AIMS Mathematics, 9(4): 9419–9434.
DOI: 10.3934/math.2024459
Received: 16 January 2024
Revised: 22 February 2024
Accepted: 29 February 2024
Published: 07 March 2024

http://www.aimspress.com/journal/Math

Research article

EMDFormer model for time series forecasting

Ana Lazcano de Rojas[1,*], Miguel A. Jaramillo-Morán[2] and Julio E. Sandubete[1]

[1] Universidad Francisco de Vitoria, Faculty of Law and Business, Spain
[2] University of Extremadura, Department of Electrical Engineering, Electronics and Automation, School of Industrial Engineering, Spain

* **Correspondence:** Email: ana.lazcano@ufv.es.

Abstract: The adjusted precision of economic values is essential in the global economy. In recent years, researchers have increased their interest in making accurate predictions in this type of time series; one of the reasons is that the characteristics of this type of time series makes predicting a complicated task due to its non-linear nature. The evolution of artificial neural network models enables us to research the suitability of models generated for other purposes, applying their potential to time series prediction with promising results. Specifically, in this field, the application of transformer models is assuming an innovative approach with great results. To improve the performance of this type of networks, in this work, the empirical model decomposition (EMD) methodology was used as data preprocessing for prediction with a transformer type network. The results confirmed a better performance of this approach compared to networks widely used in this field, the bidirectional long short term memory (BiLSTM), and long short term memory (LSTM) networks using and without EMD preprocessing, as well as the comparison of a Transformer network without applying EMD to the data, with a lower error in all the error metrics used: The root mean square error (RMSE), the root mean square error (MSE), the mean absolute percentage error (MAPE), and the R-square (R^2). Finding a model that provides results that improve the literature allows for a greater adjustment in the predictions with minimal preprocessing.

Keywords: time series forecasting; financial forecasting; recurrent neural network; LSTM BiLSTM; transformer; EMD
Mathematics Subject Classification: 68T07, 91B84

1. Introduction

Time series forecasting has become a key issue in research fields related to energy production and economic or financial analysis due to the need to know the time evolution of the variables under study. Many forecasting tools have been developed along time to deal with the prediction of future values of those variables. They can be roughly classified into two classes [1]: statistical methods and those based on artificial intelligence. The former is usually known as classical techniques since they provide the first tools to process time series [2]. The most widely used tool is the autoregressive moving integrated moving average (ARIMA). This tool has been widely used to predict time series of energy-related variables [3,4]; nevertheless, as this model has a linear structure, it has problems when dealing with hard nonlinear time series.

Although statistical models were able to provide good predictions with many time series, tools based on artificial intelligence are more widely used in this field because of their ability to deal with highly nonlinear time series. There are several tools in this field that have been used as forecasting tools: random forest, gradient and extreme gradient boosting [5], or support vector machine [6]. Among them, artificial neural networks (ANN) clearly stand out, and they have been widely used to forecast variables related to economy or energy, becoming one of the most popular forecasting methods in these fields.

Many works have been published where ANN clearly outperform classical tools. In [7], the behavior of ANN was compared with six traditional statistical methods for predicting time series, pointing out a better performance of neural networks compared to the other techniques. In [8], the existing literature was completed with new studies using a novel model that combines traditional statistical techniques and ANN, obtaining empirical results that show it as an effective way to improve the accuracy of time series prediction. There are lots of works where ANN has facilitated complex time series prediction tasks with results that improve classical statistical techniques [9–11], even by incorporating hybrid models, such as the one presented in [12], in which a recurrent hybrid model was used for the prediction of time series of different types with results that improve those obtained with other models.

Many of these applications use a simple neural model, multilayer perceptron (MLP) [13–15], because of its ability to approximate any measurable function [16], even though they are presented as a nonlinear time series or there are noisy or missing data.

New complex neural structures, capable of processing large amounts of data with strong temporal relationships between them, have been developed to address problems closely related to human abilities, such as text and speech processing or object identification. Due to the high number of layers and neurons they have, they are usually known as deep learning neural networks. One of these structures is the long short-term memory (LSTM), originally proposed to process written text or speech. Since these problems present a high time dependence among the data they process, it seemed logical to assume that they could also provide accurate predictions in time series forecasting, and thus they have been widely used to do so [17,18]. Although this model has been able to provide good results, a new structure based on it has been proposed to try to improve its performance: The bidirectional long short-term memory network (BiLSTM), which, unlike traditional LSTM networks, executes additional training by traversing the input twice from left to right and then from right to left. In [19], the possibility of incorporating more additional layers during the training phase was explored, and, as a result, the BiLSTM model provided greater efficiency compared to the results obtained by an LSTM. Along the same lines, [20] made predictions from financial time series with similar results. Similarly, [21] reached the same conclusion by comparing the BiLSTM, support vector regression, and ARIMA

models in forecasting economic time series.

In recent years, transformer neural networks (TNNs) have been presented as a revolutionary alternative in the field of prediction, although they were originally developed for applications such as natural language processing and computer vision. Their introduction has meant a change in the way in which sequential problems are addressed, with research on the application of TNNs to time series prediction accelerating in recent years, demonstrating their efficiency in modeling complex temporal patterns. Its novelty lies in its ability to handle sequences of variable length and to capture complex temporal relationships more effectively than recurrent neural networks (RNNs), such as LSTM, eliminating the need to maintain long-term memory thanks to the incorporation of attention mechanisms. In [22], it was mentioned how RNNs were proposed as an effective alternative in which despite the appearance of different variants such as LSTM, it was difficult to capture long-term dependencies in the time series data. Unlike RNNs, Transformers networks allow the model to access any part of the history regardless of distance, making it potentially suitable for capturing recurring patterns with long-term dependencies.

