Mathematical Biosciences and Engineering

MBE, 20(9): 16596–16627.
DOI: 10.3934/mbe.2023739
Received: 28 June 2023
Revised: 04 August 2023
Accepted: 07 August 2023
Published: 17 August 2023

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

Research article

Deep belief improved bidirectional LSTM for multivariate time series forecasting

Keruo Jiang[1,2], Zhen Huang[1,2], Xinyan Zhou[2], Chudong Tong[2], Minjie Zhu[3,*] and Heshan Wang[3]

[1] State Grid Ningbo Electric Power Supply Company, Ningbo 315000, China
[2] Faculty of Electrical Engineering & Computer Science, Ningbo University, Ningbo 315211, China
[3] School of Electronic and Information Engineering, Zhengzhou University, Zhengzhou 450001, China

* **Correspondence:** Email: zhumj@zzu.edu.cn; Tel: +8637167783113; Fax: +8637167783113.

Abstract: Multivariate time series (MTS) play essential roles in daily life because most real-world time series datasets are multivariate and rich in time-dependent information. Traditional forecasting methods for MTS are time-consuming and filled with complicated limitations. One efficient method being explored within the dynamical systems is the extended short-term memory networks (LSTMs). However, existing MTS models only partially use the hidden spatial relationship as effectively as LSTMs. Shallow LSTMs are inadequate in extracting features from high-dimensional MTS; however, the multilayer bidirectional LSTM (BiLSTM) can learn more MTS features in both directions. This study tries to generate a novel and improved BiLSTM network (DBI-BiLSTM) based on a deep belief network (DBN), bidirectional propagation technique, and a chained structure. The deep structures are constructed by a DBN layer and multiple stacked BiLSTM layers, which increase the feature representation of DBI-BiLSTM and allow for the model to further learn the extended features in two directions. First, the input is processed by DBN to obtain comprehensive features. Then, the known features, divided into clusters based on a global sensitivity analysis method, are used as the inputs of every BiLSTM layer. Meanwhile, the previous outputs of the shallow layer are combined with the clustered features to reconstitute new input signals for the next deep layer. Four experimental real-world time series datasets illustrate our one-step-ahead prediction performance. The simulating results confirm that the DBI-BiLSTM not only outperforms the traditional shallow artificial neural networks (ANNs), deep LSTMs, and some recently improved LSTMs, but also learns more features of the MTS data. As compared with conventional LSTM, the percentage

improvement of DBI-BiLSTM on the four MTS datasets is 85.41, 75.47, 61.66 and 30.72%, respectively.

Keywords: deep long short-term memory; time series forecasting; feature extraction; deep belief network

1. Introduction

Presently, real-life multivariate time series (MTS) data sets have drawn considerable interest from diverse fields, including time series of sensor events [1], the Internet of Things [2], compartmental epidemiological models [3], and human conduct prediction [4]. Classical linear models include the autoregressive (AR) model [5], moving average (MA) model [6], autoregressive moving average (ARMA) model [7], and autoregressive integrated moving average (ARIMA) model [8]. Some nonlinear MTS models were applied for MTS prediction, such as threshold AR [9] and bilinear models [10]. However, the classical MTS models only partially use the hidden spatial relationship between variable pairs and are limited by complex constraints. Support vector regression (SVR) [11], artificial neural network (ANN) [12], gradient-boosted regression tree (GBRT) [13], and gradient-boosting decision tree (GBDT) [14] are prominent machine learning (ML) algorithms that have gained recognition for their superior prediction performance. Conventional ML and ANN algorithms that propagate forward (FFANN) are restricted in MTS forecasting because they must consider the correlation between the multi-dimensional input variables over time [15]. Natural techniques for modeling sequence-based systems with memories are recurrent neural networks (RNNs), which are more appropriate than FFANN [16]. RNNs have been recognized as one of the most efficient models for predicting MTS [17]. Nevertheless, traditional RNNs still need to improve the handling of long-range time-dependent MTS datasets, thereby decreasing prediction accuracy [18].

To solve the shortcoming of conventional RNNs, researchers [19–22] have employed a long short-term memory (LSTM) model that is treated as an extended RNN with memory gates. In the following studies, the employment of bidirectional LSTM (BiLSTM) aims to overcome the limitation of LSTM that solely trains signals in a unidirectional manner. However, the feature learning characteristics of BiLSTM remain unclear [23]. Shallow BiLSTMs are still ineffective in representing the MTS features exhibiting high nonlinearity and long-term dependencies. Ewees et al. [24] employed a novel LSTM based on a heap-based optimizer to address complex optimization and engineering problems. Liu et al. [25] introduced an integrated hybrid convolutional neural LSTM network based on error correction and variational mode decomposition for hourly stepwise solar irradiance forecasting. Neshat et al. [26] proposed a deep hybrid LSTM prediction model utilizing a convolutional neural network featuring BiLSTM and an adaptive decomposition-based algorithm for wave power prediction. Li et al. [27] operated an evolutionary attention-based multi-layer LSTM network (EA-LSTM) through learning with competitive random search for MTS forecasting. Deep LSTM or BiLSTM architectures can generally acquire good representations of high-dimensional, complex, and strongly nonlinear MTS signals.

Notably, a deep belief network (DBN) [28] employed by Hinton et al. can be considered a layered feature extraction method that consists of multiple restricted Boltzmann machines (RBMs).

DBNs expresses the potential rules of the input features and exhibits better generalization capacities. Unlike other traditional nonlinear models, the apparent merit of DBN is its distinctive unsupervised pre-training to eliminate over-fitting in the training process. Another advantage of a DBN is that it is simple to sample signals from top to bottom, and the topology of a DBN is more suitable for ANN models based on a backpropagation algorithm. However, DBNs still has some inherent drawbacks, such as high computational complexity, slow training time, and poor application on large-scale data. One of the most troubling drawbacks of DBN application is that DBNs need to be stronger in dealing with time series with complex patterns or nonlinear dynamic systems. To solve the above disadvantage, we can either extend the DBN structure or generate hybrid models based on DBNs; for example, we can either add RNN layers with dynamic characteristics or encoder layers with the ability to handle transient data to the original DBN structure. In recent years, hybrid deep DBN-based RNNs have shown better potential for time series forecasting. Sun et al. [29] applied a DBN-based echo state network (ESN) for MTS forecasting that can efficiently learn layered dataset characteristics of sequence datasets. Li et al. [30] constructed a deep ANN based on an improved DBN and ESN for blockchain virtual currency prediction. Wu et al. [31] employed a chained structure ESN based on stacked subnetwork modules for MTS prediction.

To enhance the forecasting accuracy of MTS datasets and improve the feature extraction capability of traditional BiLSTM, this study proposes a new deep belief improved BiLSTM (DBI-BiLSTM). Although the BiLSTM can learn data features from different directions, more than a single-layer BiLSTM is needed to understand MTS data with many features and high-dimensional complexity. Therefore, the DBN is added on top of the multi-layer BiLSTM to improve the data representation of the DBI-BiLSTM, and a chained structure is introduced to prevent the loss of features during model training. In DBI-BiLSTM, the feature vectors discovered by the DBN are divided into different clusters based on a variance-based global sensitivity analysis (GSA) method. Then, these clustered features are used as inputs to other BiLSTM neurons. Among the DBI-BiLSTM, the outputs of the former BiLSTM neurons are connected with the additional clustered features based on GSA to be combined into new input sequences for the following nearby layer.

This study has three critical innovations and contributions that can be emphasized as follows:

1) Our proposed method, DBI-BiLSTM, leverages the advantages of DBN and DI-BiLSTM by utilizing a DBN for efficient feature extraction and DI-BiLSTM for generating a hierarchical data representation from different perspectives. This combination allows for more comprehensive modeling of MTS datasets, resulting in improved performance.

2) The proposed DBI-BiLSTM model utilizes a deep BiLSTM architecture composed of stacked BiLSTM modules to capture dynamic features into hierarchical layers. This design enhances the model’s flexibility and robustness in handling MTS forecasting.

3) The variance-based GSA algorithm is employed to identify the sensitivity of the DBN module's output features. Primary features with high sensitivity are injected into the first layer of the DI-BiLSTM. In contrast, supplementary features with low sensitivity are fed into the subsequent layers of the DI-BiLSTM.

This paper is structured as follows: Section 2 provides a fundamental overview of DBN, GSA, single-layer BiLSTM, and multi-layer BiLSTM. Section 3 thoroughly interprets the employed DBI-BiLSTM model's architecture and training methodology. Section 4 describes the MTS experimental outcomes and discusses the distribution of the weights. An overall conclusion is presented in the last section.

2. Materials and methods

2.1. Relevant researches and backgrounds

2.1.1. Deep belief network

A DBN is a probabilistic productive network that consists of several RBM layers. A DBN extracts data features using an unsupervised layer-by-layer training method [32], as shown in Figure 1. An RBM is an unsupervised non-linear feature extractor based on a Markov random field, including two essential layers: a visible cell and a hidden unit layer. The output of each RBM hidden unit is wholly linked to the next RBM unit by symmetric undirected synapses. These RBM properties lead to a conditional independence between visible and hidden cells.

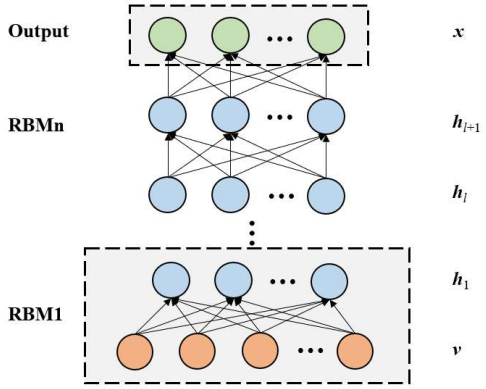

Figure 1. The architecture of DBN consists of multiple RBMs.

The joint probability distribution identified by the RBM's weights utilized an energy-based function of $\{\boldsymbol{v}, \boldsymbol{h}\}$, as shown in the following equation:

$$En(\boldsymbol{v},\boldsymbol{h};\theta) = -\boldsymbol{v}^T\boldsymbol{W}\boldsymbol{h} - \boldsymbol{a}^T\boldsymbol{v} - \boldsymbol{b}^T\boldsymbol{h} = -\sum_{i=1}^{D_v}\sum_{j=1}^{D_h} w_{ij}v_i h_j - \sum_{i=1}^{D_v} a_i v_i - \sum_{j=1}^{D_h} b_j h_j \ , \tag{1}$$

where $\theta = \{b_i, a_j, w_{ij}\}$, w_{ij} represents the weight from visible cell i to hidden cell j, and a_i and b_j are the bias of units i and j, respectively. The joint probability distribution of the RBM model over the visible-hidden cells is computed according to Eq (1) as follows:

$$P(\boldsymbol{v},\boldsymbol{h};\theta) = \frac{1}{Z(\theta)} exp(-E(\boldsymbol{v},\boldsymbol{h};\theta)) \ , \tag{2}$$

where $Z(\theta)$ is a normalizing constant value or partition function obtained from the summary of all potential energy allocations combining cells i and j.

$$Z(\theta) = \sum_{\boldsymbol{v}} \sum_{\boldsymbol{h}} exp(-E(\boldsymbol{v}, \boldsymbol{h}; \theta)) \tag{3}$$

The RBM can obtain the probability of input datasets through the energy equation. According to the joint probability distribution function, the conditional probability functions of cells i and j can be obtained by the following functions:

$$P(h_j = 1 \mid \boldsymbol{v}) = \delta(b_j + \sum_i v_i w_{ij}) \tag{4}$$

,

$$P(v_i = 1 \mid \boldsymbol{h}) = \delta(a_i + \sum_j h_j w_{ij}) \tag{5}$$

,

$$\delta(x) = \frac{1}{1 + exp(-x)} \tag{6}$$

.

The input state will be reconstructed by arranging every v_i to 1 with the probability given by Eq (5). Thus, the state of hidden cells is gradually renewed to represent the reconstruction features. The maximization implements the training method in RBM and the probability distribution of the training data with regard to the model variables as follows:

$$\text{maximize}\{b_j, a_i, w_{ij}\} \frac{1}{m} \sum_{l=1}^{m} \log(P(\boldsymbol{v}^l)) \tag{7}$$

where m represents the length of training datasets. Therefore, the objective function is a log-likelihood term; a gradient descent algorithm should solve it. However, the gradient calculation of the log-likelihood term is difficult to implement due to the presence of $Z(\theta)$. Thus, sampling methods such as contrastive divergence [33] and persistent contrastive divergence [34] in a gradient calculation can be utilized instead.

