Journal of Ambient Intelligence and Humanized Computing (2022) 13:673–691
https://doi.org/10.1007/s12652-020-02761-x

ORIGINAL RESEARCH

Improving time series forecasting using LSTM and attention models

Hossein Abbasimehr[1] · **Reza Paki**[1]

Received: 26 July 2020 / Accepted: 26 November 2020 / Published online: 3 January 2021

Abstract
Accurate time series forecasting has been recognized as an essential task in many application domains. Real-world time series data often consist of non-linear patterns with complexities that prevent conventional forecasting techniques from accurate predictions. To forecast a given time series accurately, a hybrid model based on two deep learning methods, i.e., long short-term memory (LSTM) and multi-head attention is proposed in this study. The proposed method leverages the two learned representations from these techniques. The performance of this method is also compared with some standard time series forecasting techniques as well as some hybrid cases proposed in the related literature using 16 datasets. Moreover, the individual models based on LSTM and multi-head attention are implemented to perform a comprehensive evaluation. The results of experiments in this study indicate that the proposed model outperforms all benchmarking methods in most datasets in terms of symmetric mean absolute percentage error (SMAPE). It yields the best average rank (AR) among the utilized methods. Besides, the results reveal that model based on multi-head attention is the second-best method with regard to AR, which demonstrates the predictive power of attention mechanism in time series forecasting.

Keywords Time series forecasting · LSTM · Multi-head attention · Hybrid model

1 Introduction

Time series forecasting is an important research field, successfully exploited in many application domains such as in-demand prediction (Abbasimehr et al. 2020; Murray et al. 2018; Shankar et al. 2019), automated teller machine (ATM) cash demand forecasting in banking (Martínez et al. 2018), stock trend prediction in financial markets (Fischer and Krauss 2018; Nayak et al. 2019; Takahashi et al. 2019), electric power load forecasting (Bedi and Toshniwal 2019; Gao et al. 2019; Ghadimi et al. 2018b, 2019), trading area forecast (Kim and Moon 2019), wind prediction (Mir et al. 2020), prediction of natural phenomena (Samet et al. 2019), and so on.

Various time series forecasting techniques have been thus far developed in previous studies that can be categorized into statistical methods, computational intelligence, and a combination of both (Khashei and Bijari 2011). Within the statistical techniques, autoregressive integrated moving average (ARIMA) is a popular time series forecasting method for modeling linear time series (Kumaresan and Ganeshkumar 2020; Murray et al. 2018). However, in practice time series exhibit non-linear characteristics that make non-linear modeling imperative (Panigrahi and Behera 2017). Computational intelligence techniques e.g. feedforward neural networks (NNs) can further model the nonlinear patterns.

In Ghadimi et al. (2018a), an ensemble model comprised of an artificial neural network (ANN), a radial basis function NN (RBFNN), and a support vector machine (SVM) with ordered weighted averaging (OWA) as data fusion, was thus proposed to forecast electricity demand and price. As well, Gao et al. (2019) developed a multi-block forecast engine based on the Elman NN (ENN) to forecast price and load in electricity market, in a way that the proposed method outperformed the conventional ones. Moreover Ghadimi et al. (2018b), proposed a two-stage forecasting engine combining ridgelets and ENNs for electricity load forecasting.

In this sense, several hybrid techniques have been so far developed in the literature, taking advantage of linear and non-linear models in time series forecasting [e.g. Khandelwal et al. 2015; Khashei and Bijari 2011; Panigrahi and Behera 2017; Zhang 2003)]. Such hybrid models have been constructed by combining traditional statistical methods

✉ Hossein Abbasimehr
abbasimehr@azaruniv.ac.ir

[1] Azarbaijan Shahid Madani University, Tabriz, Iran

such as ARIMA and computational intelligence ones like ANN.

