.006.

[14] D. Kobiela, D. Krefta, W. Król, P. Weichbroth, ARIMA vs LSTM on NASDAQ stock exchange data, Procedia Comput. Sci. 207 (2022) 3836–3845, http://dx.doi.org/10.1016/j.procs.2022.09.445, Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 26th International Conference KES2022.

[15] I. Paliari, A. Karanikola, S. Kotsiantis, A comparison of the optimized LSTM, XGBOOST and ARIMA in Time Series forecasting, in: 2021 12th International Conference on Information, Intelligence, Systems & Applications, IISA, 2021, pp. 1–7, http://dx.doi.org/10.1109/IISA52424.2021.9555520.

[16] M. Yang, J. Wang, Adaptability of financial time series prediction based on BiLSTM, Procedia Comput. Sci. 199 (2022) 18–25, http://dx.doi.org/10.1016/j.procs.2022.01.003, The 8th International Conference on Information Technology and Quantitative Management (ITQM 2020 & 2021): Developing Global Digital Economy after COVID-19.

[17] U.M. Sirisha, M.C. Belavagi, G. Attigeri, Profit prediction using ARIMA, SARIMA and LSTM models in time series forecasting: a comparison, IEEE Access 10 (2022) 124715–124727, http://dx.doi.org/10.1109/ACCESS.2022.3224938.

[18] P. Theerthagiri, A.U. Ruby, Seasonal learning based ARIMA algorithm for prediction of Brent oil Price trends, Multimedia Tools Appl. (2023) 1–20, http://dx.doi.org/10.1007/s11042-023-14819-x.

[19] I.R. Parray, S.S. Khurana, M. Kumar, A.A. Altalbe, Time series data analysis of stock price movement using machine learning techniques, Soft Comput. 24 (2020) 16509–16517, http://dx.doi.org/10.1007/s00500-020-04957-x.

[20] A. Rastogi, A. Qais, A. Saxena, D. Sinha, Stock market prediction with lasso regression using technical analysis and time lag, in: 2021 6th International Conference for Convergence in Technology (I2CT), 2021, pp. 1–5, http://dx.doi.org/10.1109/I2CT51068.2021.9417935.

[21] S. Raubitzek, T. Neubauer, An exploratory study on the complexity and machine learning predictability of stock market data, Entropy 24 (3) (2022) http://dx.doi.org/10.3390/e24030332.

[22] J. Patel, S. Shah, P. Thakkar, K. Kotecha, Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques, Expert Syst. Appl. 42 (1) (2015) 259–268, http://dx.doi.org/10.1016/j.eswa.2014.07.040.

[23] W. Khan, U. Malik, M.A. Ghazanfar, M.A. Azam, K.H. Alyoubi, A.S. Alfakeeh, Predicting stock market trends using machine learning algorithms via public sentiment and political situation analysis, Soft Comput. 24 (2020) 11019–11043, http://dx.doi.org/10.1007/s00500-019-04347-y.

[24] M. Nabipour, P. Nayyeri, H. Jabani, A. Mosavi, E. Salwana, S. Shahab, Deep learning for stock market prediction, Entropy 22 (8) (2020) http://dx.doi.org/10.3390/e22080840.

[25] N. Naik, B.R. Mohan, Stock price movements classification using machine and deep learning techniques-the case study of indian stock market, in: Engineering Applications of Neural Networks: 20th International Conference, EANN 2019, Xersonisos, Crete, Greece, May 24–26, 2019, Proceedings 20, Springer, 2019, pp. 445–452, http://dx.doi.org/10.1007/978-3-030-20257-6_38.

[26] K.K. Yun, S.W. Yoon, D. Won, Prediction of stock price direction using a hybrid GA-XGBoost algorithm with a three-stage feature engineering process, Expert Syst. Appl. 186 (2021) 115716, http://dx.doi.org/10.1016/j.eswa.2021.115716.

[27] T. Bashir, C. Haoyong, M.F. Tahir, Z. Liqiang, Short term electricity load forecasting using hybrid prophet-LSTM model optimized by BPNN, Energy Rep. 8 (2022) 1678–1686, http://dx.doi.org/10.1016/j.egyr.2021.12.067.