2.1.2. Bidirectional LSTM

LSTM is an improved version of the conventional RNNs, that utilizes specially designed memory units to efficiently express the long-term dependencies of MTS datasets. In contrast to classical RNNs, the design of LSTM offers an efficient answer to the vanishing gradient issue. The LSTM cell learns the present hidden units' state according to the current input and the prior hidden units' state. Nonetheless, it replaces the architecture of the hidden units with a memory cell that can represent the long-term dependence of the MTS signals. As shown in Figure 2, the LSTM network introduces four controlled gates, including one input, one output, one forgets, and one self-loop memory cell, to manipulate the interactions of the information streams between various memory neurons. In the hidden LSTM layers, the forget gate determines which information from the previous moment should be preserved or ignored. At the same time, the entrance of the input neurons can decide whether input signals should be injected into the memory units' information. The gate of the

output neurons handles whether the state of different memory units can be changed. Given the input $\boldsymbol{x}_t$ of MTS and the dynamic output state $\boldsymbol{h}_t$, the gates states, outputs of hidden layers, and neuron states can be computed using the following equations:

$$\boldsymbol{i}_t = \sigma\left(\boldsymbol{U}_i \boldsymbol{x}_t + \boldsymbol{W}_i \boldsymbol{h}_{t-1} + \boldsymbol{b}_i\right), \tag{8}$$

$$\boldsymbol{f}_t = \sigma\left(\boldsymbol{U}_f \boldsymbol{x}_t + \boldsymbol{W}_f \boldsymbol{h}_{t-1} + \boldsymbol{b}_f\right), \tag{9}$$

$$\boldsymbol{o}_t = \sigma\left(\boldsymbol{U}_o \boldsymbol{x}_t + \boldsymbol{W}_o \boldsymbol{h}_{t-1} + \boldsymbol{b}_o\right), \tag{10}$$

$$\tilde{\boldsymbol{c}}_t = tanh\left(\boldsymbol{U}_c \boldsymbol{x}_t + \boldsymbol{W}_c \boldsymbol{h}_{t-1} + \boldsymbol{b}_c\right), \tag{11}$$

$$\boldsymbol{c}_t = \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \odot \tilde{\boldsymbol{c}}_t, \tag{12}$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t \odot tanh\left(\boldsymbol{c}_t\right). \tag{13}$$

In the equations used for LSTM, the recurrent weight matrices are represented by $\boldsymbol{W}_i$, $\boldsymbol{W}_f$, $\boldsymbol{W}_o$, and $\boldsymbol{W}_c$, while the weight matrices for the input, forget, output, and memory cell gates are denoted by $\boldsymbol{U}_i$, $\boldsymbol{U}_f$, $\boldsymbol{U}_o$, and $\boldsymbol{U}_c$, respectively. The gates biases are expressed as $\boldsymbol{b}_i$, $\boldsymbol{b}_f$, $\boldsymbol{b}_o$, and $\boldsymbol{b}_c$. At each time step, $\boldsymbol{h}_t$ refers to the state of the hidden layer, and $\boldsymbol{o}_t$ is the output. The candidate cell state $\tilde{\boldsymbol{c}}_t$ is used to update the original memory cell state $\boldsymbol{c}_t$. The hyperbolic tangent function is represented by *tanh*, and the logistic sigmoid activation function is denoted by σ. The multiplication operation by elementwise is described by $\odot$.

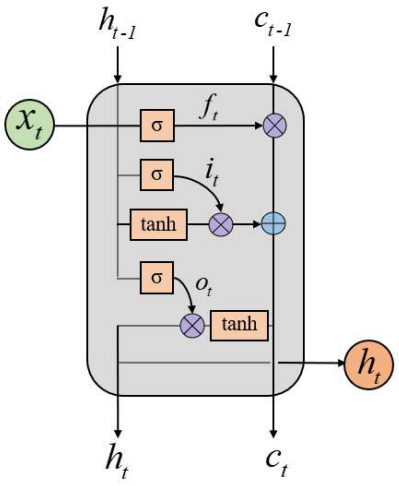

Figure. 2. The framework for the basic shallow LSTM.

The traditional LSTM may inadvertently discard important sequential information during training because it processes input signals in only one direction. As a result, the time series data

cannot be thoroughly analyzed. To address this limitation, the BiLSTM was developed with a bidirectional structure that can capture representations of MTS data through forward and backward directions. This procedure is illustrated in Figure 3.

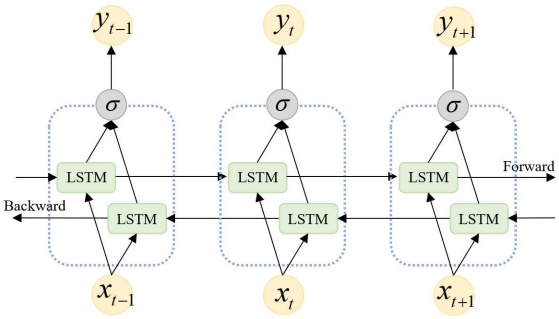

Figure. 3. The architecture diagram of BiLSTM network.

The BiLSTM consists of two LSTM layers that run in parallel in two opposite directions. In the forward propagation direction, the information of the hidden LSTM neurons, represented by $\boldsymbol{h}_{f(t)}$, retains information from past sequence values. In the reverse propagation direction, the hidden state, represented by $\boldsymbol{h}_{b(t)}$, contains data from future MTS values. $\boldsymbol{h}_{f(t)}$ and $\boldsymbol{h}_{b(t)}$ are linked together to create the ultimate outputs of BiLSTM. The t-th hidden states of BiLSTM for the forward and backward states are calculated using the following equations:

$$\boldsymbol{h}_{f(t)} = \psi(\boldsymbol{W}_{fh}\boldsymbol{x}_t + \boldsymbol{W}_{fhh}\boldsymbol{h}_{f(t-1)} + \boldsymbol{b}_{fb}), \tag{14}$$

$$\boldsymbol{h}_{b(t)} = \psi(\boldsymbol{W}_{bh}\boldsymbol{x}_t + \boldsymbol{W}_{bhh}\boldsymbol{h}_{b(t+1)} + \boldsymbol{b}_b). \tag{15}$$

The weight matrices $\boldsymbol{W}_{fh}$ and $\boldsymbol{W}_{bh}$ represent the forward and backward synapses weights from the input to the internal unit weights. Similarly, $\boldsymbol{W}_{fhh}$ and $\boldsymbol{W}_{bhh}$ represent the forward and backward feedback recurrent weights. Additionally, $\boldsymbol{b}_{fb}$ and $\boldsymbol{b}_b$ correspond to bias information in two directions. This study gives the activation function of the hidden layer ψ as *tanh*. Using these components, the output of BiLSTM $\boldsymbol{y}_t$ is described using the following equation:

$$\boldsymbol{y}_t = \sigma\left(\boldsymbol{W}_{fhy}\boldsymbol{h}_{f(t)} + \boldsymbol{W}_{bhy}\boldsymbol{h}_{b(t)} + \boldsymbol{b}_y\right) \tag{16}$$

Where the output layer's forward and backward weights are denoted by $\boldsymbol{W}_{fhy}$ and $\boldsymbol{W}_{bhy}$, respectively. The activation function of the output layer σ is generally given as either a sigmoidal or linear function. Additionally, $\boldsymbol{b}_y$ denotes the output bias.

2.1.3. Variance based global sensitivity analysis

Due to the high dimensionality and numerous features of the MTS data, the input of DI-BiLSTM or the output of DBN becomes more complex after DBN processing. Therefore, conducting a

sensitivity analysis on the DBN output is necessary to reduce the input complexity of the multi-layer BiLSTM. Among different available algorithms for network output sensitivity analysis (SA), variance-based SA algorithms, which apply Monte Carlo sampling and are based on model decomposition, have demonstrated their versatility and effectiveness. An improved variance-based GSA algorithm was proposed by Saltelli et al. [35] to compute a model's total sensitivity indices (SI). GSA has been applied to a parametric sensitivity analysis for various models with excellent stability and minimal computational cost. Considering the output of a nonlinear model $\boldsymbol{Y} = f(\boldsymbol{X}) = f(x_1, x_2, \cdots, x_k)$, where $\boldsymbol{Y}$ is the output and $\boldsymbol{X} = (x_1, x_2, \cdots, x_k)$ contains k input factors, the final model variance is computed as follows:

$$V(Y) = \sum_{i=1}^{k} V_i + \sum_{i=1}^{k} \sum_{j=i+1}^{k} (V_{ij} + \cdots + V_{i,\ldots,k}), \tag{17}$$

where $\boldsymbol{X}$ is readjusted to a k-dimensional hypercube Ω^k ($\Omega^k = \{\boldsymbol{X} \mid 0 \le x_i \le 1, i = 1, \cdots, k\}$), $V(Y)$ denotes the sum variance, $V_i = V\left|E(Y \mid x_i)\right|$ is the variance of the parameter x_i, and V_{ij} is the variance of the interaction between parameters x_i and x_j .The first-order sensitivity S_i and total sensitivity S_{Ti} of parameter x_i are expressed as follows:

$$S_i = \frac{V_i}{V(Y)} = \frac{V\left|E(Y \mid x_i)\right|}{V(Y)}, \tag{18}$$

$$S_{Ti} = S_1 + \sum_{j \ne i} S_{ij} + \cdots = \frac{E\left|V(Y \mid x_{\sim i})\right|}{V(Y)}, \tag{19}$$

where $x_{\sim i}$ stands for all factors except x.

To calculate the total sensitivity values for factor x_i, we must create two independent sampling matrices, $\boldsymbol{P}$ and $\boldsymbol{Q}$, each of size (N, k), where N is the sample length and k is the number of model variables. Every row in the matrices corresponds to an input parameter vector $\boldsymbol{X}$ for the model. We can use Monte Carlo methods to approximate $\hat{V}(Y)$, $\hat{S}_i$ and $\hat{S}_{Ti}$. The computation function is shown as follows:

$$\hat{f}_0 = \frac{1}{N} \sum_{j=1}^{N} f(\boldsymbol{P})_j, \tag{20}$$

$$\hat{V}(Y) = \frac{1}{N} \sum_{j=1}^{N} (f(\boldsymbol{P})_j)^2 - \hat{f}_0^2, \tag{21}$$

$$\hat{S}_i = \frac{1}{N}\sum_{j=1}^{N}\frac{f(\boldsymbol{Q})_j(f(\boldsymbol{P}_{\boldsymbol{Q}}^{(i)})_j - f(\boldsymbol{P})_j)}{\hat{V}(Y)}, \tag{22}$$

$$\hat{S}_{Ti} = \frac{1}{2N}\sum_{j=1}^{N}\frac{(f(\boldsymbol{P})_j - f(\boldsymbol{P}_{\boldsymbol{Q}}^{(i)})_j)^2}{\hat{V}(Y)}, \tag{23}$$

where $\hat{f}_0$ is the average value of a model's output, $\boldsymbol{P}_{\boldsymbol{Q}}^{(i)}$ represents the new matrix obtained by replacing the *i*th column data in matrix $\boldsymbol{P}$ with the *i*th column data in matrix $\boldsymbol{Q}$. To calculate $\hat{S}_i$ and $\hat{S}_{Ti}$, the model simulation number is set to $N(k+2)$. Due to interaction effects, the sum of $\hat{S}_{Ti}$ of k parameters would be greater than 1, therefore $\hat{S}_{Ti}$ was normalized. In this study, the total sensitivity of each parameter in the model was evaluated according to Eq (23).

2.2. Deep belief improved BiLSTM

Comprehensive research has shown that adding depth to ANNs can efficiently increase the overall performance of ANN models [36]. Similarly, the academic community has been impressed by the learning capabilities of deep RNNs in MTS forecasting [32, 37]. Inspired by the learning capacity of multi-layer RNNs and the characteristic extraction ability of DBN, we develop a DBI-BiLSTM model for MTS prediction, consisting of multiple BiLSTM modules connected through a chained structure, as shown in Figure 4. The DBI-BiLSTM model has three main parts: a DBN-based feature extraction component, a sensitivity analysis component, and an improved BiLSTM-based prediction model. In the DBI-BiLSTM network, the DBN output vectors are categorized into major and minor features based on their sensitivity to the model's output (SI). The primary features are those that are highly sensitive to the model's production, while the minor features are those that are less sensitive. The significant features with a high sensitivity are injected into the first layer of the DI-BiLSTM. In contrast, supplementary features with low sensitivity are fed into the subsequent layers of the DI-BiLSTM.

The significant benefits of the proposed DBI-BiLSTM can be outlined in three aspects.

1) DBN's powerful feature extraction capability ensures the feature representation ability of DBI-BiLSTM for MTS datasets, which further reduces the prediction error of the employed DBI-BiLSTM for MTS prediction.

2) The DBI-BiLSTM model incorporates both multi-layer and bidirectional RNN structures. The multi-layer enhances the training and generalization capabilities of DBI-BiLSTM. In contrast, the propagation in different directions enables the DBI-BiLSTM to efficiently train the MTS signals from two directions. This characteristic makes DBI-BiLSTM more capable of adapting to the natural features of the MTS datasets.

3) The multi-layer architecture of DBI-BiLSTM is different from the traditional deep LSTM structure; it is an improved structure that includes multiple BiLSTM modules. Each BiLSTM module can dynamically map learned feature vectors from the DBN layer. The forecasting results of

DBI-BiLSTM are affected by continuously training the output of the former BiLSTM neurons and by increasing additional input data features in different BiLSTM hidden states. This makes the addressed DBI-BiLSTM more reliable and stable for MTS forecasting compared to traditional deep LSTM structures.

A structure explanation diagram of the DI-BiLSTM with three layers is shown in Figure 5, in which each layer is represented through a distinguishing color.