Although traditional computational intelligence techniques such as feedforward NNs can model complex patterns among samples, they fail to capture long-term dependency existing in time series. Therefore, recurrent NNs (RNNs), a special kind of ANNs, have been introduced as a suitable alternative for time series forecasting (Chen et al. 2018). Despite being capable of retaining sequential information, RNNs suffer from the vanishing gradient problem, which makes them hard to train (Bengio et al. 1994; Parmezan et al. 2019). As a result, a long short term memory (LSTM) network, an extension of RNNs, has been developed (Farzad et al. 2019; Hochreiter and Schmidhuber 1997) to address problems in RNNs and to apply them successfully for sequence data processing e.g. natural language processing (NLP) and speech recognition (Fischer and Krauss 2018). Recently, RNNs are increasingly being used in the time series forecasting areas, as they are inherently appropriate for sequence modeling tasks. The effectiveness of RNNs, and in particular, the LSTM network has been thus far proved in some recent forecasting studies (Abbasimehr et al. 2020; Fischer and Krauss 2018; Gundu and Simon 2020; Kulshrestha et al. 2020; Law et al. 2019). Utilizing a particle swarm optimization (PSO), an LSTM model for electricity price forecasting was proposed in Gundu and Simon (2020). As well, Abbasimehr et al. (2020) developed an optimized stacked LSTM model for demand forecasting in a furniture company. The proposed method did better than the conventional benchmark models. Similarly Fischer and Krauss (2018), investigated the performance of LSTM networks in financial market forecasting tasks and demonstrated that LSTM outperformed standard forecasting methods. In Law et al. (2019), a deep learning framework was also proposed and applied on tourism demand forecasting. Correspondingly, Kulshrestha et al. (2020) addressed a combined model based on the bidirectional LSTM and Bayesian optimization (BO) for tourism demand forecasting. The proposed model performed better than popular methods such as support vector regression (SVR), RBFNN, and autoregressive distributed lag model (ADLM).

In application domains such as NLP (Sangeetha and Prabha 2020) and speech recognition (Chorowski et al. 2015), extending RNN and LSTM with attention mechanisms (Bahdanau et al. 2014) has accordingly improved the performance. In this sense Fu et al. (2018), showed that attention mechanisms could enhance the performance of LSTM in capturing sentiment information. In Sangeetha and Prabha (2020), a fusion model, based on LSTM and multi-head attention layers, was further proposed for sentiment analysis tasks whose results indicated the better performance of the proposed model compared with the individual ones.

Considering the remarkable improvements obtained using in the mentioned application domains and the importance of accurate time series forecasting in this regard, this study aims at examining the performance of the attention mechanism in time series forecasting. Therefore, we exploit the attention mechanism (Vaswani et al. 2017) and propose a hybrid time series forecasting method based on LSTM and multi-head attention to enhance the accuracy of the time series prediction. The objectives of this study are as follows:

1. Exploring the performance of the attention mechanism in time series forecasting using an extensive set of experiments (16 public time series data are selected from multiple contexts and the performance of the methods is studied)
2. Proposing a hybrid model contributing to accurate time series forecasting by exploiting the capabilities of attention mechanism in focusing on important information and the LSTM layer in capturing short-term and long-term dependencies.
3. Conducting a comparative study using some state-of-the-art individual and hybrid methods proposed in the related literature to demonstrate the power of the proposed hybrid model.

The experiments in this study show that the proposed hybrid model has good ability for time series forecasting. It also reaches the best average rank (AR) among all models. The procedure of the proposed method is as follows:

After preprocessing the input data, the multi-head attention mechanism learns a representation of such data. As well, another representation of the input data is achieved using the LSTM layer by considering the dependency among the data. The final representation is then produced by concatenating the obtained representations through attention mechanism and LSTM, as well as a copy of the original data. Upon reshaping through a flatten layer, the final representation is fed into a fully connected layer wherein the prediction is produced.

The predictive power of the proposed hybrid model is compared with some popular ones such as exponential smoothing (ETS), ARIMA, multilayer perceptron (MLP), as well as some hybrid methods including those proposed in (Babu and Reddy 2014; Panigrahi and Behera 2017; Zhang 2003). As well, LSTM and multi-head attention are implemented for comparison purposes. The results of experiments on public time series data demonstrate that the proposed hybrid model outperforms other utilized in terms of SMAPE. Moreover, the results of comparisons suggest that the method developed based on multi-head is the second-best model.

The remainder of this paper is organized as follows: In Sect. 2, a literature review on time series studies is

presented. Section 3 describes LSTM and multi-head attention and then portrays the proposed model. In Sect. 4, the empirical study is illustrated and the results are compared. Section 5 concludes the study.