[28] X. Zhong, D. Enke, Predicting the daily return direction of the stock market using hybrid machine learning algorithms, Financ. Innov. 5 (1) (2019) 1–20, http://dx.doi.org/10.1186/s40854-019-0138-0.

[29] S.R. Das, D. Mishra, M. Rout, Stock market prediction using firefly algorithm with evolutionary framework optimized feature reduction for OSELM method, Expert Syst. Appl.: X 4 (2019) 100016, http://dx.doi.org/10.1016/j.eswax.2019.100016.

[30] E. Guresen, G. Kayakutlu, T.U. Daim, Using artificial neural network models in stock market index prediction, Expert Syst. Appl. 38 (8) (2011) 10389–10397, http://dx.doi.org/10.1016/j.eswa.2011.02.068.

[31] C. Yang, J. Zhai, G. Tao, et al., Deep learning for price movement prediction using convolutional neural network and long short-term memory, Math. Probl. Eng. 2020 (2020) http://dx.doi.org/10.1155/2020/2746845.

[32] I.E. Livieris, E. Pintelas, P. Pintelas, A CNN–LSTM model for gold price time-series forecasting, Neural Comput. Appl. 32 (2020) 17351–17360, http://dx.doi.org/10.1007/s00521-020-04867-x.

[33] G. Ding, L. Qin, Study on the prediction of stock price based on the associated network model of LSTM, Int. J. Mach. Learn. Cybern. 11 (2020) 1307–1317, http://dx.doi.org/10.1007/s13042-019-01041-1.

[34] X. Pang, Y. Zhou, P. Wang, W. Lin, V. Chang, An innovative neural network approach for stock market prediction, J. Supercomput. 76 (2020) 2098–2118, http://dx.doi.org/10.1007/s11227-017-2228-y.

[35] C. Luo, L. Pan, B. Chen, H. Xu, et al., Bitcoin price forecasting: an integrated approach using hybrid LSTM-ELM models, Math. Probl. Eng. 2022 (2022) http://dx.doi.org/10.1155/2022/2126518.

[36] K. He, Q. Yang, L. Ji, J. Pan, Y. Zou, Financial time series forecasting with the deep learning ensemble model, Mathematics 11 (4) (2023) http://dx.doi.org/10.3390/math11041054.

[37] P. Masset, Analysis of financial time-series using fourier and wavelet methods, Univ. Fribourg (2008) http://dx.doi.org/10.2139/ssrn.1289420.

[38] K.K. Teo, L. Wang, Z. Lin, Wavelet packet multi-layer perceptron for chaotic time series prediction: effects of weight initialization, in: Computational Science - ICCS 2001, Springer, 2001, pp. 310–317, http://dx.doi.org/10.1007/3-540-45718-6_35.

[39] D. Jothimani, R. Shankar, S.S. Yadav, Discrete wavelet transform-based prediction of stock index: a study on national stock exchange fifty index, J. Financ. Manag. Anal. 28 (2) (2015) 35–49, URL https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2769529.

[40] S.K. Chandar, M. Sumathi, S.N. Sivanandam, Prediction of stock market price using hybrid of wavelet transform and artificial neural network, Indian J. Sci. Technol. 9 (8) (2016) 1–5, http://dx.doi.org/10.17485/ijst/2016/v9i8/87905.

[41] J. Jin, J. Kim, Forecasting natural gas prices using wavelets, time series, and artificial neural networks, PLoS ONE 10 (11) (2015) http://dx.doi.org/10.1371/journal.pone.0142064.

[42] J. Wang, Z. Wang, J. Li, J. Wu, Multilevel wavelet decomposition network for interpretable time series analysis, in: KDD, 2018, pp. 2437–2446, http://dx.doi.org/10.1145/3219819.3220060.

[43] A. Arévalo, J. Nino, D. León, G. Hernandez, J. Sandoval, Deep learning and wavelets for high-frequency price forecasting, in: Computational Science – ICCS 2018, Springer International Publishing, 2018, pp. 385–399, http://dx.doi.org/10.1007/978-3-319-93701-4_29.

[44] Y. Liu, L. Guan, C. Hou, H. Han, Z. Liu, Y. Sun, M. Zheng, Wind power short-term prediction based on LSTM and discrete wavelet transform, Appl. Sci. 9 (6) (2019) http://dx.doi.org/10.3390/app9061108.