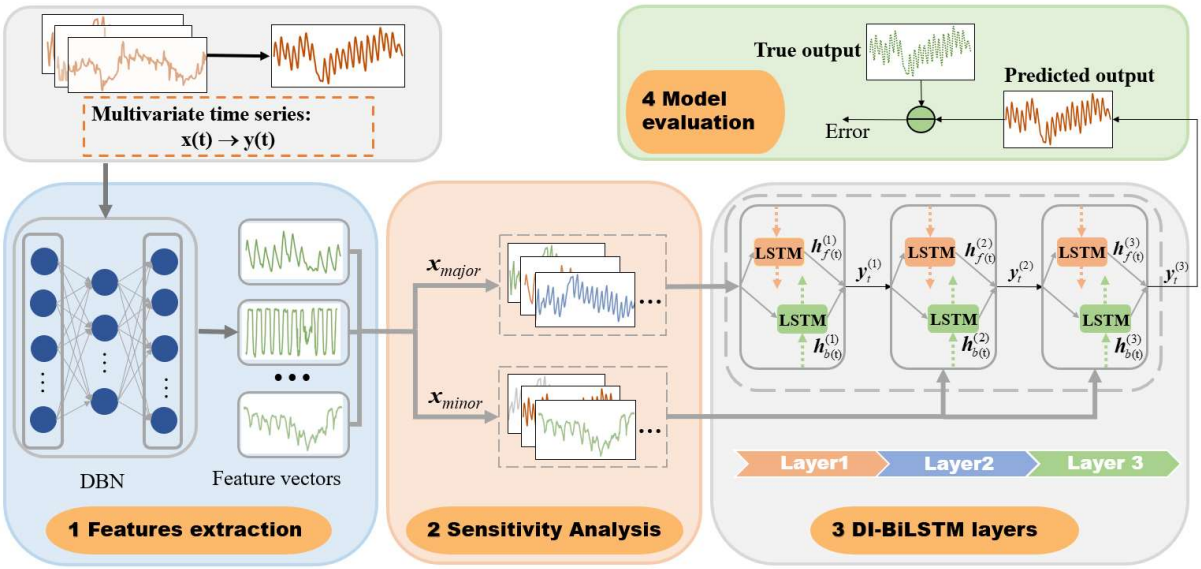

Figure. 4. The proposed DBI-BiLSTM structure.

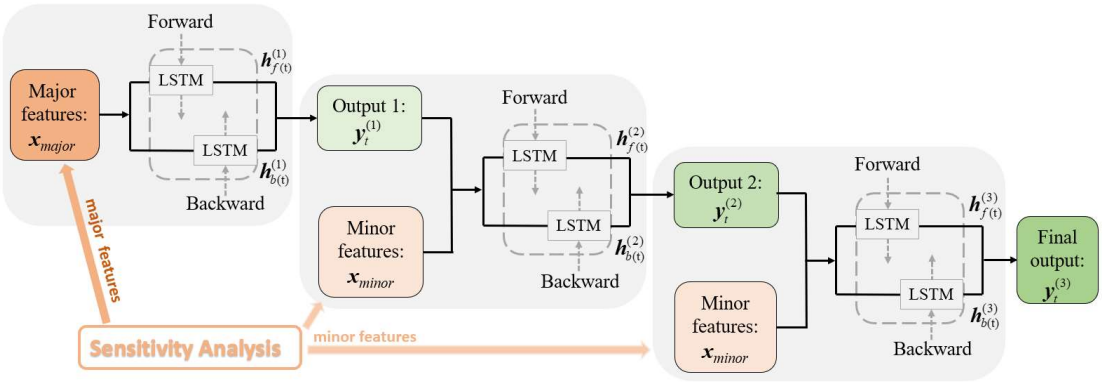

Figure. 5. Structure diagram and principle of DI-BiLSTM with three layers.

A consistent hidden layer size of N_h is used for each BiLSTM with different propagation direction layers to make the experiments more compact. Let I_i and O_i represent the number of input and output units, respectively, for layer i. The export signals of the forward and backward recurrent neurons of the i-th BiLSTM layer are represented as $\boldsymbol{h}_{f(t)}^{(i)}$ and $\boldsymbol{h}_{b(t)}^{(i)}$, while the bias signals of the i-th BiLSTM module for different directional neurons are represented by $\boldsymbol{b}_{fb}^{(i)}$ and $\boldsymbol{b}_{b}^{(i)}$. $\boldsymbol{b}_{y}^{(i)}$ denotes the

output bias signals of the *i*-th BiLSTM module. The output states of the recurrent layer and the final output can be expressed as follows:

$$\boldsymbol{h}_{f(t)}^{(i)} = \psi(\boldsymbol{W}_{fh}[\boldsymbol{x}_t^{(i)}; \boldsymbol{y}_t^{(i-1)}] + \boldsymbol{W}_{fhh}\boldsymbol{h}_{f(t-1)}^{(i)} + \boldsymbol{b}_{fb}^{(i)}), \tag{24}$$

$$\boldsymbol{h}_{b(t)}^{(i)} = \psi(\boldsymbol{W}_{bh}[\boldsymbol{x}_t^{(i)}; \boldsymbol{y}_t^{(i+1)}] + \boldsymbol{W}_{bhh}\boldsymbol{h}_{b(t+1)}^{(i)} + \boldsymbol{b}_{b}^{(i)}), \tag{25}$$

$$\boldsymbol{y}_t^{(i)} = \sigma\left(\boldsymbol{W}_{fhy}\boldsymbol{h}_{f(t)}^{(i)} + \boldsymbol{W}_{bhy}\boldsymbol{h}_{b(t)}^{(i)} + \boldsymbol{b}_{y}^{(i)}\right). \tag{26}$$

In the experiments of this paper, the Adaptive Moment Estimation (Adam) method [38] is used as the learning optimizer to renew the weights of the DBI-BiLSTM model with an original learning rate (set to 0.001). For an MTS prediction, the training and testing procedure loss function can be defined as the mean square error (MSE) function:

$$MSE = \frac{1}{n}\sum_{t=1}^{n}\left(\boldsymbol{y}_t - \hat{\boldsymbol{y}}_t\right)^2 \tag{27}$$

where $\hat{\boldsymbol{y}}_t$ represents the actual forecasted signal, $\boldsymbol{y}_t$ denotes the wanted output signal and *n* denotes the length of $\boldsymbol{y}_t$.

Figure 4 shows that the model begins by using multiple DBN layers to map inputs to their feature representations. Then, the clustered feature vectors based on GSA are fed into the DI-BiLSTM layers, which process them in different directions. The outputs of the DI-BiLSTM layers are then passed through a wholly connected layer, which serves as the regression neurons and uses an adjusted linear activation function. A suitable dropout probability (set to 0.2) is applied to the proposed forecasting model to prevent overfitting of the MTS datasets. Table 1 summarizes the proposed DBI-BiLSTM parameters.

The learning mechanism of DBI-BiLSTM is shown as follows:

Algorithm: Employed DBI-BiLSTM model

Initialization: datasets are divided into training, validation, and testing set which are then normalized before training;

Input: MTS data $\boldsymbol{V} = \{\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_n\}$;

Output: extracted feature vectors $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n\}$ of data $\boldsymbol{V}$;

for data in training and testing data do

Extract features ($\boldsymbol{X}$) of all the datasets by the DBN layer and update the weights of the DBN layer through Eqs (4)–(7);

end for

The total SI of $\hat{S}_{Ti}$ for each input feature vector ($\boldsymbol{X}$) is calculated according to Eqs (20)–(23)

and ranks the SI of feature ($\boldsymbol{X}$);

Divide $\boldsymbol{X}$ into major input features (high SI) and minor input features (low SI) based on the ranked SI and initialize the weights and layers of BiLSTM;

Input: feature representation vectors $\boldsymbol{X}=[\boldsymbol{X}_{major},\boldsymbol{X}_{minor}]^T=\{\boldsymbol{x}_1,\boldsymbol{x}_2,...,\boldsymbol{x}_n\}$;

Output: desired output $\boldsymbol{Y}=\{\boldsymbol{y}_1,\boldsymbol{y}_2,...,\boldsymbol{y}_n\}$

for $t \leftarrow 1$ to n **do**

Calculate the gates outputs through Eq (8)–(13);

Update the forward and backward hidden layers states through Eqs (24) and (25);

Obtain the actual output $\boldsymbol{Y}_t$ of the DI-BiLSTM through Eq (26);

end for

Calculate the training error *MSE* through Eq (27);

Update all the weights of the BiLSTM layers by the Adam optimizer;

Repeat until the training *MSE* converges;

Test the DBI-BiLSTM on testing datasets.

3. Results

In this section, we simulate four real-world MTS datasets to evaluate the performance of the employed DBI-BiLSTM network. These datasets include the heat exchangers (HX) system [39], the small-medium-large (SML) system [40], the bike sharing (BS) dataset [41], and the metro interstate traffic volume (MITV) dataset from the UCI machine learning repository. The data was divided into three sections for each MTS simulation experiment: training set, validation set, and testing set. The neuron number and layers in different DBN and BiLSTM modules are jointly determined based on the length and dimensionality of the MTS data. In this study, we used the validation data. We identified the hyperparameters of the DBI-BiLSTM model, such as the number of DBN layers (M_d), the neurons number per RBM (N_d), the number of BiLSTM layers (M_h), and the number of neurons per BiLSTM recurrent layer (N_h) by the grid search or greedy search algorithm until the established DBI-BiLSTM's validation MSE is minimized. All experiments were performed on an Intel-based Core i5-8265U (1.60 GHz CPU with 8 GB RAM). Table 1 summarizes the hyperparameters used in the simulation experiments.

The DBI-BiLSTM model is assessed through the following loss functions: percentage improvement (*IM%*), normalized mean squared error (*NMSE*), mean absolute error (*MAE*), and symmetric mean absolute percentage error (*SMAPE*):

$$NMSE=\sum_{t=1}^{n}\frac{(\boldsymbol{y}_t-\hat{\boldsymbol{y}}_t)^2}{n\sigma^2}, \tag{28}$$

$$MAE=\frac{1}{n}\sum_{t=1}^{n}|\boldsymbol{y}_t-\hat{\boldsymbol{y}}_t|, \tag{29}$$

$$SMAPE=\frac{100}{n}\sum_{t=1}^{n}\frac{|\boldsymbol{y}_t-\hat{\boldsymbol{y}}_t|}{(|\hat{\boldsymbol{y}}_t|+|\boldsymbol{y}_t|)/2}, \tag{30}$$

$$IM\%=\frac{NMSE_{LSTM}-NMSE_{Model}}{NMSE_{LSTM}}\times 100\%, \tag{31}$$

where $\hat{\boldsymbol{y}}_t$ represents the actual forecasted signal, $\boldsymbol{y}_t$ denotes the wanted output signal, σ^2 is the variance of $\boldsymbol{y}_t$, and n denotes the length of $\boldsymbol{y}_t$ $NMSE_{LSTM}$ represents the prediction performance of a single-layer LSTM, while $NMSE_{Model}$ indicates the forecasting performance of the comparison method. The *IM*% value denotes the percentage improvement in performance achieved by different prediction models compared to a single-layer BiLSTM model. In the following experiments, the mean *NMSE*, *MAE*, and *SMAPE* errors for testing are acquired by experimentally averaging ten times on the MTS datasets.

To demonstrate the performance and efficacy of the DBI-BiLSTM model on different MTS, five ablation models were employed as a single-layer LSTM, a shallow BiLSTM (single-layer), a multi-layer LSTM, a multi-layer BiLSTM, and a DI-BiLSTM. These ablation models were used to further illustrate the DBI-BiLSTM model's superiority. Furthermore, we evaluated and compared the performance of the proposed DBI-BiLSTM model with several classical and state-of-the-art models, including SVR, GBRT, DBN-ANN, Elman [42], gated recurrent units (GRU) [43], ESN, attention-LSTM [44], stacked bidirectional and unidirectional LSTM (SBU-LSTM) [45], EA-LSTM [27], and LSTM-FCN [46], where the SVR and GRRT are typical ML models used for comparison, DBN-ANN and Elman are classical ANN and single-layer RNN models for comparison, GRU is a gate-based single-layer RNN similar to LSTM, and ESN is an RNN with a different training algorithm compared with LSTM. Similar to DBI-BiLSTM, attention-LSTM, SBU-LSTM, EA-LSTM, and LSTM-FCN are recently proposed structural improvements to the original LSTM, which are multi-layer LSTM models that acquire features from the original data by adding some feature representation layers before the multi-layer LSTM layers. Overall, this study employed various models to demonstrate the effectiveness of the proposed DBI-BiLSTM model, which was then compared to classical and state-of-the-art models to prove its relative performance.

3.1. Parameter settings

In the following experiments, several parameters were identified as significantly impacting the model's performance, including M_d, M_h, N_d, and N_h. The parameters that resulted in the minimum loss for validating *NMSE* were chosen as the ultimate values for the model. Figures 6–9 illustrate the validation *NMSE* values obtained by varying the M_d, M_h, N_d, and N_h parameters for the four MTS tasks. Table 1 summarizes the ultimate parameter values for the employed DBI-BiLSTM models. To ensure a fair comparison for testing, the parameters of the LSTM-based models used for comparison, including hidden layer neurons number, learning method, activation or fire function, original training rate, and dropout constant, were set equal to the values of the parameters in the DBI-BiLSTM model.