2 Literature review

Time series forecasting has been an active area of research in recent years due to its application in many domains. Table 1 outlines some studies in the context of time series forecasting. It also summarizes contribution, forecasting technique, and data domain. As well, ARIMA and ETS (Hyndman et al. 2008) are popular techniques belonging to the category of statistical methods. ANN is also the widely used method from the computational intelligence category. Moreover, some other machine learning techniques such as K-nearest neighbors (KNN) and SVM have been employed for time series forecasting. As Table 1 reveals, several hybrid models via a combination of statistical and machine-learning techniques have been correspondingly proposed in the related literature. Besides, deep learning techniques such as LSTM, which consider long-term dependencies, have been extensively utilized in recent years. However, reviewing literature indicates that the performance of the methods developed based on the attention mechanism (Sangeetha and Prabha 2020) has not been thus far comprehensively evaluated. Consequently, in this study, a hybrid forecasting method is proposed by adopting LSTM and the attention mechanism and then its effectiveness is assessed using publicly available datasets.

Reviewing the literature indicates that although the multi-head attention mechanism (Vaswani et al. 2017) has led to improvements in other application domains especially in NLP, the performance of methods based on the multi-head attention mechanism (Sangeetha and Prabha 2020) has not been comprehensively evaluated. In fact, to the best of our knowledge, none of the previous studies in time series forecasting considered the multi-head attention-based forecasting models and their combinations with other deep learning models such as LSTM for time series forecasting. Our research contribution is exploiting the ability of multi-head attention mechanism in learning important features from time series data and the effectiveness of the LSTM in learning representation considering short-term and long-term dependencies. Attention mechanism learns a representation for each time point in a time series by determining how much focus to place on other time points (Vaswani et al. 2017). Therefore, produces a good representation of time series of input time series and leads to improved time series forecasting. In this study comprehensive performance evaluation of various standard statistical techniques, hybrid methods, and deep learning methods are provided using 16 publicly available datasets.

3 Proposed model

Prior to description of the proposed method, we firstly describe the LSTM and multi-head attention methods. Table 2 gives a list of the nomenclature defining all symbols and parameters used in this study.

3.1 LSTM

The LSTM network developed by (Hochreiter and Schmidhuber 1997) is an extension of RNNs, redesigned to tackle vanishing and exploding problems in RNNs (Chollet 2015; Olah 2015). Each LSTM block is also comprised of a memory cell along with three gates including an input gate $i(t)$, the forget gate $f(t)$ and an output gate $o(t)$ which regulate the flow of information to its cell state c (t). In this sense, the $c(t)$: The architecture of an LSTM block is depicted in Fig. 1. Each of the three gates also performs a different operation (Graves 2013):

- The forget gate $f(t)$ determines which information is discarded.
- The input gate $i(t)$ decides which information is input to the cell state.
- The output gate $o(t)$ regulates outgoing information of the LSTM cell

Considering the input vector $x(t)$ at timestep t and the notations given below, the modeling process of LSTM are defined by Equations (1–6).

$h(t-1)$ and $h(t)$: These values are corresponding to the output values at time point $t-1$ and t.

$c(t-1)$ and $c(t)$: Cell states at time points $t-1$ and t.

$b = \{b_i, b_f, b_c, b_o\}$ are bias vectors of.

$W = \{W_i, W_f, W_c, W_o\}$ are weight matrixes.