Transformers have significantly increased their use for time series forecasting tasks [23], as they appear to be more successful at extracting complex correlations between data than other models used for these tasks, such as LSTM. To date, much superior performance has been demonstrated in many natural language processing and computer vision tasks [24,25], a fact that has sparked researchers' interest in using this type of network for time series forecasting [26,27]. Some works have also tested the performance of variations of the basic Transformer model by comparing their performance with other forecasting tools [28].

Other ANN models have also been used to forecast time series although they have not deserved the same attention as those described above. Some works [29,30] have used the convolutional neural network (CNN) to forecast time series with good performance. Gated recurrent gates (GRU), a simplification of the basic LSTM structure, have also been used [31,32]. Even though CNN and GRU have been able to provide better performance than other neural models when forecasting specific time series, they have not been as widely used as the previously mentioned models because they are case-oriented tools; CNN for image processing and GRU for text processing, which makes them less adapted to the time series forecasting task. Among the most recent models studied for time series prediction are those of graph convolutional networks (GCN), showing their effectiveness in taking advantage of the relationships between the data and representing each time series as if it were a graph (see [33,34]).

In addition to the use of new forecasting models, new more sophisticated structures have been proposed. They combine two or more forecasting tools trying to achieve more accurate predictions. Thus, we can find in literature several combinations of forecasting models: CNN-LSTM [35], LSTM-MLP-extreme gradient boosting [36], ARIMA-CNN-LSTM [37], and CCNN-MLP-transformers [38]. Although all these combinations were able to perform well with the time series they were designed to handle, it is not clear that they would be able to overcome other simpler models with time series different from those they were used with.

Following the strategy of combining several forecasting tools in a hybrid model, some authors have proposed an alternative way to improve the accuracy of predictions: preprocessing of the time series to be forecast. Apart from applying statistical tools to improve the quality of these data to facilitate forecasting [39], the time series could be decomposed into sub-series with a uniform behavior, making them easier to forecast. This assumption is because many energy or economic variables are influenced by social and weather factors that have a certain periodic behavior. Therefore, the objective should be to decompose the time series into sub-series that are closely related to those periodic

behaviors. In this way, each of these sub-series should retain a certain periodic behavior closely related to specific frequencies embedded in the overall behavior of the time series; thus, they could be more accurately predicted. Empirical mode decomposition (EMD) is one of the techniques that has attracted the attention of many researchers because it has been able to improve the performance of the forecasting tools with which it has been used. It has been applied to several neural models such as MLP [40], LSTM [40,41], or Transformers [42]. In [43], an LSTM was combined with two attention mechanisms to process a time series previously decomposed with a complete ensemble EMD with adaptive noise (CEEMDAN) to forecast several datasets. This last variation of the basic EMD model has also been used with transformers [26]. All these works showed that the models with preprocessing were able to outperform those without it.

Since EMD has been shown to significantly improve the performance of neural networks for time series forecasting, it is used in this work along with a transformer to forecast economic and energy-related time series. The aim is to prove that the combination of a preprocessing stage with a forecasting neural network can provide a better performance than the forecasting tool alone. It is worth noting that the basic structure of both tools has been used instead of more sophisticated modifications proposed in the literature with the aim of also showing that it is not necessary to use very sophisticated models to achieve good performance. The transformer was chosen as the forecasting tool because it is being tested to find out whether it can outperform other neural models in these tasks. In this work, it will be demonstrated that it can outperform both LSTM and BiLSTM, tools that have been widely used for these tasks.

This paper is organized as follows: Section 2 provides a detailed description of the methodology and its architecture. Section 3 focuses on the experiments carried out and the comparison of the results. Finally, Section 4 describes the major conclusions and future lines of work.

2. Methodology

In this work, a Transformer is used with an EMD to forecast time series, defining a unique forecasting tool called EMDFormer. Its performance will be compared with that of the transformer without EMD. Two other neural models, LSTM and BiLSTM, are also tested with and without EMD for comparison. The proposed EMDFormer model is described below.

The choice of the BiLSTM and LSTM models to perform a contrast is due to the results reflected in the existing literature, which reflect truly adjusted predictions, which are essential in this type of time series.

Zhao et al. [44] describe how LSTM networks treat the hidden layer as a memory unit so that it can cope with correlation within both short-term and long-term time series.

2.1. EMD Model

EMD decomposes the original data into a collection of intrinsic mode functions (IMFs) and a residual based on the local characters of the time series, including maximums, minimums, and zero crossings [45]. IMFs must meet two conditions:

(1) The number of extremes and the number of zero crossings must differ by a maximum of one.

(2) The mean value of the envelope defined by local maxima and the envelope defined by local minima must be zero at any point.

The decomposition method is called sieving process [46] and is described as follows:

(1) Given a time series x_t, identify all the maximums and minimums and form with all the

maximums an envelope u_t, and with the minimums a lower envelope l_t using an interpolation technique.

(2) Calculate the average value of the maximum and minimum envelopes.

$$m_t = \frac{u_t - l_t}{2}. \tag{1}$$

(3) Calculate the difference between x_t and m_t to obtain a detailed component $h_t = x_{xt} - m_t$.

(4) Steps 1–3 will be repeated with h_t as the new input until it satisfies the two conditions mentioned above or the number of iterations reaches the maximum defined by the user. Now h_t will be defined as the first IMF as c_{1t}.

(5) Subtracting c_{1t} from x_t a new sequence will be obtained without the high-frequency components $r_{1t} = x_t - c_{1t}$.

(6) The process will be repeated until all the IMFs and a residue are obtained.