Table 1. Parameter values for the DBI-BiLSTM models of different MTS simulations.

Tasks	Batch size	DBN layers (M_d)	Neurons in RBM (N_d)	Hidden neurons of BiLSTM (N_h)	BiLSTM layers (M_h)	Activation function ψ
HX	100	2	10	50	3	tanh
SML	100	3	12	40	3	tanh
BS	100	2	8	35	4	tanh
MITV	100	3	10	40	3	tanh

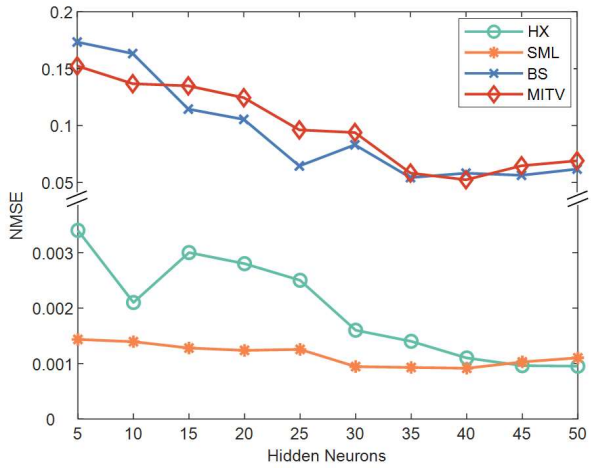

Figure 6. *NMSE* for validating that vary with different BiLSTM hidden neurons.

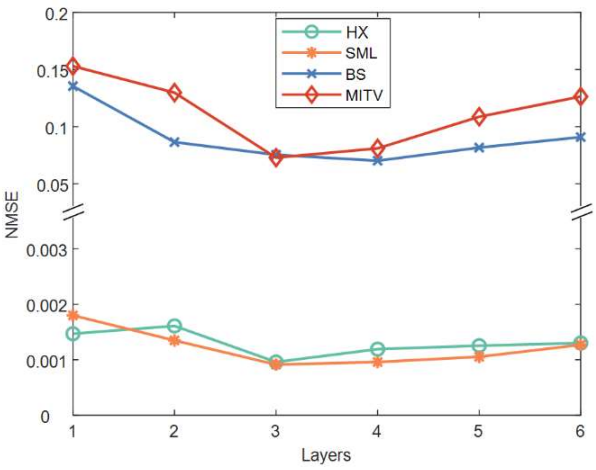

Figure 7. *NMSE* for validating that vary with different BiLSTM layers.

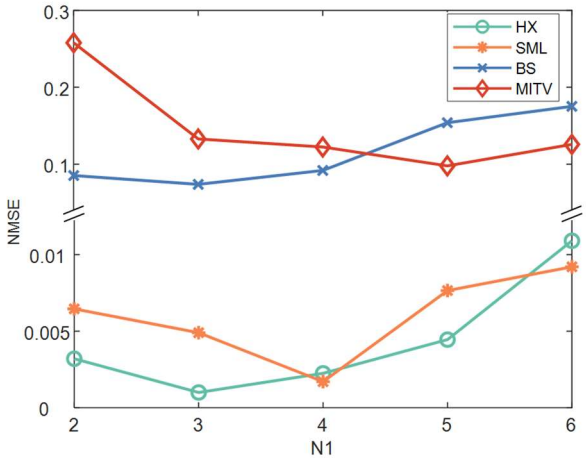

Figure 8. *NMSE* for validating that vary with different N_1.

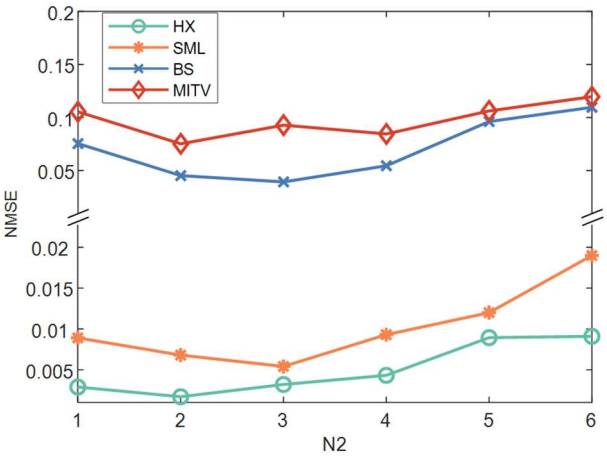

Figure 9. *NMSE* for validating that vary with different N_2.

3.2. Heat exchangers system

The HX task used in this paper is from the literature [39]. HX is a complex non-linear MTS task involving the effective heat exchange between two streams utilizing the temperature difference. The task presents significant difficulties, including flow turbulence, fluid flow geometry, and complex thermal behavior. Figure 10 illustrates the whole sequential data of the HX datasets, which is comprised of a total of 4,000 datasets. First, the HX data are normalized between −1 and 1 ([−1, 1]); 4,000 length data steps are taken out for modeling and testing the DBI-BiLSTM model, of which the first 2,000 steps are used for training, the next 1,000 steps are used as validation set for parameter selection, and the last 1,000 steps are used for performance testing of the DBI-BiLSTM. The grid search method is used to select the optimal parameters (M_d, M_h, N_d and N_h), as shown in Table 1. Then, the DBI-BiLSTM model is constructed based on the parameters in Table 1; the

initial learning rate is set to 0.001, and the bias of each BiLSTM is set to 1. Meanwhile, a dropout probability of 0.2 is set for the BiLSTM layers to ensure that the DBI-BiLSTM does not overfit the time series datasets.

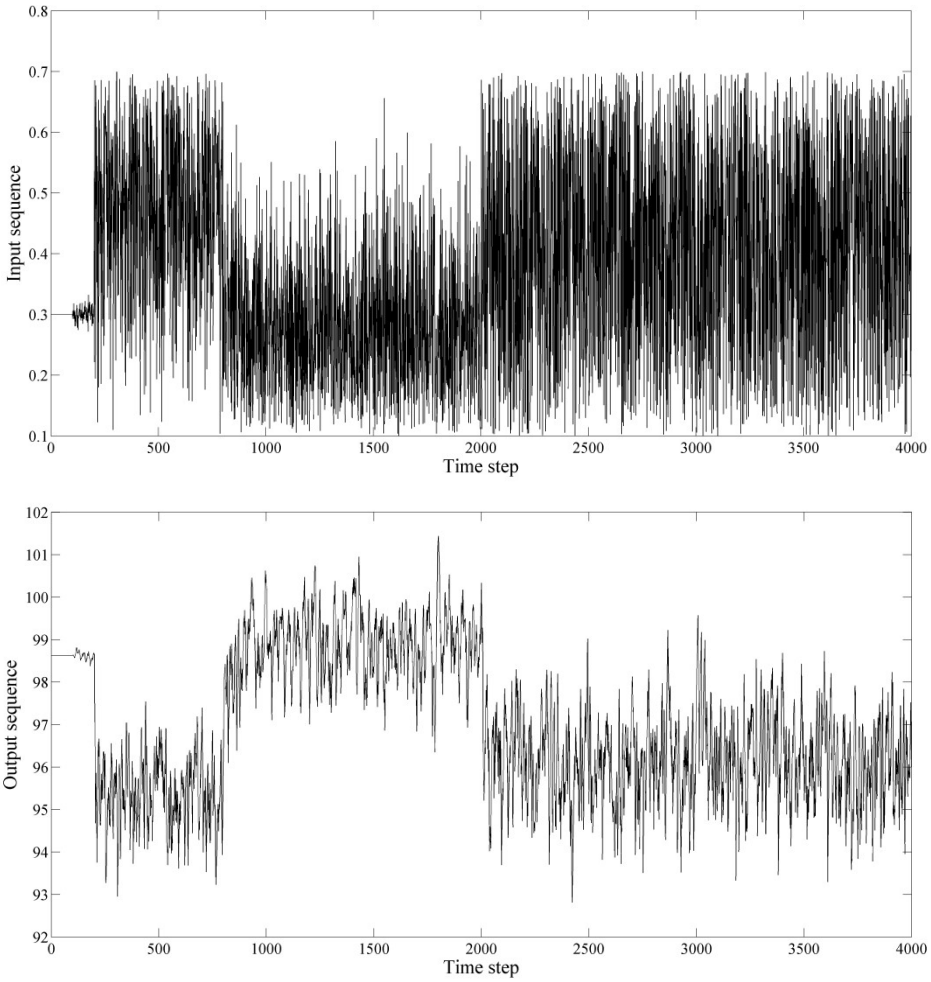

Figure 10. Sequential trend of the HX task.

The experimental results of the DBI-BiLSTM model were evaluated against several ablation LSTM-based models, including a shallow BiLSTM (one layer), a multi-layer LSTM, a multi-layer BiLSTM, and a DI-BiLSTM. To ensure a fair comparison, the same parameter values were used for the ablation models as those used for the DBI-BiLSTM model, as shown in Table 1. The forecasting outcomes acquired by DBI-BiLSTM and shallow BiLSTM for the HX dataset over a selected length of 200 datasets are shown in Figure 11. The loss error of *NMSE*, *MAE*, *SMAPE*, and enhancement *IM*% for the shallow BiLSTM, multi-layer LSTM, multi-layer BiLSTM, DI-BiLSTM, and DBI-BiLSTM for the HX experiment are summarized in Table 2.

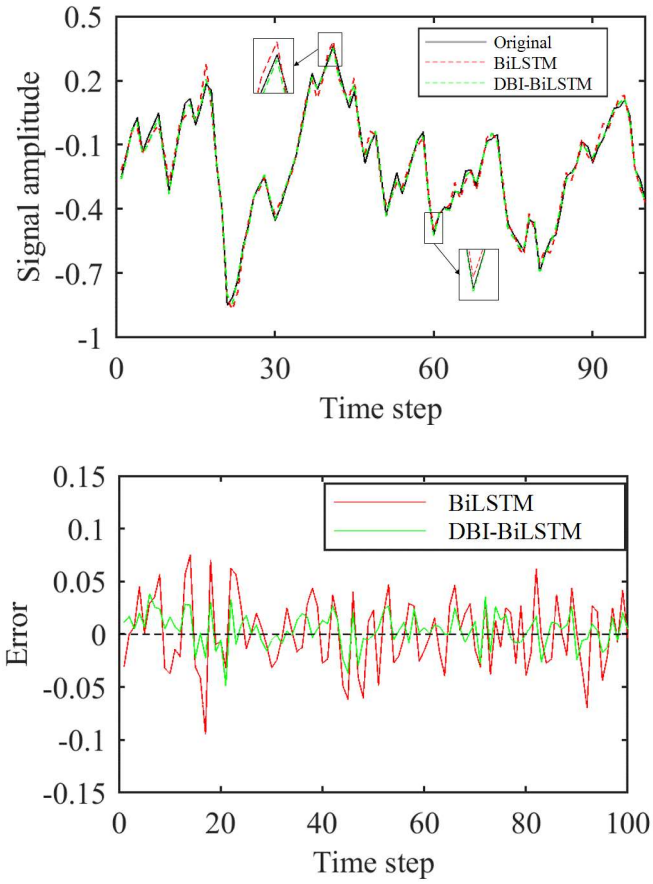

Figure 11. Fitting and test absolute error of DBI-BiLSTM and shallow BiLSTM for the HX experiment.

3.3. Small medium large system

The SML dataset is an open dataset from the UCI machine learning repository that collected data from a monitor system mounted in an intelligent house. SML data are sampled at one-minute intervals and smoothed by averaging the data over 15 minutes (Open-source download link: https://archive.ics.uci.edu/dataset/274/sml2010). This dataset includes sensor readings such as weather forecast temperature, relative humidity, lighting, rain, sun dusk, wind speed, sunlight in the west, east and south facades, sun irradiance, outdoor temperature, outdoor relative humidity, and room temperature, which is the target output for this experiment. Figure 12 illustrates the input and output sequential data of the SML benchmark. The SML data are normalized in this experiment between −1 and 1 ([−1, 1]). The total length of the SML dataset used in this paper is 4137. The first 3,000 items are used for training, the next 537 steps are used as validation set for parameter selection, and the last 6,00 steps are used for performance testing of the DBI-BiLSTM. The grid search method is used to select the optimal parameters (M_d, M_h, N_d and N_h), as shown in Table 1. Then, the DBI-BiLSTM model is constructed based on the parameters in Table 1; the initial learning rate, the bias of each BiLSTM, and the dropout probability are similarly set in the HX experiment.

To evaluate the effectiveness of the proposed DBI-BiLSTM model, we compare its performance with several ablation LSTM-based models, as described previously. The ablation models' parameters

are chosen according to the values of DBI-BiLSTM (see Table 1) to ensure a fair comparison. The prediction results acquired by DBI-BiLSTM and a shallow BiLSTM over a selected length of 200 testing signals for the SML benchmark are presented in Figure 13. Table 2 summarizes the *NMSE*, *MAE*, *SMAPE*, and enhancement *IM*% of the shallow BiLSTM, multi-layer LSTM, multi-layer BiLSTM, DI-BiLSTM, and DBI-BiLSTM models for the SML benchmark.