$U = \{U_i, U_f, U_c, U_o\}$ are the recurrent weights

$$a(t) = \sigma(W_i x(t) + U_i h(t-1) + b_i) \tag{1}$$

$$f(t) = \sigma(W_f x(t) + U_f h(t-1) + b_f) \tag{2}$$

$$\tilde{c}(t) = \tanh(W_c x(t) + U_c(h(t-1) + b_c) \tag{3}$$

$$c(t) = f_t \times c(t-1) + i_t \times \tilde{c}(t) \tag{4}$$

$$o(t) = \sigma(W_o x(t) + U_o h(t-1) + b_o) \tag{5}$$

Table 1 A summary of literature on time series forecasting in recent years

Study	Contribution	Forecasting technique	Data set
(Babu and Reddy 2014)	Developing a combined model that firstly explores the nature of time series using a moving-average filter in order to apply the models appropriately	ANN, ARIMA, ARIMA-ANN	Sunspot dataset, electricity market data, and dataset from stock market
(Khandelwal et al. 2015)	Proposing a hybrid method to improve the time series forecasting accuracy by exploiting the advantages discrete wavelet transform (DWT), ARIMA and ANN	ARIMA, ANN	Public time series: Lynx exchange rate Indian mining US temperature
(de Oliveira and Ludermir 2016)	Developing a combined model by adopting support vector regression (SVR) and ARIMA	ARIMA, ETS, SVR	12 public time series datasets, and an electricity price dataset
(Atsalakis 2016)	Carbon price forecasting based on a novel hybrid neuro-fuzzy model	ANN, ANFIS, and PATSOS	Carbon price time series
(Martínez et al. 2019)	Formulating a methodology to explore the impact of different preprocessing and modeling alternatives in employing KNN for time series forecasting	k-nearest neighbors (KNN)	ATM cash forecasting
(Panigrahi and Behera 2017)	Devising a combined method through combining ETS and ANN for time series prediction	ETS, ARIMA, ANN, ETS-ANN	16 public datasets from various domains
(Martínez et al. 2018)	Devising a novel approach to tackle the seasonality of time series using multiple specialized KNN models	KNN	ATM cash forecasting
(Fischer and Krauss 2018)	Adopting LSTM for financial market forecasting	LSTM, random forests, a standard deep net, logistic regression	S&P 500 index constituents
(Parmezan et al. 2019)	Conducting comprehensive experiments using statistical and machine learning methods to propose guidelines about employing forecasting methods	Most of the traditional methods such as ARIMA, Seasonal ARIMA, and ETS. Also, some computational intelligence models including MLP, SVM, KNN, and LSTM	95 artificial and real-life series data from various domains
(Büyükşahin and Ertekin 2019)	Proposing a new hybrid method by combining ARIMA-ANN with the empirical mode decomposition (EMD) technique	ARIMA, ANN, and EMD	4 publicly available datasets from different applications including Sunspot, Lynx, Exchange rate and electricity price data
(Shankar et al. 2019)	Implementing LSTM to improve the performance in container throughput prediction	LSTM, ARIMA, Holt-Winters, ANN, ARIMA + ANN, trigonometric regressors	Container throughput data
(Bandara et al. 2020)	Proposing a framework that firstly identify similar time series using clustering algorithms and then use LSTM to build forecasting model	LSTM	CIF2016, NN5
(Abbasimehr et al. 2020)	Proposing an optimized LSTM model for demand series prediction	ARIMA, ETS, SVM, ANN, KNN, LSTM	Demand series of a furniture company

Table 1 (continued)

Study	Contribution	Forecasting technique	Data set
(Abbasimehr and Shabani 2020)	Developing a combined method using various traditional time series forecasting method to predict customers behavior	ARIMA, KNN, moving average(MA)	Time series of POS transaction data
(Sengar and Liu 2020)	Proposing a hybrid approach by combining the deep neural network (DNN) and chicken swarm optimization (CSO)	DNN-CSO, DNN, ANN, ARIMA, MLP	Wind energy system

$$h(t) = o(t) \times \tanh(c(t)) \tag{6}$$

where functions σ and tanh are sigmoid and hyperbolic tangent activation functions, respectively. The $\times$ indicates the element-wise multiplication of two vectors.

3.2 Multi-head attention

Recently, attention mechanisms have been successfully used in NLP applications (Sangeetha and Prabha 2020). The work of Vaswani et al. (2017) in this respect suggested that the attention mechanism could be effective for sequence data processing. In this study, we utilized the multi-head attention mechanism developed in (Vaswani et al. 2017) (Fig. 2). An attention function takes a query Q and a set of keys and values $< K, V >$ to get the output O. This procedure is often called Scaled Dot-Product attention. Multi-head attention is a set of multiple heads that jointly learn different representations at every position in the sequence (Li et al. 2018).

3.3 Architecture of the proposed method

Figure 3 illustrates the components of the proposed method. The proposed method initially computes the LSTM representation and the multi-head attention representation. Subsequently, it concatenates these representations with the original input data and finally predicts the output value using the fully connected dense layer. The main advantage of LSTM is its ability to capture long-term dependency in a time series. The reason for selecting LSTM representation as one input is that the LSTM consists of mechanisms that take into account the sequential nature of time series. Also, multi-head attention has the ability to capture the most important input features and gives higher weights to them. In time series data the importance of different time points may be different. Therefore, it is essential to focus on key features using the attention mechanism. For both LSTM and multi-head attention the input is a raw time series data represented through input–output format.