(7) In this way the original time series is decomposed as:

$$x_t = \sum_i c_{it} + r_t. \tag{2}$$

2.2. Transformer

Transformers neural networks are a deep learning architecture based on attention mechanisms. [23] introduced the scaled dot-product attention algorithm as a goal to ensure that models were able to focus only on the most relevant elements of long sequences. To achieve this, it is necessary to obtain the weighted sum of the values V, where the weights are calculated by applying the softmax function to the scalar products of the queries Q with the keys K, scaled by the square root of the dimension of the keys d_k.

$$ttention(Q, K, V) = softmax\left(\frac{QK^t}{\sqrt{d_k}}\right)V. \tag{3}$$

Transformers use a variant of this algorithm called multi-head attention. This version uses h learnable linear projections to the queries, keys, and values before applying individual attention to each of the projections. After this step, the result obtained in each of the attentions will be concatenated before the last linear projection.

The input at each time step will first be transferred by an embedding layer to a vector, which consists of a representation of information in a high-dimensional space. The vector will then be combined with the positional information to be the input to the multi-head attention layer (Figure 1). For each attention head, 3 parameter matrices are generated for learning the transformer: the key weights WK, the query to be keyed weights W_Q, and the value weights W_V. The embedding X is multiplied with the above 3 matrices to get the key matrix K, query matrix Q and value matrix V [47].

A tuple of the weight matrix (W_k,W_q,W_v) is called the attention head, and in a multi-head attention layer, there are several heads. As seen in the figure, the results of each head will be added and normalized to move to the next layer. The feedforward layer is the weight matrix that is trained during training and applied to each respective time step position.

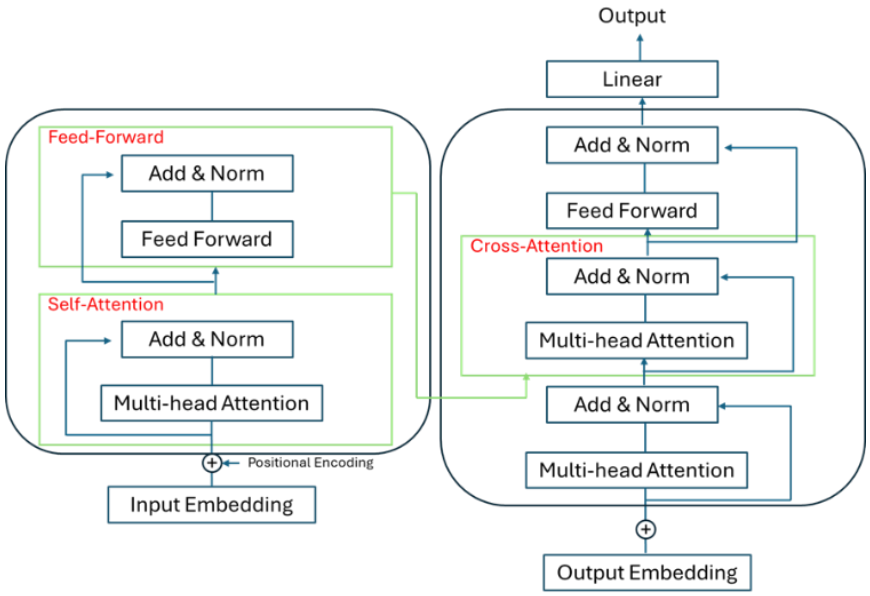

Figure 1. Transformer artificial neural network architecture.

2.3. EMDFormer

The proposed model combines the predictions obtained from the different IMF's generated from the application of EMD to the time series to be studied. The process flow is represented in Figure 2.

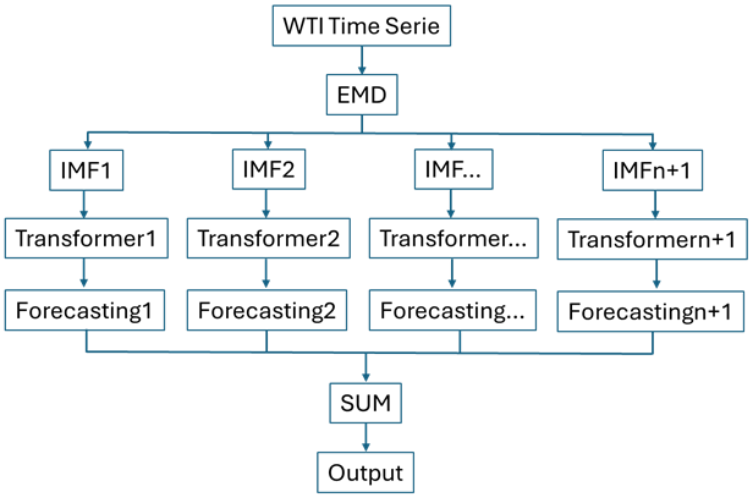

Figure 2. EMDFormer model process flow.

Initially, the EMD is applied to the time series, and this process will result in 11 IMF's, represented in Figure 3, which will subsequently be the input of the transformer neural network.