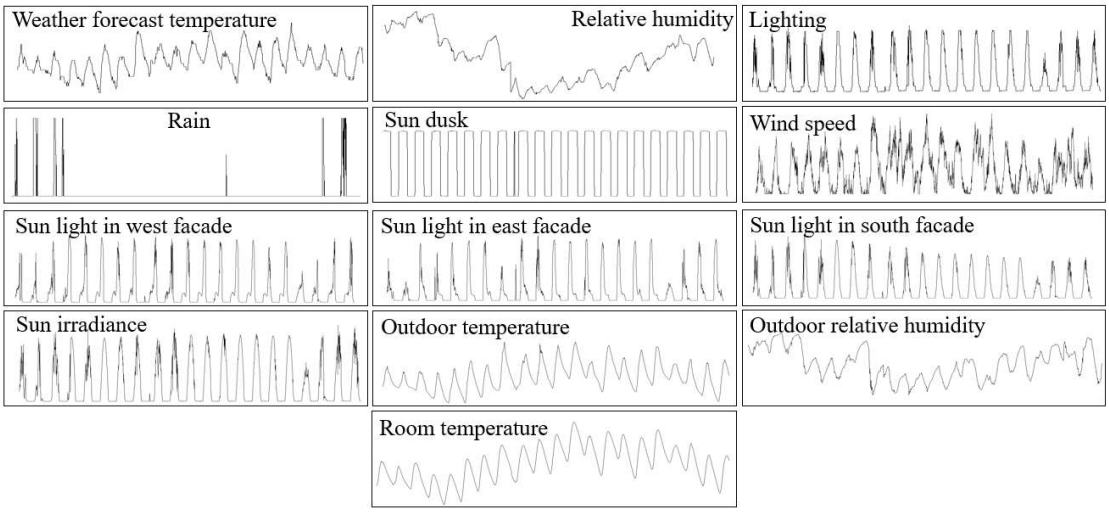

Figure 12. Sequential trend of the SML task.

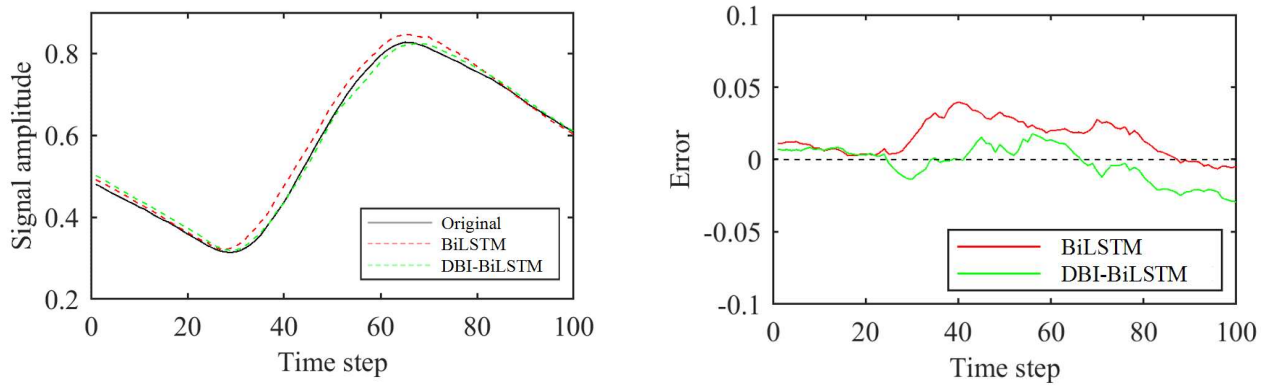

Figure 13. Fitting and test absolute error of DBI-BiLSTM and shallow BiLSTM for the SML benchmark.

3.4. Bike sharing dataset

The BS dataset is an open dataset from the UCI machine learning repository that represents a new generation of traditional bike rental systems (Open-source download link: https://archive.ics.uci.edu/dataset/275/bike+sharing+dataset). The BS dataset represents a new generation of conventional bike rental systems. This automated system allows for membership eligibility, bike rentals, and returns to be completed entirely through mechanical processes. The dataset consists of a two-year usage record (2011–2012) of the capital bike-share system and

corresponding weather and seasonal information. Sensor data includes working day, weather, temperature, feeling temperature, humidity, wind speed, and the total number of bicycle rentals per hour. The target value to be predicted in this experiment is the total number of bicycle rentals per hour. Figure 14 displays the sequential trend (input and output) of the BS benchmark. In this experiment, the BS data are normalized between −1 and 1 ([−1, 1]), as described previously. The total length of the BS dataset used in this paper is 17,389, but only 5,000 were used for experiments due to a periodic pattern in the data. The first 3000 items are used for training, the next 1000 steps are used as validation sets for parameter selection, and the last 1000 data are used for performance testing of the DBI-BiLSTM. The grid search method is used to select the optimal parameters (M_d, M_h, N_d and N_h), as shown in Table 1. Then, the DBI-BiLSTM model is constructed based on the parameters in Table 1; the initial learning rate, the bias of each BiLSTM, and the dropout probability are similarly set in the HX and SML experiments.

The performance of the DBI-BiLSTM model was simulated using the same ablation LSTM-based models, as described earlier. The parameters used in the ablation model were selected according to the values of DBI-BiLSTM model (see Table 1). The forecasting performance of the shallow BiLSTM and DBI-BiLSTM models for the BS task over a 200-length testing dataset are shown in Figure 15. The performance of the shallow BiLSTM, multi-layer LSTM, multi-layer BiLSTM, DI-BiLSTM, and DBI-BiLSTM models for the BS task in terms of *NMSE*, *MAE*, *SMAPE*, and *IM*% is shown in Table 2.

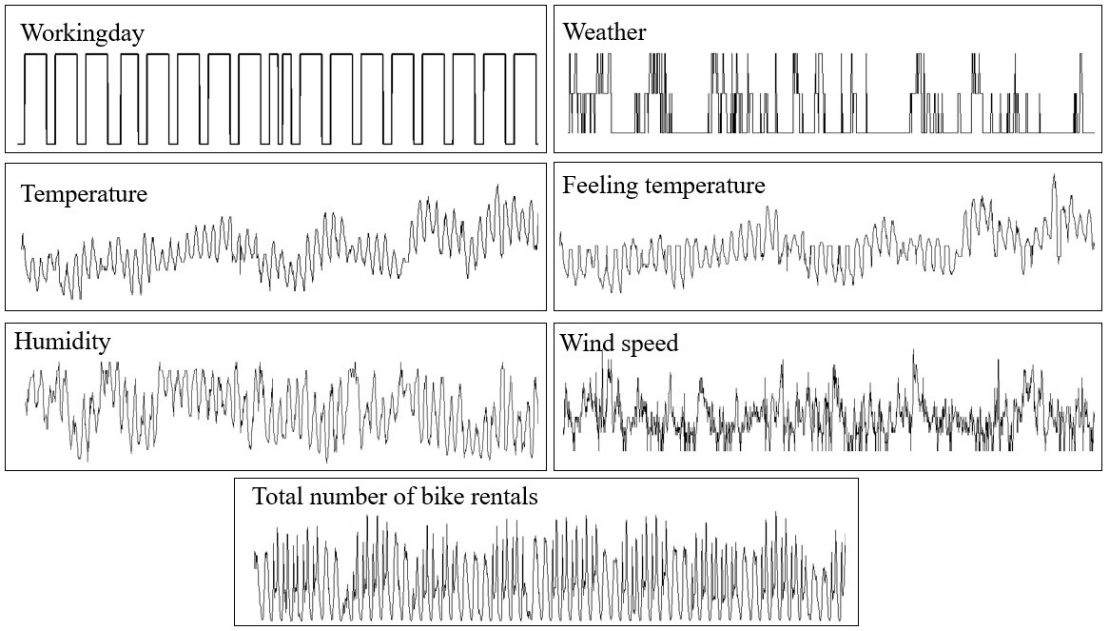

Figure 14. Sequential trend of the BS benchmark.

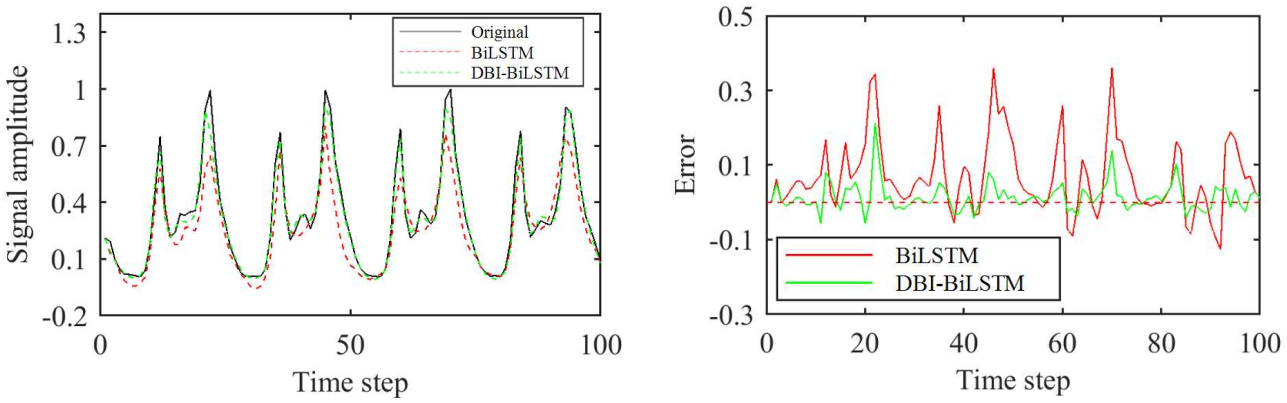

Figure 15. Fitting and test absolute error of DBI-BiLSTM and shallow BiLSTM for the BS benchmark.

3.5. Metro interstate traffic volume dataset

The MITV dataset is an open dataset from the UCI machine learning repository that represents a situation of MTS regression, where the employed network aims to forecast the continuous variables (Open download link: https://archive.ics.uci.edu/dataset/492/metro+interstate+traffic+volume). This dataset includes hourly traffic volume data for the MN DoT ATR station 301, located approximately halfway between Minneapolis and St. Paul, MN. The sensor data includes holidays, temperature, rainfall, snowfall, percentage of cloud cover, weather descriptions, and hourly traffic volume. The hourly traffic volume serves as the target predicted variable in this experiment. The input and output sequential data for the MITV task are displayed in Figure 16. In this experiment, the MITV data are normalized between −1 and 1 ([−1, 1]), as described previously. The total length of the BS dataset used in this paper is 48,204, but only 10,000 were used for experiments due to a robust periodic pattern. The first 6,000 items are used for training, the next 2,000 steps are used as validation sets for parameter selection, and the last 2,000 steps are used for performance testing of the DBI-BiLSTM. The grid search method is used to select the optimal parameters (M_d, M_h, N_d and N_h), as shown in Table 1. Then, the DBI-BiLSTM model is constructed based on the parameters in Table 1; the initial learning rate, the bias of each BiLSTM, and the dropout probability are similarly set in the HX, SML, and BS experiments.

As with the previous experiments, we simulated the experimental results of the DBI-BiLSTM model using the same ablation LSTM-based models. The parameters for the ablation model were set to the same values as the DBI-BiLSTM model, as shown in Table 1. Figure 17 displays the prediction performance over a 200-length testing dataset for both the DBI-BiLSTM and shallow BiLSTM models for the MITV task. Table 2 shows the *NMSE*, *MAE*, *SMAPE*, and enhancement *IM*% for the shallow BiLSTM, multi-layer LSTM, multi-layer BiLSTM, DI-BiLSTM, and DBI-BiLSTM models for the MITV task.

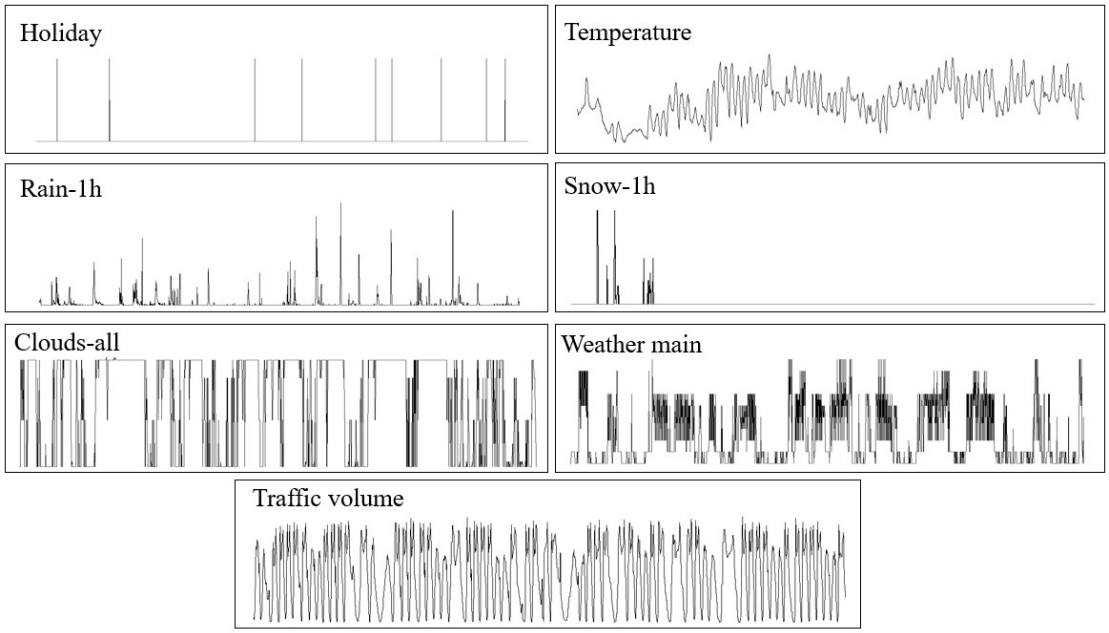

Figure 16. Sequential trend of the MITV benchmark.