[45] C. Napoli, G. De Magistris, C. Ciancarelli, F. Corallo, F. Russo, D. Nardi, Exploiting wavelet recurrent neural networks for satellite telemetry data modeling, prediction and control, Expert Syst. Appl. 206 (2022) 117831, http://dx.doi.org/10.1016/j.eswa.2022.117831.

[46] S. Momeneh, V. Nourani, Application of a novel technique of the multi-discrete wavelet transforms in hybrid with artificial neural network to forecast the daily and monthly streamflow, Model. Earth Syst. Environ. 8 (4) (2022) 4629–4648, http://dx.doi.org/10.1007/s40808-022-01387-6.

[47] Y. Li, L. Xu, S. Ying, DWNN: deep wavelet neural network for solving partial differential equations, Mathematics 10 (12) (2022) http://dx.doi.org/10.3390/math10121976.

[48] Y. Lin, K. Chen, X. Zhang, B. Tan, Q. Lu, Forecasting crude oil futures prices using BiLSTM-Attention-CNN model with Wavelet transform, Appl. Soft Comput. 130 (2022) 109723, http://dx.doi.org/10.1016/j.asoc.2022.109723.

[49] Z. Pu, J. Yan, L. Chen, Z. Li, W. Tian, T. Tao, K. Xin, A hybrid Wavelet-CNN-LSTM deep learning model for short-term urban water demand forecasting, Front. Environ. Sci. Eng. 17 (2) (2023) http://dx.doi.org/10.1007/s11783-023-1622-3.

[50] F. Alenezi, A. Armghan, K. Polat, Wavelet transform based deep residual neural network and ReLU based Extreme Learning Machine for skin lesion classification, Expert Syst. Appl. 213 (2023) 119064, http://dx.doi.org/10.1016/j.eswa.2022.119064.

[51] J.D. Ramirez-Zamora, O.A. Dominguez-Ramirez, L.E. Ramos-Velasco, G. Sepulveda-Cervantes, V. Parra-Vega, A. Jarillo-Silva, E.A. Escotto-Cordova, HRpI system based on wavenet controller with human cooperative-in-the-loop for neurorehabilitation purposes, Sensors 22 (20) (2022) http://dx.doi.org/10.3390/s22207729.

[52] S. Deepa, A. Banerjee, Intelligent neural learning models for multi-step wind speed forecasting in renewable energy applications, J. Control Autom. Electr. Syst. 33 (3) (2022) 881–900, http://dx.doi.org/10.1007/s40313-021-00862-2.

[53] A. Kaushik, N. Singal, A hybrid model of wavelet neural network and metaheuristic algorithm for software development effort estimation, Int. J. Inf. Technol. (Singapore) 14 (3) (2022) 1689–1698, http://dx.doi.org/10.1007/s41870-019-00339-1.

[54] S. Yahia, S. Said, M. Zaied, Wavelet extreme learning machine and deep learning for data classification, Neurocomputing 470 (2022) 280–289, http://dx.doi.org/10.1016/j.neucom.2020.04.158.

[55] A.R. Sadri, T. DeSilvio, P. Chirra, S. Singh, S.E. Viswanath, Residual wavelon convolutional networks for characterization of disease response on MRI, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 13433 LNCS, 2022, pp. 366–375, http://dx.doi.org/10.1007/978-3-031-16437-8_35.

[56] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (1997) 1735–1780, http://dx.doi.org/10.1162/neco.1997.9.8.1735.

[57] V. Viswanath, Deep learning - ANN, RNN, LSTM networks, 2022, https://vishnuviswanath.com/ann_rnn_lstm.html. (Accessed 17 November 2022).

[58] C. Olah, Understanding LSTM networks, 2022, http://colah.github.io/posts/2015-08-Understanding-LSTMs/. (Accessed 17 November 2022).

[59] R. Gençay, F. Selçuk, B. Whitcher, An Introduction to Wavelets and Other Filtering Methods in Finance and Economics, Academic Press, San Diego, 2002, http://dx.doi.org/10.1016/B978-0-12-279670-8.X5000-9.