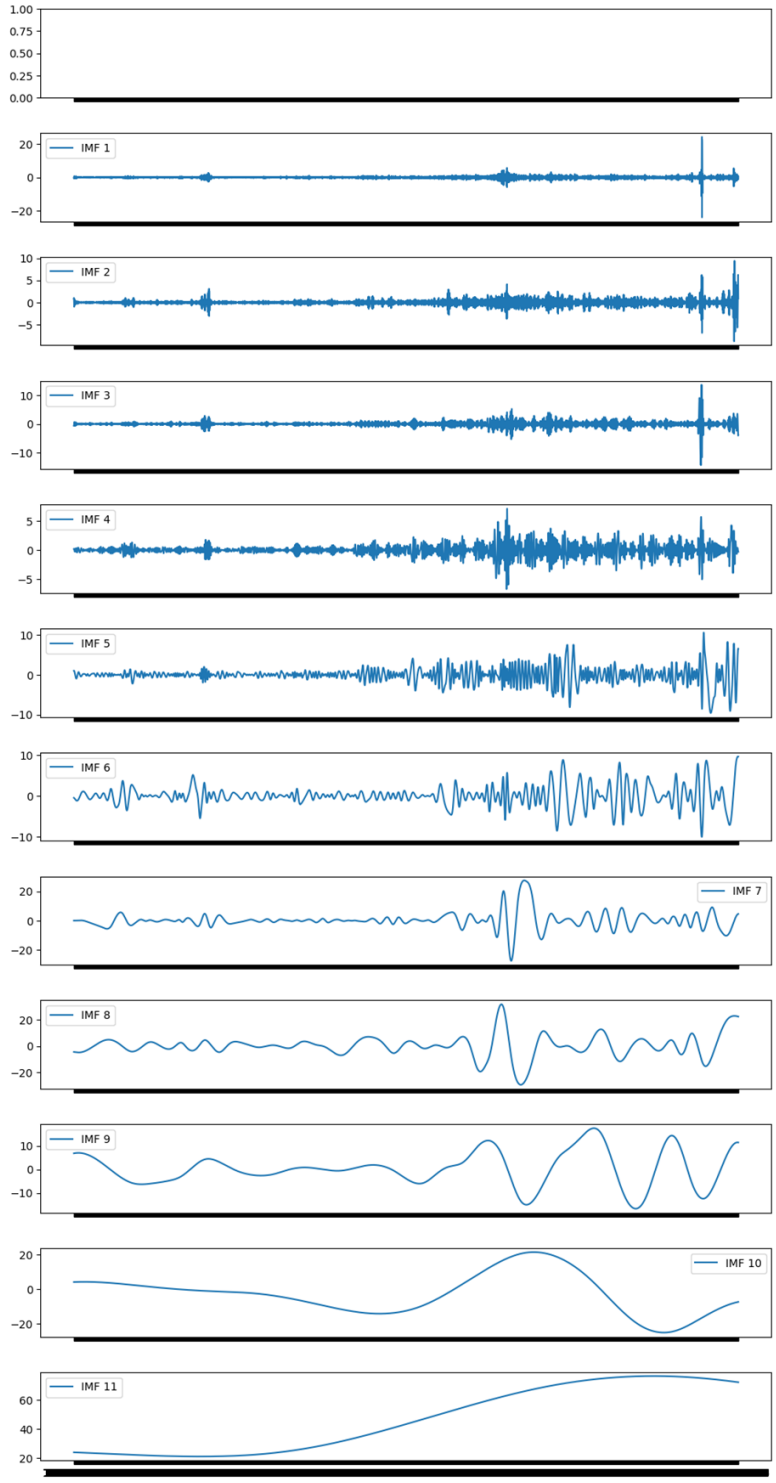

Figure 3. IMF's obtained.

3. Experiments

3.1. Data description

Next, the data used in the evaluation process of the TNNs will be described by predicting the resulting IMFs after the EMD process. To test the robustness of the proposed model, the experiments

are carried out with two-time series; the first contains the data related to the west texas intermediate (WTI) crude price index obtained through the Thomson Eikon Reuters platform, and the data is in a temporal daily form from January 10, 1983 to June 15, 2022. WTI prices are the most widely used spot price of oil, along with Brent spot prices as a reference to set the price of oil.

The total number of observations analyzed is 10,289, included in a DataFrame from the Pandas library in Python.

The second time series used is the Bloomberg commodities total return (BCTR) commodity index on a daily basis, from January 2, 1991 to May 25, 2022. The data were obtained from Thomson's Eikon database, Reuters. The index represents certain commodities related to energy, livestock, soft commodities, industrial metals, precious metals, and grains.

The forecast is made on the 8,192 observations collected in the indicated interval and is imported into a Python DataFrame generated with the Pandas library.

3.2. Evaluation metrics

In order to evaluate the performance of the different experiments, the following error metrics are used:

Root mean squared error (RMSE):

$$MSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_t - \widehat{x_t})}, \tag{4}$$

mean squared error (MSE):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(x_t - \widehat{x_t})^2, \tag{5}$$

mean absolute percentage error (MAPE):

$$MAPE = \frac{\sum_{t=1}^{n}\frac{|x_t - \widehat{x_t}|}{|x_t|}}{n}, \tag{6}$$

R-squared (R^2):

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(x_t - \widehat{x_t})^2}{\sum_{i=1}^{n}(x_t - \bar{x})^2}. \tag{7}$$

In these expressions, x_t is the i-th element of the original time series, $\widehat{x_t}$ is its corresponding forecasted value, $\bar{x}$ is the mean value, and n is the number of elements it has. The MAPE and RMSE metrics are used to measure the error in the predictions, which will allow us to know the adjustment of the neural network; thus, the value should be as low as possible. Frechtling [48] announced that the optimal value for the MAPE metric should be between 10% and 20%, where an optimal prediction between 20 and 30 will be considered acceptable.

The R^2 metric is like MSE and will be used to evaluate the performance of the model. It will reflect the variation in the dependent variable that occurs from the prediction of the independent variable. The closer to 1, the better the network performance.

3.3. Model parameters

The best performance of the model will be obtained by adjusting the hyperparameters of the network; tt is necessary to keep in mind that we must minimize the risk of overfitting.

(1) *Learning rate*: A value that is too low may make it necessary to increase the number of epochs and make training slower.

(2) *Batch size*: Defines the number of samples that will be analyzed before updating the internal parameters of the model.

(3) *Epoch*: Defines the number of times that the learning algorithm will run on the entire set of training data.

(4) *Hidden layers*: The number of hidden layers and the number of neurons largely determine the complexity of the model and thus its potential learning capacity. For the selection of the number of hidden layers, different units were experimented with, selecting the optimal value by comparing the evaluation metrics.