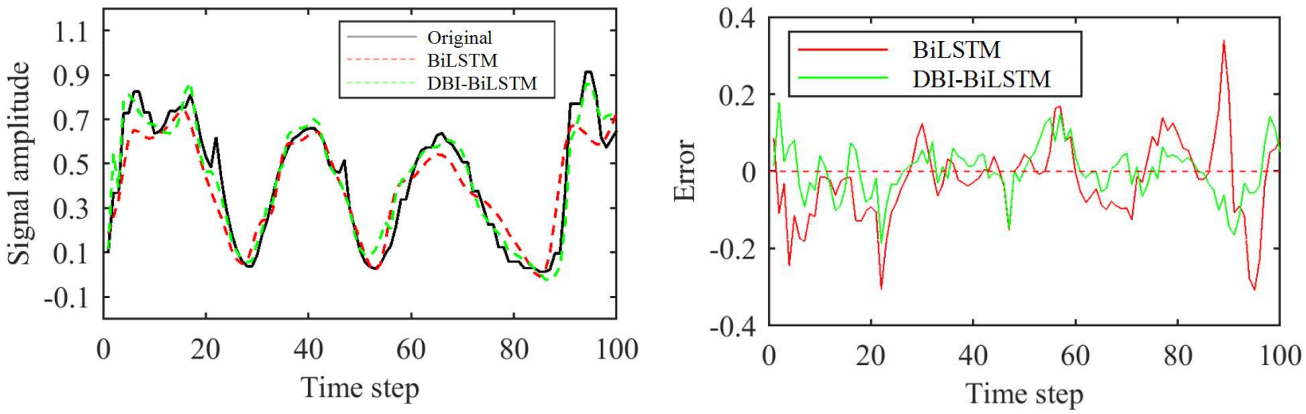

Figure 17. Fitting and test absolute error of DBI-BiLSTM and shallow BiLSTM for the MITV task.

3.6. Comparison of DBI-BiLSTM and various other MTS models

To analyze and validate the impact of DBN layers and deep chained structure in the performance of DBI-BiLSTM, comparative ablation models, including single-layer and multi-layer LSTM-based models, are used to test the selected time series datasets. The forecasting results for the comparison of DBI-BiLSTM and the ablation models are shown in Table 2. The running time(s) performance in Table 2 is the total time of the training and testing process.

Table 2. The testing *NMSE*, *MAE*, *MAPE*, *IM%* and running time of DBI-BiLSTM and ablation models for the four MTS forecasting benchmarks.

Benchmarks	Testing Performance	Proposed and ablation models					
		One layer LSTM	One layer BiLSTM	Multi-layer LSTM	Multi-layer BiLSTM	DI-BiLSTM	DBI-BiLSTM
HX	*NMSE*	0.0658	0.0282	0.0273	0.0215	0.0158	**0.0096**
	MAE	0.0527	0.0332	0.0324	0.0279	0.0193	**0.0138**
	MAPE	52.976	28.612	27.492	24.787	21.035	**15.418**
	IM%	–	57.14	58.51	67.32	77.61	**85.41**
	Time(s)	30.2	59.1	76.1	128.9	129.1	130.7
SML	*NMSE*	0.2451	0.1582	0.1295	0.1131	0.0801	**0.0601**
	MAE	0.0623	0.0361	0.0304	0.0285	0.0199	**0.0149**
	MAPE	17.253	13.571	9.417	8.836	8.036	**6.713**
	IM%	–	35.45	47.16	53.85	66.94	**75.47**
	Time(s)	34.4	52.7	84.3	145.2	145.8	147.8
BS	*NMSE*	0.3712	0.3276	0.3033	0.1751	0.1502	**0.1423**
	MAE	0.1053	0.0987	0.0891	0.0561	0.0507	**0.0501**
	MAPE	58.468	53.169	49.051	37.849	36.306	**35.513**
	IM%	–	11.74	18.29	52.82	56.91	**61.66**
	Time(s)	27.1	40.5	87.5	146.6	147.1	148.2
MITV	*NMSE*	0.4723	0.4587	0.4453	0.4229	0.3472	**0.3272**
	MAE	0.1352	0.1236	0.1198	0.1125	0.0839	**0.0897**
	MAPE	41.821	40.243	38.602	35.145	34.070	**32.493**
	IM%	–	2.87	5.71	10.46	20.01	**30.72**
	Time(s)	38.4	59.3	95.9	166.3	167.6	168.4

From Table 2, we can see that the running time of the bidirectional ablation models is significantly longer than that of the unidirectional LSTM models, and the running time of the multi-layer ablation model is longer than that of the single-layer ablation model. The computational complexity of our proposed DBI-BiLSTM model is comparable to that of the multi-layer BiLSTM, which suggests that the DBN module and the stack chained structure in the DBI-BiLSTM do not significantly enhance the computational burden of the multi-layer BiLSTM.

4. Discussion and statistical analysis

To comprehensively assess and evaluate the performance of the DBI-BiLSTM model in MTS forecasting, we conducted a comparative test with several fundamental models. These models include conventional SVR, GBRT, DBN-ANN, traditional Elman RNN, classical variant GRU of RNN, and ESN. Additionally, we compared the DBI-BiLSTM model with several lately proposed LSTM models, including the EA-LSTM model that can be adjusted by evolutionary computation, the SBU-LSTM model with deep stack structure, and the attention-LSTM model. To ensure a fair comparison, we set all parameters used in the DBN-based and LSTM-based models to the same values as those used in the DBI-BiLSTM model (developed as Table 1). The performance

comparison between DBI-BiLSTM and many other MTS forecasting models is presented in Table 3. The running time(s) performance in Table 3 is the total time of the training and testing process. Furthermore, we conducted a 10-fold cross-validation experiment on four selected MTS datasets and obtained *NMSE* performance results. These results are shown in Figure 18 in the form of a box plot.

Table 3. The performance comparison of DBI-BiLSTM and various MTS forecasting models.

Forecasting models	Test Performance	MTS Datasets			
		HX	SML	BS	MITV
SVR	*NMSE*	1.3657	1.5824	0.6700	0.5766
	MAE	0.2967	0.1969	0.1007	0.1311
	SMAPE	121.368	46.960	60.795	52.047
	IM%	−1975.53	−545.61	−80.49	−22.08
	Time(s)	21.2	35.6	38.3	50.1
GBRT	*NMSE*	1.4174	1.7838	0.2902	0.4502
	MAE	0.2850	0.1978	0.0712	0.1315
	SMAPE	125.196	47.867	33.743	43.491
	IM%	−2054.10	−627.78	21.82	4.68
	Time(s)	23.2	38.1	40.8	53.7
DBN-ANN	*NMSE*	1.0372	1.7490	0.8257	0.5203
	MAE	0.1823	0.2675	0.1223	0.1396
	SMAPE	84.619	54.822	66.427	52.225
	IM%	−1476.29	613.58	−122.44	−10.16
	Time(s)	16.1	20.3	25.1	34.8
Elman	*NMSE*	0.0892	0.2201	0.3902	0.4554
	MAE	0.0813	0.0586	0.1023	0.1481
	SMAPE	38.305	8.292	60.581	51.307
	IM%	−33.56	10.20	−5.11	3.58
	Time(s)	18.3	26.6	30.1	43.3
GRU	*NMSE*	0.0832	0.2045	0.3648	0.4352
	MAE	0.0724	0.0519	0.0915	0.1349
	SMAPE	36.363	7.203	59.205	50.275
	IM%	−26.44	16.56	1.72	7.855
	Time(s)	29.2	33.8	26.1	37.9
ESN (N_h = 500)	*NMSE*	0.0216	0.3060	0.4113	0.4899
	MAE	0.0263	0.0810	0.1079	0.1430
	SMAPE	25.876	9.820	65.836	52.266
	IM%	67.17	−24.84	−10.80	3.72
	Time(s)	4.2	5.6	5.8	7.1

Continued on next page

Forecasting models	Test Performance	MTS Datasets			
		HX	SML	BS	MITV
Attention-LSTM	*NMSE*	0.0480	0.1858	0.1923	0.3871
	MAE	0.0401	0.0396	0.0728	0.1254
	SMAPE	32.372	8.710	47.726	46.289
	IM%	27.05	24.19	48.19	18.04
	Time(s)	138.1	168.7	186.5	260.7
SBU-LSTM	*NMSE*	0.0383	0.1926	0.1830	0.3995
	MAE	0.0361	0.0446	0.0604	0.1129
	SMAPE	30.845	7.576	47.157	40.482
	IM%	41.79	21.42	50.70	15.41
	Time(s)	127.8	140.9	145.5	160.9
EA-LSTM	*NMSE*	0.0210	0.0791	0.1654	0.3427
	MAE	0.0221	0.0252	0.0608	0.1262
	SMAPE	17.845	8.435	48.191	45.577
	IM%	68.08	67.76	55.44	27.44
	Time(s)	156.6	191.8	212.3	307.9
LSTM-FCN	*NMSE*	0.0279	0.1135	0.2131	0.3722
	MAE	0.0292	0.0314	0.0766	0.1335
	SMAPE	23.467	8.290	49.478	46.270
	IM%	57.60	53.69	42.58	21.19
	Time(s)	161.8	205.6	240.4	319.2
DBI-BiLSTM	*NMSE*	**0.0096**	**0.0601**	**0.1423**	**0.3272**
	MAE	**0.0138**	**0.0149**	**0.0501**	**0.0897**
	SMAPE	**15.418**	**6.713**	**35.513**	**32.493**
	IM%	**85.41**	**75.47**	**61.66**	**30.72**
	Time(s)	130.7	147.8	148.2	168.4

As can be seen from the running time in Table 3, the computational complexity of the DBI-BiLSTM model is significantly higher than that of the statistical models and the single-layer RNN models. However, the computational complexity of the DBI-BiLSTM model is still acceptable compared with the recently proposed multi-layer LSTM model (Attention-LSTM, SBU-LSTM, EA-LSTM, and LSTM-FCN) based on feature learning.

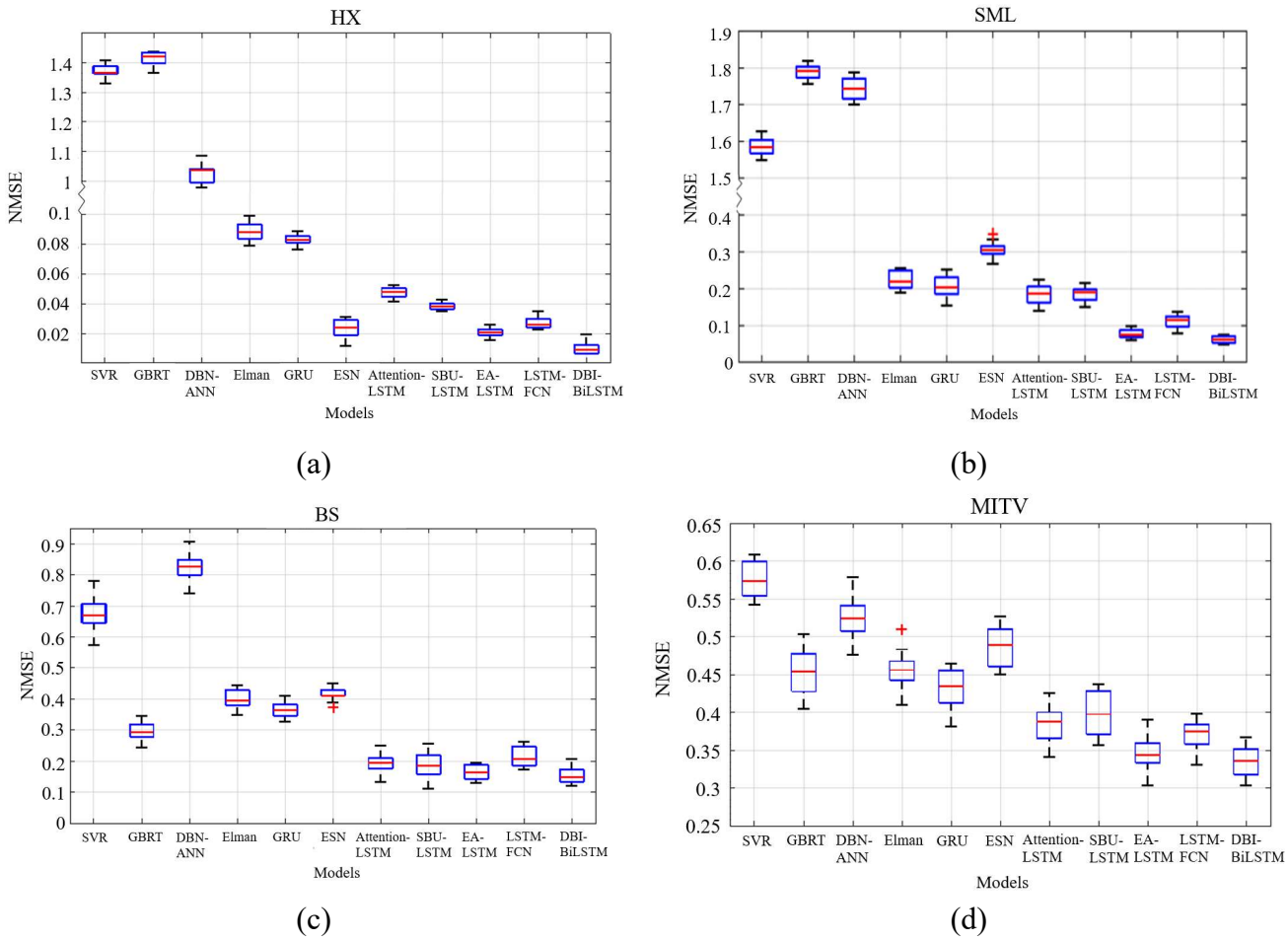

Figure 18. Box plot of test *NMSE* using DBI-BiLSTM and many other MTS forecasting models with 10-fold cross validation.