[60] D.B. Percival, A.T. Walden, Wavelet Methods for Time Series Analysis, Cambridge University Press, 2000, http://dx.doi.org/10.1017/CBO9780511841040.

[61] P.Z. Fryzlewicz, Wavelet Techniques for Time Series and Poisson Data (Ph.D. thesis), The University of Bristol, 2003.

[62] T.A. Vuorenmaa, A Multiresolution Analysis of Stock Market Volatility Using Wavelet Methodology (Licentiate thesis), Universtiy of Helsinki, 2004.

[63] B. Videkovic, Basics of wavelets, 2022, http://www2.isye.gatech.edu/~brani/isyebayes/bank/handout20.pdf. (Accessed 20 October 2022).

[64] S.G. Mallat, A theory for multiresolution signal decomposition: The wavelet representation, IEEE Trans. Pattern Anal. Mach. Intell. 11 (7) (1989) 674–693, http://dx.doi.org/10.1109/34.192463.

[65] N. Bhatnagar, Introduction to Wavelet Transforms, CRC Press, 2020, http://dx.doi.org/10.1201/9781003006626.

[66] L. Iliadis, I. Maglogiannis, G. Tsoumakas, I. Vlahavas, M. Bramer, Artificial intelligence applications and innovations: proceedings of the 5th IFIP conference on artificial intelligence applications and innovations (AIAI'2009), April 23–25, 2009, Thessaloniki, Greece, in: IFIP Advances in Information and Communication Technology, Springer US, 2009, http://dx.doi.org/10.1007/978-1-4419-0221-4.

[67] T. Zheng, K. Fataliyev, L. Wang, Wavelet neural networks for stock trading, in: Proceedings of SPIE - the International Society for Optical Engineering, Vol. 8750, 2013, p. 87500A, http://dx.doi.org/10.1117/12.2018040.

[68] X. Wen, H. Zhang, F. Wang, A wavelet neural network for SAR image segmentation, Sensors 9 (9) (2009) 7509–7515, http://dx.doi.org/10.3390/s90907509.

[69] R. Benali, F. Bereksi Reguig, Z. Hadj Slimane, Automatic classification of heartbeats using wavelet neural network, J. Med. Syst. 36 (2012) 883–892, http://dx.doi.org/10.1007/s10916-010-9551-7.

[70] D. Veitch, Wavelet Neural Networks and Their Application in the Study of Dynamical Systems (Master's thesis), Universtiy of York, 2005.

[71] J.F. Marar, A. Bordin, Multidimensional wavelet neural networks Based on polynomial powers of sigmoid: A framework to image verification, Transdiscipl.: Proj. Mater. Process. 1 (2016) 106–123, http://dx.doi.org/10.29147/2526-1789.DAT.2016v1i2p106-123.

[72] J. Fernando Marar, E.C.B. Carvalho Filho, W. Li, L. Deane Sa, Activation function study for wavelet network, Appl. Sci. Artif. Neural Netw. III 3077 (1997) 690–697, http://dx.doi.org/10.1117/12.271531.

[73] A.A. Minai, R.D. Williams, On the derivatives of the sigmoid, Neural Netw. 6 (1993) 845–853, http://dx.doi.org/10.1016/S0893-6080(05)80129-7.

[74] C. Bergström, O. Hjelm, Impact of Time Steps on Stock Market Prediction with LSTM, Tech. Rep. 2019:292, KTH, School of Electrical Engineering and Computer Science (EECS), 2019, p. 10, URL http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-262221.

[75] M. Kotila, et al., Autonomio talos [computer software], TALOS: hyperparameter experiments with tensorflow, PyTorch and keras, 2019, http://github.com/autonomio/talos.

[76] F. Chollet, et al., Keras, 2015, https://keras.io.

[77] J. Brownlee, How to accelerate learning of deep neural networks with batch normalization, 2022, https://machinelearningmastery.com/how-to-accelerate-learning-of-deep-neural-networks-with-batch-normalization/. (Accessed 19 November 2022).

[78] J. Collis, Glossary of deep learning: batch normalisation, 2022, https://medium.com/deeper-learning/glossary-of-deep-learning-batch-normalisation-8266dcd2fa82. (Accessed 14 November 2022).