(5) *Optimization algorithm*: The choice of the optimization algorithm can have a notable impact on the learning of the models. It will update the parameter values based on the set learning rate. In this case, it was selected as Adam because it tries to combine the advantages of RMSProp (similar to gradient descent) together with the advantages of gradient descent with momentum [49].

(6) *Num_heads*: This parameter refers to the number of attention heads in the multi-head attention layer of the transformer network. Multi-head attention allows the network to focus on different parts of the input sequence simultaneously, and the number of heads controls how many different perspectives the network can be considered when processing information.

(7) *Ff_dim*: This parameter indicates the dimension of the feedforward layer within the transformer network structure. The advance layer is a dense layer that is applied after the focus layer. The selection of this parameter can affect the network's ability to learn more complex or simplified patterns in the data.

Goodfellow [50] made recommendations for the optimal parameters for predictions with neural networks, suggesting comparisons using different numbers of cycles. These will be determined according to the computational limitations and possible overfitting of the model. The values selected for each of these parameters are reflected in the Table 1.

Table 1. Selected parameters.

Model	*Learning rate*	*Batch*	*Epoch*	*Hidden layers*	*Optimizer*	*Ff_dim*	*Num_heads*
Transformer	0.001	150	150	1	Adam	75	6
BiLSTM	0.001	25	100	2	Adam	-	-
LSTM	0.001	25	100	2	Adam	-	-

Beck and Arnold [51] stated that the choice of parameters can be easily estimated and differentiated if the parameters are not dependent on each other. This approximation is possible when two parameters are compared, since, if there were more parameters, the computational cost would increase exponentially. Smith [52] highlights the importance of an appropriate choice of the hyperparameters of a DNN to minimize the error obtained describing a new method for choosing the learning rate that eliminates the need to experiment with different values to find the maximum network performance.

3.4. Experimental results and discussion

The results obtained by the combination of EMD and Transformer are compared with the results obtained from analyzing the same time series with a Transformer type network without applying EMD

and after this process the comparison is made with the results obtained by a BiLSTM network. Under the same conditions, applying EMD and without applying it, the choice of a BiLSTM network is justified by recent literature, where these models have shown great accuracy in predicting time series. As stated previously [21,53], this process is repeated with an LSTM network, which allows the results of the proposed model to be contrasted with two models widely used for this purpose.

Table 2 shows the results obtained for the 4 processes used, observing better performance of the Transformer network after applying EMD to the time series, obtaining lower values in all evaluation metrics, demonstrating its effectiveness for price forecasting of WTI.

It is observed that the Transformer models have better performance and greater precision than the predictions made with BiLSTM networks, obtaining a significantly lower error. The RMSE error metric is reduced by 82.47% after applying EMDFormer compared to the model applying EMD to a BiLSTM network, with a reduction of 46.11% in the RSME metric with a traditional BiLSTM model and 40.53% compared to the Transformer model without applying EMD.

Table 2. Models performance in terms of MSE, RMSE, R^2 and MAPE. The best values for each metric are in bold.

	MSE	RMSE	R2	MAPE
EMDFormer	**4.676**	**2.162**	**0.958**	**1.178**
Transformer	13.218	3.636	0.921	6.452
BiLSTM	16.100	4.012	0.951	7.750
EMD-BiLSTM	152.081	12.332	-1.556	8.033
LSTM	261.808	16.180	0.654	14.111
EMD-LSTM	427.339	20.672	-3.778	18.810

The results obtained are mainly due to the ability of transformer networks to capture long-term patterns, as well as to efficiently handle dependencies at different distances, facilitating their learning. Furthermore, its lower propensity for overfitting and the lower need for hyperparameter adjustment make this model ideal for this type of time series, improving the analysis, and allowing the error to be minimized.

A negative result in the R^2 parameter of the EMD-BiLSTM and EMD-LSTM models may be due to overfitting of the network, which coincides with insufficient performance in the rest of the metrics obtained.

The graphs belonging to the six model-based approaches with and without EMD show predictions very close to the actual WTI index price values in Figure 4, especially in the predictions obtained by the EMDFormer model.

Table 3 shows the results obtained in the predictions of the BCTR time series using the same models. Once again, results are observed that improve the remaining results from the EMDFormer model, which confirms that the combination of EMD with the Transformer models provides new possibilities in time series prediction.