5. Discussion and statistical analysis

Heteroscedasticity, in contrast to homoscedasticity, is the case when the variance of the stochastic error items of the fitting network is not constant. Specifically, if the variance of the error term changes with changes in the independent variable, we have either variable variance or heteroscedasticity. Figure 19 displays the results of the heteroscedasticity test for each MTS dataset. According to the change of the residuals with the fitness numbers in Figure 19, it can be concluded that the values are typically spread around 0, and the variance keeps significantly steady with increasing fitness values. Figure 19 shows little heteroskedasticity among the DBI-BiLSTM models, and the homoskedasticity can be maintained.

Given that the feature vectors output by DBN are typically more complex and have a higher dimension, it becomes necessary to incorporate the GSA process. This is because GSA provides a framework for attributing the uncertainty in the model's output to various sources of uncertainty in the input factors of the model. Figure 20 illustrates the schematic chart of the total SI acquired by the GSA algorithm for the four MTS benchmarks. The SI in Figure 20 enables us to identify the output of the DBN with high and low sensitivity to DBI-BiLSTM. This information helps us to classify the input data of DI-BiLSTM into major and minor features.

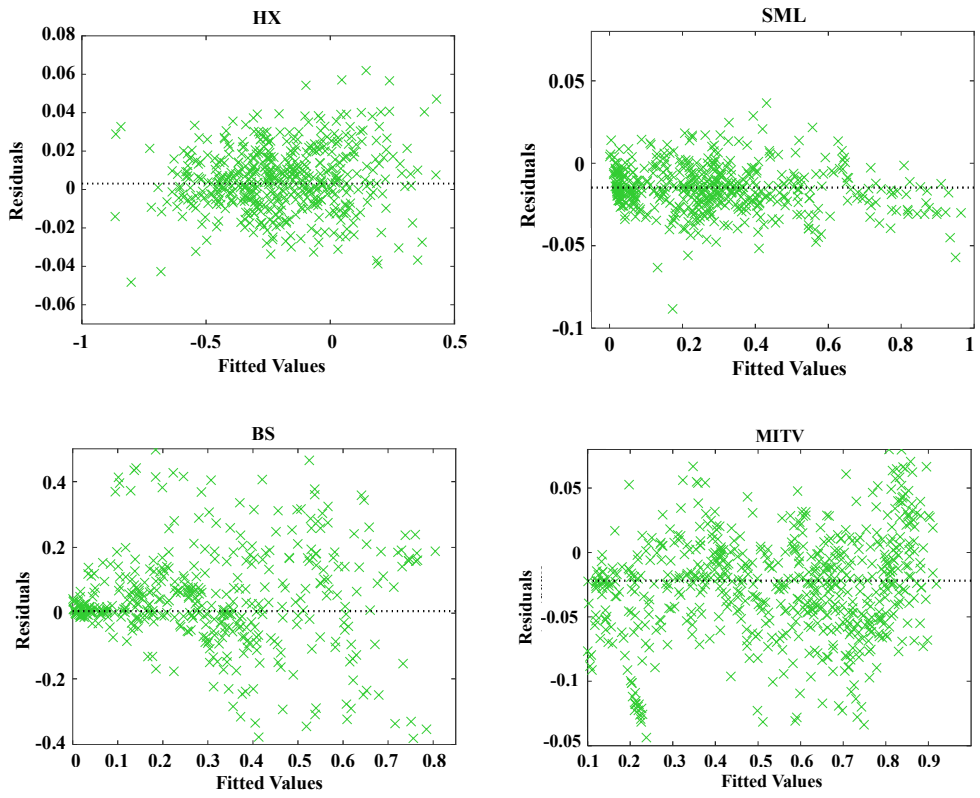

Figure 19. Performance and results of the heterogeneity test for the DBI-BiLSTM prediction model.

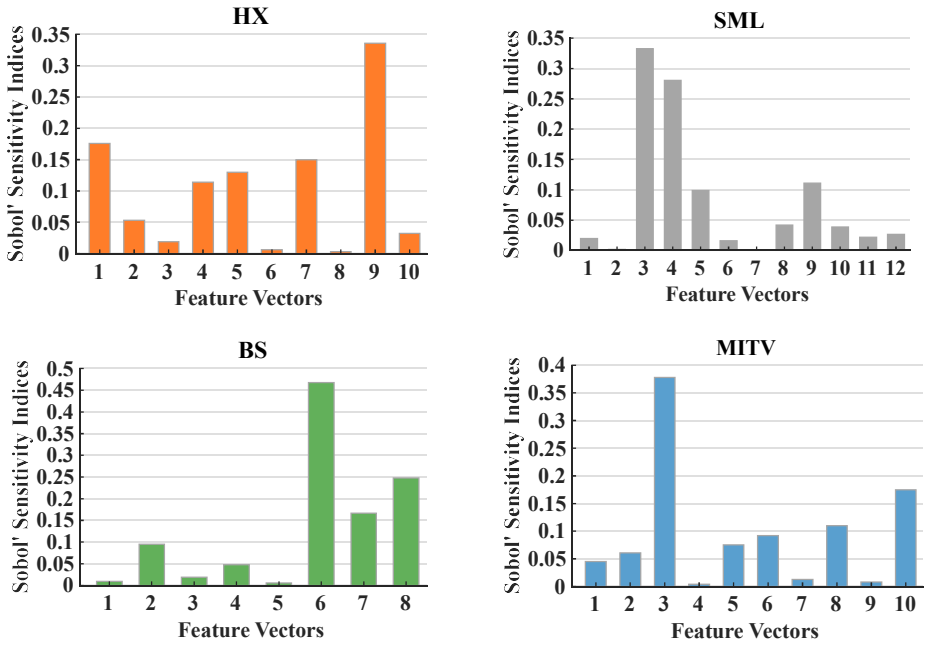

Figure 20. SA of DBN's output to DBI-BiLSTM performance.

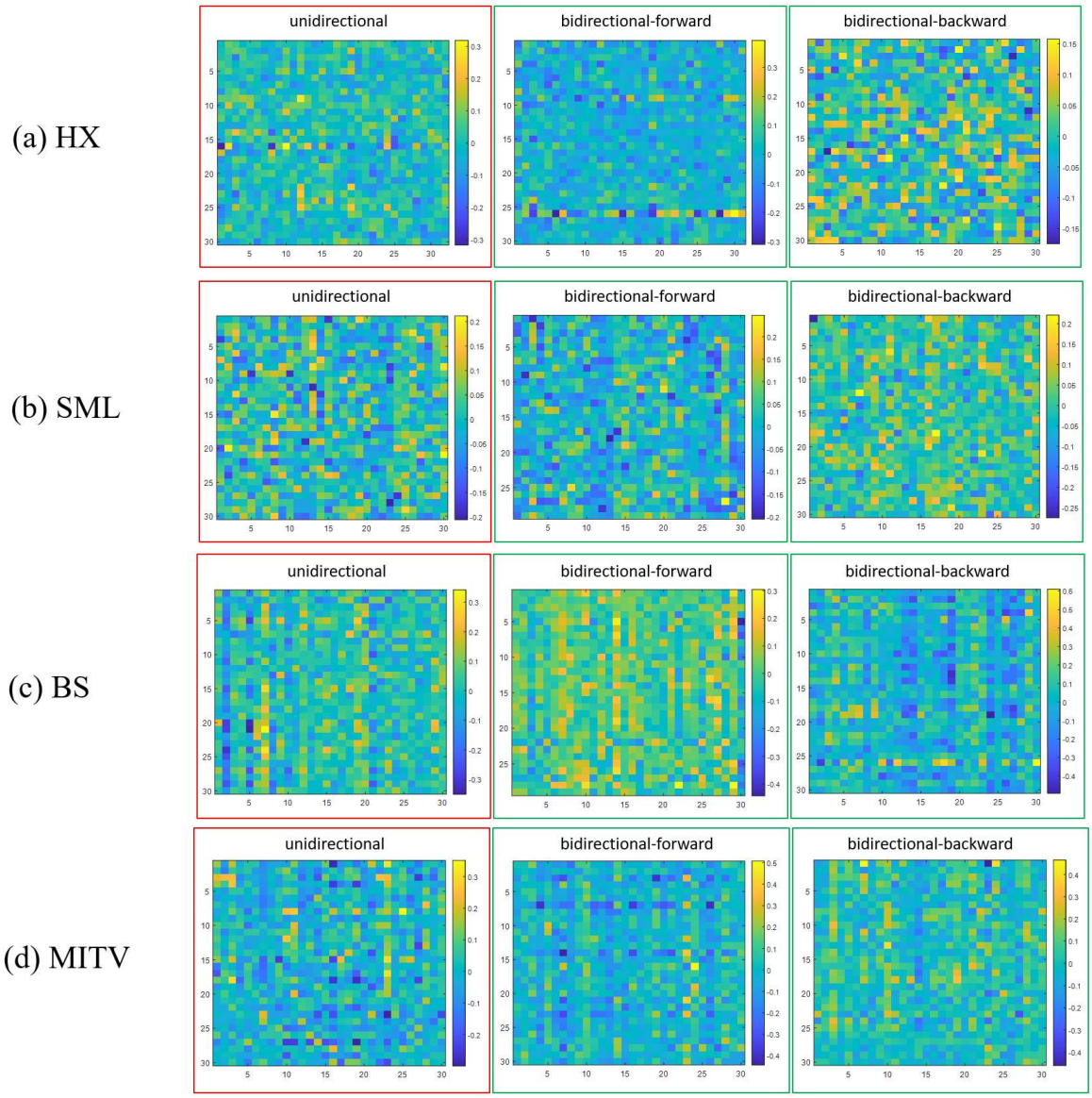

Figure 21. Heatmaps analysis presentation of the partial output weights (left part is unidirectional).

To demonstrate the different characteristics of the one-direction prediction model (DBI-LSTM) and the bidirectional forecasting model (DBI-BiLSTM), as well as the differences in synapsis distribution resulting from the two-direction transmission, partial output weight heat maps of DBI-LSTM (unidirectional) and DBI-BiLSTM (bidirectional-forward and bidirectional-backward) are plotted. Figure 21 displays these heatmaps, where the red rectangle on the left represents the weight heatmap for DBI-LSTM, and the green boxes represent the forward and backward heatmaps of DBI-BiLSTM. As indicated in Table 2 and Figure 21, bidirectional propagation outperforms one-direction propagation for shallow and multi-layer LSTM structures. Additionally, Figure 21 shows that the weights from the recurrent layer to the output layer of the one-direction LSTM are mostly uniformly distributed within the symmetric interval of 0, indicating that the features in the one-direction LSTM are transmitted in a single direction. Conversely, the forward and backward output synapses of the bidirectional LSTM models are distinguishable in Figure 21, meaning that if

most of the forward weights are more significant than zero, then most of the backward consequences are more minor than zero, and vice versa. Therefore, it clearly illustrates that the DBI-BiLSTM model can effectively learn and train the characteristics of the MTS in different directions, further increasing the interpretability of the DBI-BiLSTM.

6. Conclusions

This paper proposes a novel, profound, improved BiLSTM network for MTS forecasting. The proposed network is comprised of a DBN, a GSA module, and a stacked BiLSTM network. The DBN layer is used for unsupervised feature learning, and the learned features based on GSA are divided into major and minor parts, which are then fed into each part of the BiLSTM modules. The different layers of the BiLSTM learn the features of the input data and integrate the final output results. The DBI-BiLSTM network leverages the BiLSTM and RBM to thoroughly understand the transient information of the signals of different layers, collecting diverse and rich information in forward and backward directions. Four real-world MTS datasets were applied to test the performance of DBI-BiLSTM. Comparative experimental results on the MTS tasks demonstrate that the proposed DBI-BiLSTM outperforms some conventional ML forecasting models, several classical RNN-based MTS prediction models, and a few recently proposed LSTM-based models. The percentage improvement of DBI-BiLSTM compared with the original shallow LSTM on the four MTS datasets is 85.41, 75.47, 61.66 and 30.72%, respectively.

The proposed DBI-BiLSTM can effectively extract features from MTS data and learn features sufficiently in a multi-layer chained structure to improve performance. Additionally, the DBI-BiLSTM is a more robust and flexible in forecasting MTS datasets. The visualization and interpretation of the input and output weights reflect the proposed model's reasonability. The ideas presented in this paper can also be used in other neural networks. Future work will involve evaluating other tasks with the proposed method and using advanced optimization algorithms to optimize network parameters and improve training efficiency.

Use of AI tools declaration

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

Acknowledgments

The authors gratefully acknowledge the support of the following foundations: Research Project of State Grid Ningbo Electric Power Supply Company under Grant 2022YXKJ-002, National Natural Science Foundation of China (62002183), Zhejiang Provincial Natural Science Foundation of China (LQ20F020012).

Conflict of 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.

References

1. Y. Liu, H. Yang, S. Gong, Y. Liu, X. Xiong, A daily activity feature extraction approach based on time series of sensor events, *Math. Biosci. Eng.*, **17** (2020), 5173–5189. https://doi.org/0.3934/mbe.2020280
2. H. Li, J. Tong, A novel clustering algorithm for time-series data based on precise correlation coefficient matching in the IoT, *Math. Biosci. Eng.*, **16** (2019), 6654–6671. https://doi.org/10.3934/mbe.2019331
3. H. M. Srivastava, I. C. Area Carracedo, J. L. Nieto, Power-series solution of compartmental epidemiological models, *Math. Biosci. Eng.*, **18** (2021), 3274–3290. https://doi.org/10.3934/mbe.2021163
4. M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, Q. Tian, Symbiotic graph neural networks for 3d skeleton-based human action recognition and motion prediction, *IEEE Trans. Pattern Anal. Mach. Intell.*, **44** (2021), 3316–3333. https://doi.org/10.1109/TPAMI.2021.3053765
5. M. Gan, Y. Cheng, K. Liu, G. Zhang, Seasonal and trend time series forecasting based on a quasi-linear autoregressive model, *Appl. Soft Comput.*, **24** (2014), 13–18. https://doi.org/10.1016/j.asoc.2014.06.047
6. J. Wang, S. Zhang, An improved deep learning approach based on exponential moving average algorithm for atrial fibrillation signals identification, *Neurocomputing*, **513** (2013), 127–136. https://doi.org/10.1016/j.neucom.2022.09.079
7. Y. Hu, F. Hao, C. Meng, L Sun, D. Xu, T. Zhang, Spatial general autoregressive model-based image interpolation accommodates arbitrary scale factors, *Math. Biosci. Eng.*, **17** (2020), 6573–6600. https://doi.org/ 10.3934/mbe.2020343
8. X. Yu, Z. Chen, L. Qi, Comparative study of SARIMA and NARX models in predicting the incidence of schistosomiasis in China, *Math. Biosci. Eng.*, **16** (2019), 2266–2276. https://doi.org/10.3934/mbe.2019112
9. H. Tong, *Non-Linear Time Series: A Dynamical System Approach*, Oxford University Press, 1990.
10. D. T. Tran, A. Iosifidis, J. Kanniainen, M. Gabbouj, Temporal attention-augmented bilinear network for financial time-series data analysis, *IEEE Trans. Neural Networks Learn. Syst.*, **30** (2018), 1407–1418. https://doi.org/10.1109/TNNLS.2018.2869225
11. D. Li, X. Wang, J. Sun, H. Yang, AI-HydSu: An advanced hybrid approach using support vector regression and particle swarm optimization for dissolved oxygen forecasting, *Math. Biosci. Eng.*, **18** (2021), 3646–3666. https://doi.org/10.3934/mbe.2021182
12. Y. C. Kuan, C. T. Hong, P. C. Chen, W. T. Liu, C. C. Chung, Logistic regression and artificial neural network-based simple predicting models for obstructive sleep apnea by age, sex, and body mass index, *Math. Biosci. Eng.*, **19** (2022), 11409–11421. https://doi.org/10.3934/mbe.2022532
13. F. Yang, D. Wang, F. Xu, Z. Huang, K. L. Tsui, Lifespan prediction of lithium-ion batteries based on various extracted features and gradient boosting regression tree model, *J. Power Sources*, **476** (2020), 228654. https://doi.org/10.1016/j.jpowsour.2020.228654
14. Y. Liang, S. Zhang, H. Qiao, Y. Cheng, iEnhancer-MFGBDT: Identifying enhancers and their strength by fusing multiple features and gradient boosting decision tree, *Math. Biosci. Eng.*, **18** (2021), 8797–8814. https://doi.org/10.3934/mbe.2021434

15. H. Wan, S. Guo, K. Yin, X. Liang, Y. Lin, CTS-LSTM: LSTM-based neural networks for correlated time series prediction, *Knowl. Based Syst.*, **191** (2020), 105239. https://doi.org/10.1016/j.knosys.2019.105239
16. Y. Rizk, M. Awad, On extreme learning machines in sequential and time series prediction: A non-iterative and approximate training algorithm for recurrent neural networks, *Neurocomputing*, **325** (2019), 1–19. https://doi.org/10.1016/j.neucom.2018.09.012
17. Y. Liu, C. Gong, L. Yang, Y. Chen, DSTP-RNN: A dual-stage two-phase attention-based recurrent neural network for long-term and multivariate time series prediction, *Expert Syst. Appl.*, **143** (2020), 113082. https://doi.org/10.1016/j.eswa.2019.113082
18. Y. Bengio, P. Simard, P. Frasconi, Learning long-term dependencies with gradient descent is difficult, *IEEE Trans. Neural Networks*, **5** (1994), 157–166. https://doi.org/10.1109/72.279181
19. S. Hochreiter, J. Schmidhuber, Long short-term memory, *Neural Comput.*, **9** (1997), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
20. V. Eramo, F. G. Lavacca, T. Catena, P. J. P. Salazar, Application of a long short term memory neural predictor with asymmetric loss function for the resource allocation in NFV network architectures, *Comput. Networks*, **193** (2021), 108104. https://doi.org/10.1016/j.comnet.2021.108104
21. V. Eramo, T. Catena, Application of an innovative convolutional/LSTM neural network for computing resource allocation in NFV network architectures, *IEEE Trans. Network Service Manage.*, **19** (2022), 2929–2943. https://doi.org/10.1109/TNSM.2022.3142182
22. T. Catena, V. Eramo, M. Panella, A. Rosato, Distributed LSTM-based cloud resource allocation in network function virtualization architectures, *Comput. Networks*, **213** (2022), 109111. https://doi.org/10.1016/j.comnet.2022.109111
23. M. Schuster, K. K. Paliwal, Bidirectional recurrent neural networks, *IEEE Trans. Signal Process.*, **45** (1997), 2673–2681. https://doi.org/10.1109/78.650093
24. A. A. Ewees, M. A. Al-qaness, L. Abualigah, M. Abd Elaziz, HBO-LSTM: Optimized long short term memory with heap-based optimizer for wind power forecasting, *Energy Convers. Manage.*, **268** (2022), 116022. https://doi.org/10.1016/j.enconman.2022.116022
25. J. Liu, X. Huang, Q. Li, Z. Chen, G. Liu, Y. Tai, Hourly stepwise forecasting for solar irradiance using integrated hybrid models CNN-LSTM-MLP combined with error correction and VMD, *Energy Convers. Manage.*, **280** (2023), 116804. https://doi.org/10.1016/j.enconman.2023.116804
26. M. Neshat, M. M. Nezhad, N. Y. Sergiienko, S. Mirjalili, G. Piras, D. Astiaso Garcia, Wave power forecasting using an effective decomposition-based convolutional Bi-directional model with equilibrium Nelder-Mead optimizer, *Energy*, **256** (2022), 124623. https://doi.org/10.1016/j.energy.2022.124623
27. Y. Li, Z. Zhu, D. Kong, H. Han, Y. Zhao, EA-LSTM: Evolutionary attention-based LSTM for time series prediction, *Knowl. Based Syst.*, **181** (2019), 104785. https://doi.org/10.1016/j.knosys.2019.05.028
28. G. E. Hinton, R. R. Salakhutdinov, Reducing the dimensionality of data with neural networks, *Science*, **313** (2006), 504–507. https://doi.org/10.1126/science.1127647
29. X. Sun, T. Li, Q. Li, Y. Huang, Y. Li, Deep belief echo-state network and its application to time series prediction, *Knowl. Based Syst.*, **130** (2017), 17–29. https://doi.org/10.1016/j.knosys.2017.05.022

30. X. Li, Q. Liu, Y. Wu, Prediction on blockchain virtual currency transaction under long short-term memory model and deep belief network, *Appl. Soft Comput.*, **116** (2022), 108349. https://doi.org/10.1016/j.asoc.2021.108349
31. Z. Wu, Q. Li, H. Zhang, Chain-structure echo state network with stochastic optimization: Methodology and application, *IEEE Trans. Neural Networks Learn. Syst.*, **33** (2021), 1974–1985. https://doi.org/10.1109/TNNLS.2021.3098866
32. H. Zhang, B. Hu, X. Wang, J. Xu, L. Wang, Q. Sun, et al., Self-organizing deep belief modular echo state network for time series prediction, *Knowl. Based Syst.*, **222** (2021), 107007. https://doi.org/10.1016/j.knosys.2021.107007
33. G. E. Hinton, S. Osindero, Y. W. The, A fast learning algorithm for deep belief nets, *Neural Comput.*, **18** (2006), 1527–1554. https://doi.org/10.1162/neco.2006.18.7.1527
34. T. Tieleman, Training restricted Boltzmann machines using approximations to the likelihood gradient, in *Proceedings of the 25th International Conference on Machine Learning*, 2008, 1064–1071. https://doi.org/10.1145/1390156.1390290
35. A. Saltelli, P. Annoni, I. Azzini, F. Campolongo, M. Ratto, S. Tarantola, Variance based sensitivity analysis of model output, Design and estimator for the total sensitivity index, *Comput. Phys. Commun.*, **181** (2010), 259–270. https://doi.org/10.1016/j.cpc.2009.09.018
36. Y. LeCun, Y. Bengio, G. Hinton, Deep learning, *Nature*, **521** (2015), 436–444. https://doi.org/10.1038/nature14539
37. G. Kurnaz, A. S. Demir, Prediction of SO2 and PM10 air pollutants using a deep learning-based recurrent neural network: Case of industrial city Sakarya. *Urban Climate*, **41** (2022), 101051. https://doi.org/10.1016/j.uclim.2021.101051
38. D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, preprint, arXiv:1412.6980, 2014. https://doi.org/10.48550/arXiv.1412.6980
39. H. Wang, Q. Wu, J. Xin, J. Wang, H. Zhang, Optimizing deep belief echo state network with a sensitivity analysis input scaling auto-encoder algorithm, *Knowl, Based Syst.*, **191** (2020), 105257. https://doi.org/10.1016/j.knosys.2019.105257
40. F. Zamora-Martinez, P. Romeu, P. Botella-Rocamora, J. Pardo, On-line learning of indoor temperature forecasting models towards energy efficiency, *Energy Build,*, **83** (2014), 162–172. https://doi.org/10.1016/j.enbuild.2014.04.034
41. T. H. Fanaee, J. Gama, Event labeling combining ensemble detectors and background knowledge, *Progress Artif. Intell.*, **2** (2014), 113–127. https://doi.org/10.1007/s13748-013-0040-3
42. J. L. Elman, Finding structure in time, *Cognit. Sci.*, **14** (1990), 179–211. https://doi.org/10.1207/s15516709cog1402_1
43. J. Chung, C. Gulcehre, K. H. Cho, et al. Empirical evaluation of gated recurrent neural networks on sequence modeling, preprint, arXiv: 1412.3555.
44. S. Kim, M. Kang, Financial series prediction using attention LSTM, preprint, arXiv: 1902.10877.

45. Z. Cui, R. Ke, Z. Pu, Y. Wang, Stacked bidirectional and unidirectional LSTM recurrent neural network for forecasting network-wide traffic state with missing values, *Trans. Res. Part C Emerging Technol.*, **118** (2020), 102674. https://doi.org/10.1016/j.trc.2020.102674
46. F. Karim, S. Majumdar, H. Darabi, S. Harford, Multivariate LSTM-FCNs for time series classification, *Neural Networks*, **116** (2019), 237–245. https://doi.org/10.1016/j.neunet.2019.04.014

AIMS Press