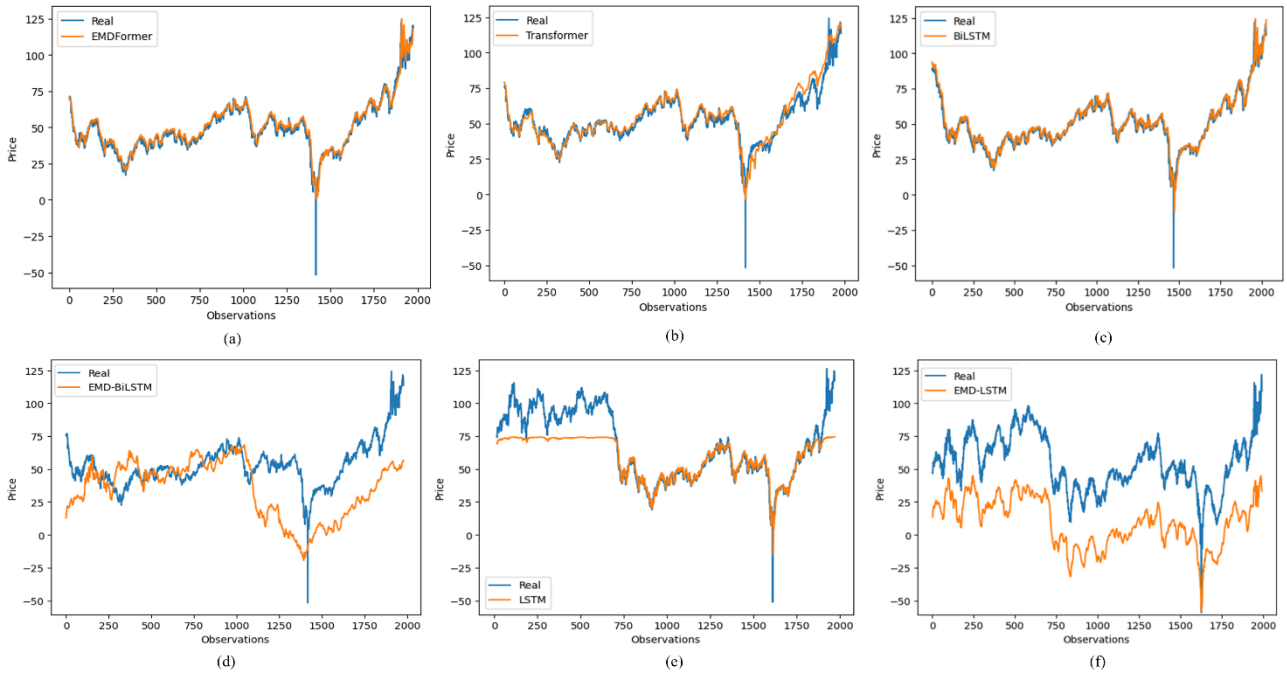

Figure 4. Results from the models: (a) EMDFormer; (b) transformer; (c) BiLSTM; (d) EMD-BiLSTM; (e) LSTM; and (f) EMD-LSTM.

Table 3. Models performance in terms of MSE, RMSE, R^2 and MAPE for BCTR time series. The best values for each metric are in bold.

	MSE	RMSE	R2	MAPE
EMDFormer	14.044	2.214	0.983	1.026
Transformer	4507.082	67.135	-0.505	20.006
BiLSTM	24.802	4.980	0.977	3.412
EMD-BiLSTM	2115.250	45.992	-0.676	18.067
LSTM	16357.522	127.897	0.018	156.978
EMD-LSTM	1129.931	33.614	0.104	12.697

The BCTR time series shows less adjusted results than those obtained with the other time series; however, once decomposed into IMF's, the results obtained by the EMDFormer model show much lower error metrics than the rest of the models, improving the results by 96% with respect to the transformer model without applying EMD.

The ability of Transformer networks to efficiently handle dependencies at different distances is revealed once again, as well as how the application of EMD preprocessing allows the error to be minimized.

The rest of the models present less efficient results and with negative results in the R^2 metric in the Transformer and EMD-BiLSTM models, which shows a lower capacity to capture the characteristics of the time series.

The detailed results of the 6 approaches are shown in Figure 5, with a comparison of the BCTR values with the predictions returned by the models.

Figure 5. Results from the models: (a) EMDFormer; (b) transformer; (c) BiLSTM; (d) EMD-BiLSTM; (e) LSTM; and (f) EMD-LSTM.

Despite the positive results obtained, it is worth noting that the use of empirical mode decomposition as a data preprocessing method for time series prediction with TNNs, while promising, poses several challenges. Implementation complexity arises as it requires advanced technical knowledge, and its effectiveness heavily depends on the quality of input data, making it susceptible to inaccuracies or biases. Additionally, employing EMD adds time and computational resources to the modeling process, which can be prohibitive in resource-constrained environments. Finally, the interpretability of the results may be affected, as EMD decomposition could obscure underlying patterns, making it difficult for end-users to understand and trust the generated predictions.

4. Conclusions and future lines of research

We propose a new methodology for time series forecasting that combines the potential of Transformer networks together with the EMD to achieve greater precision. First, EMD is applied to the time series with the objective of obtaining the resulting IMF's to later be processed by the Transformer network, which will be able to capture the characteristics of each of the series and make more accurate forecasts. This type of networks has the ability to capture the characteristics of different time series of WTI oil price data or data from raw materials using the BCTR index, not only being limited to the economic field, but it is also possible to apply its performance to other types of time series.

Based on this study, further investigation will continue into the feasibility of the proposed methodology for predicting various economic values, as well as time series of different characteristics, considering other recent models based on transformers such as PatchTSTST and DLinear. Different prediction horizons will be considered to identify if there are significant differences in each period.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work was supported in part by the project TED2021-131671B-I00 funded by MCIN/AEI /10.13039/501100011033 and by the European Union NextGenerationEU/ PRTR.

Conflict of interest

All authors declare no conflicts of interest that could affect the publication of this paper.

References

1. J. Lago, F. De Ridder, B. De Schutter, Forecasting spot electricity prices: deep learning approaches and empirical comparison of traditional algorithms, *Appl. Energy*, **221** (2018), 386–405. https://doi.org/10.1016/j.apenergy.2018.02.069
2. G. E. Box, G. M. Jenkins, G. C. Reinsel, G. M. Ljung, *Time series analysis: forecasting and control*, John Wiley & Sons, 2015.
3. J. Contreras, R. Espinola, F. Nogales, A. Conejo, ARIMA models to predict next-day electricity prices, *IEEE Power Eng. Rev.*, **22** (2002), 57–57. https://doi.org/10.1109/MPER.2002.4312577
4. S. Saab, E. Badr, G. Nasr, Univariate modeling and forecasting of energy consumption: the case of electricity in Lebanon, *Energy*, **26** (2001), 1–14. https://doi.org/10.1016/S0360-5442(00)00049-9
5. Lucas, K. Pegios, E. Kotsakis, D. Clarke, Price forecasting for the balancing energy market using machine-learning regression, *Energies*, **13** (2020), 5420. https://doi.org/10.3390/en13205420
6. B. Zhu, D. Han, P. Wang, Z. Wu, T. Zhang, Y. Wei, Forecasting carbon price using empirical mode decomposition and evolutionary least squares support vector regression, *Appl. Energy*, **191** (2017), 521–530. https://doi.org/10.1016/j.apenergy.2017.01.076
7. T. Hill, M. O'Connor, W. Remus, Neural network models for time series forecasts, *Manage. Sci.*, **42** (1996), 1082–1092. https://doi.org/10.1287/mnsc.42.7.1082
8. M. Khashei, M. Bijari, An artificial neural network (*p,d,q*) model for timeseries forecasting, *Expert Syst. Appl.*, **37** (2010), 479–489. https://doi.org/10.1016/j.eswa.2009.05.044
9. S. Bhardwaj, E. Chandrasekhar, P. Padiyar, V. M. Gadre, A comparative study of wavelet-based ANN and classical techniques for geophysical time-series forecasting, *Comput. Geosci.*, **138** (2020), 104461. https://doi.org/10.1016/j.cageo.2020.104461
10. A. D. Piazza, M. C. D. Piazza, G. L. Tona, M. Luna, An artificial neural network-based forecasting model of energy-related time series for electrical grid management, *Math. Comput. Simul.*, **184** (2021), 294–305. https://doi.org/10.1016/j.matcom.2020.05.010
11. B. K. Rajput, P. Sunil, N. Yadav, A novel hybrid model combining *βSARMA* and LSTM for time series forecasting, *Appl. Soft Comput.*, **134** (2023), 110019. https://doi.org/10.1016/j.asoc.2023.110019
12. E. Egrioglu, E. Bas, A new hybrid recurrent artificial neural network for time series forecasting, *Neural Comput. Appl.*, **35** (2023), 2855–2865. https://doi.org/10.1007/s00521-022-07753-w

13. P. H. Borghi, O. Zakordonets, J. P. Teixeira, A COVID-19 time series forecasting model based on MLP ANN, *Proc. Comput. Sci.*, **181** (2021), 940–947. https://doi.org/110.1016/j.procs.2021.01.250
14. S. A. Chen, C. L. Li, N. Yoder, S. O. Arik, T. Pfister, Tsmixer: an all-MLP architecture for time series forecasting, *ArXiv*, 2023. https://doi.org/10.48550/arXiv.2303.06053
15. C. Voyant, M. L. Nivet, C. Paoli, M. Muselli, G. Notton, Meteorological time series forecasting based on MLP modelling using heterogeneous transfer functions, *J. Phys.*, **574** (2015), 012064. https://doi.org/10.1088/1742-6596/574/1/012064
16. K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, *Neural Networks*, **2** (1989), 359–366. https://doi.org/10.1016/0893-6080(89)90020-8
17. T. Ciechulski, S. Osowski, High precision LSTM model for short-time load forecasting in power systems, *Energies*, **14** (2021), 2983. https://doi.org/10.3390/en14112983
18. B. S. Kwon, R. J. Park, K. B. Song, Short-term load forecasting based on deep neural networks using LSTM layer, *J. Electron. Eng. Technol.*, **15** (2020), 1501–1509. https://doi.org/10.1007/s42835-020-00424-7
19. S. Siami-Namini, N. Tavakoli, A. S. Namin, The performance of LSTM and BiLSTM in forecasting time series, *2019 IEEE International Conference on Big Data (Big Data)*, 2019, 3285–3292. https://doi.org/10.1109/BigData47090.2019.9005997
20. J. Kim, N. Moon, BiLSTM model based on multivariate time series data in multiple field for forecasting trading area, *J. Ambient Intell. Humanized Comput.*, 2019. https://doi.org/10.1007/s12652-019-01398-9
21. M. Yang, J. Wang, Adaptability of financial time series prediction based on BiLSTM, *Proc. Comput. Sci.*, **199** (2022), 18–25. https://doi.org/10.1016/j.procs.2022.01.003
22. D. Salinas, V. Flunkert, J. Gasthaus, T. Januschowski, DeepAR: probabilistic forecasting with autoregressive recurrent networks, *Int. J. Forecast.*, **36** (2020), 1181–1191. https://doi.org/10.1016/j.ijforecast.2019.07.001
23. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, et al., Attention is all you need, *ArXiv*, 2017. https://doi.org/10.48550/arXiv.1706.03762
24. J. Devlin, M. W. Chang, K. Lee, K. Toutanova, Bert: pre-training of deep bidirectional transformers for language understanding, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019, 4171–4186. https://doi.org/10.18653/v1/N19-1423
25. A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, et al., An image is worth 16×16 words: transformers for image recognition at scale, *ArXiv*, 2020. https://doi.org/10.48550/arXiv.2010.11929
26. P. Ran, K. Dong, X. Liu, J. Wang, Short-term load forecasting based on CEEMDAN and transformer, *Electr. Power Syst. Res.*, **214** (2023) 108885. https://doi.org/10.1016/j.epsr.2022.108885
27. L. Li, X. Su, X. Bi, Y. Lu, X. Sun, A novel transformer-based network forecasting method for building cooling loads, *Energy Build.*, **296** (2023), 113409. https://doi.org/10.1016/j.enbuild.2023.113409
28. M. J. Walczewski, H. Wöhrle, Prediction of electricity generation using onshore wind and solar energy in Germany, *Energies*, **17** (2024), 844. https://doi.org/10.3390/en17040844

29. A. Rosato, R. Araneo, A. Andreotti, F. Succetti, M. Panella, 2-D convolutional deep neural network for the multivariate prediction of photovoltaic time series, *Energies*, **14** (2021), 2392. https://doi.org/10.3390/en14092392
30. P. Wibawa, A. B. P. Utama, H. Elmunsyah, U. Pujianto, F. A. Dwiyanto, L. Hernandez, Time-series analysis with smoothed convolutional neural network, *J. Big Data*, **9** (2022), 44. https://doi.org/10.1186/s40537-022-00599-y
31. S. Jung, J. Moon, S. Park, E. Hwang, An attention-based multilayer GRU model for multistep-ahead short-term load forecasting, *Sensors*, **21** (2021), 1639. https://doi.org/10.3390/s21051639
32. S. Ungureanu, V. Topa, A. Cziker, Deep learning for short-term load forecasting-industrial consumer case study, *Appl. Sci.*, **11** (2021), 10126. https://doi.org/10.3390/app112110126
33. Y. Chen, F. Ding, L. Zhai, Multi-scale temporal features extraction based graph convolutional network with attention for multivariate time series prediction, *Expert Syst. Appl.*, **200** (2022), 117011. https://doi.org/10.1016/j.eswa.2022.117011
34. A. Lazcano, P. J. Herrera, M. Monge, A combined model based on recurrent neural networks and graph convolutional networks for financial time series forecasting, *Mathematics*, **11** (2023), 224. https://doi.org/10.3390/math11010224
35. X. Guo, Q. Zhao, D. Zheng, Y. Ning, Y. Gao, A short-term load forecasting model of multi-scale CNN-LSTM hybrid neural network considering the real-time electricity price, *Energy Reports*, **6** (2020), 1046–1053, https://doi.org/10.1016/j.egyr.2020.11.078
36. W. Yang, J. Shi, S. Li, Z. Song, Z. Zhang, Z. Chen, A combined deep learning load forecasting model of single household resident user considering multi-time scale electricity consumption behavior, *Appl. Energy*, **307** (2022), 118197. https://doi.org/10.1016/j.apenergy.2021.118197
37. L. Ji, Y. Zou, K. He, B. Zhu, Carbon futures price forecasting based with ARIMA-CNN-LSTM model, *Proc. Comput. Sci.*, **62** (2019), 33–38. https://doi.org/10.1016/j.procs.2019.11.254
38. Z. Zeng, R. Kaur, S. Siddagangappa, S. Rahimi, T. Balch, M. Veloso, Financial time series forecasting using CNN and transformer, *ArXiv*, 2023. https://doi.org/10.48550/arXiv.2304.04912
39. S. Kotsiantis, D. Kanellopoulos, P. Pintelas, Data preprocessing for supervised learning, *Int. J. Comput. Sci.*, **1** (2006), 111–117.
40. D. Yang, J. Guo, S. Sun, J. Han, S. Wang, An interval decomposition-ensemble approach with data-characteristic-driven reconstruction for short-term load forecasting, *Appl. Energy*, **306** (2022), 117992. https://doi.org/10.1016/j.apenergy.2021.117992
41. Neeraj, J. Mathew, R. J. Behera, EMD-Att-LSTM: a data-driven strategy combined with deep learning for short-term load forecasting, *J. Mod. Power Syst. Clean Energy*, **10** (2022), 1229–1240. https://doi.org/10.35833/MPCE.2020.000626
42. N. Li, J. Dong, L. Liu, H. Li, J. Yan, A novel EMD and causal convolutional network integrated with Transformer for ultra short-term wind power forecasting, *Int. J. Electr. Power Energy Syst.*, **154** (2023), 109470. https://doi.org/10.1016/j.ijepes.2023.109470
43. X. Wang, S. Dong, R. Zhang, An integrated time series prediction model based on empirical mode decomposition and two attention mechanisms, *Information*, **14** (2023), 610. https://doi.org/10.3390/info14110610
44. Z. Zhao, W. Chen, X. Wu, P. C. Y. Chen, J. Liu, LSTM network: a deep learning approach for short-term traffic forecast, *IET Intell. Transp. Syst.*, **11** (2017), 68–75. https://doi.org/10.1049/iet-its.2016.0208
45. Y. Ren, P. N. Suganthan, N. Srikanth, A comparative study of empirical mode decomposition-based short-term wind speed forecasting methods, *IEEE Trans. Sustain. Energy*, **6** (2014), 236–244. https://doi.org/10.1109/TSTE.2014.2365580

46. N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zhen, et al., The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis, *Proc. R. Soc. London*, **454** (1998), 903–995. https://doi.org/10.1098/rspa.1998.0193
47. C. Li, G. Qian, Stock price prediction using a frequency decomposition based GRU Transformer neural network, *Appl. Sci.*, **13** (2022), 222. https://doi.org/10.3390/app13010222
48. D. C. Frechtling, *Practical tourism forecasting*, Oxford: Butterworth-Heinemann, 1996.
49. R. Sun, Optimization for deep learning: theory and algorithms, *ArXiv*, 2019. https://doi.org/10.48550/arXiv.1912.08957
50. I. Goodfellow, Nips 2016 tutorial: generative adversarial networks, *ArXiv*, 2016. https://doi.org/10.48550/arXiv.1701.00160
51. J. V. Beck, K. J. Arnold, *Parameter estimation in engineering and science*, James Beck, 1977. https://doi.org/10.2307/1403212
52. L. N. Smith, A disciplined approach to neural network hyper-parameters: part 1-learning rate, batch size, momentum, and weight decay, *ArXiv*, 2018. https://doi.org/10.48550/arXiv.1803.09820
53. M. Pirani, P. Thakkar, P. Jivrani, M. H. Bohara, D. Garg, A comparative analysis of ARIMA, GRU, LSTM and BiLSTM on financial time series forecasting, *2022 IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE)*, 2022. https://doi.org/10.1109/ICDCECE53908.2022.9793213

AIMS Press