In order to concatenate the obtained representations and a copy of original time series data, a flatten layer is utilized. The flatten layer performs operations to reshape data into a format so that they can be concatenated. Therefore, the output of both LSTM and Multi-head attention models, as well as a copy of original time series data are concatenated into one vector suitable for the dense layer. The dense layer generates the output.

4 Empirical study

In this section, the utilized datasets are firstly described. Next, the preprocessing steps required to transform the input into a suitable form that can be modeled by the employed

Table 2 Nomenclature

Nomenclature			
$x(t)$	The value of input series at time step t	tanh	Hyperbolic tangent activation
$i(t)$	Input gate	TS	Time series
$f(t)$	Forget gate	×	Element-wise multiplication of two vectors
$o(t)$	Output gate	Q	A query
$c(t)$	Cell state at time point t	L	Lag size
$h(t)$	Output values at time point t	N_L	Number of units in LSTM
$b = \{b_i, b_f, b_c, b_o\}$	Bias vectors of	N_d	Number of neurons in dense layer
$W = \{W_i, W_f, W_c, W_o\}$	Weight matrixes	LR	Learning rate
σ	Sigmoid activation function		

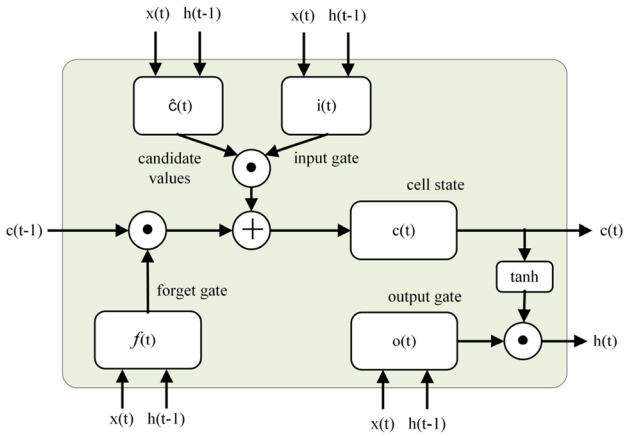

Fig. 1 The architecture of LSTM (Graves 2013; Olah 2015)

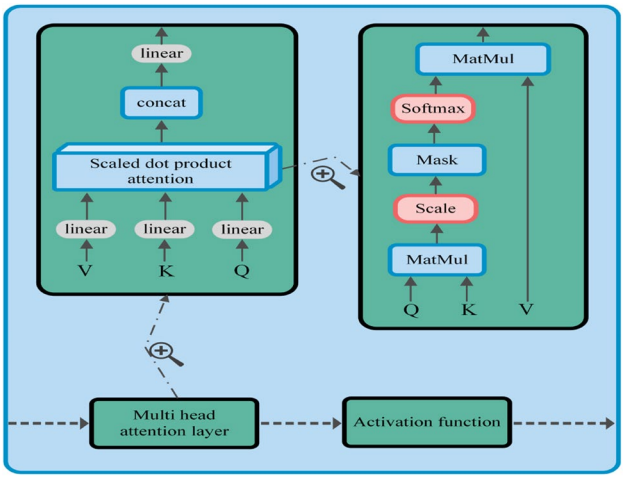

Fig. 2 The architecture of multi-head attention

methods are explained. Then, the experimental setup for each method is described. Finally, the analysis of the results is provided.

4.1 Datasets

Table 3 portrays 16 public datasets to compare the performance of the proposed methods. As well, the size of train, validation, and test parts for each is given in Table 3, directly adopted from (Panigrahi and Behera 2017). All the experiments are conducted using the data division setting displayed in Table 3.

4.2 Instance creation

LSTM as a kind of machine learning techniques needs the input to be instances of input–output format. In this step, using a lag L, subsequences of length $L + 1$ is extracted from the series. The first L points of a sequence are considered as the input and the last point, $L + 1$ is labeled as the target. These subsequences are in fact the instances which are in a suitable format for training LSTM. The construction of instances is depicted in Fig. 4.

4.3 Evaluation measures

We use the SMAPE measure which is suitable for evaluating the performance of a time series forecasting method. The definition of SMAPE is given by Equation (7):

$$SMAPE = \frac{1}{n}\sum_{t=1}^{n}\frac{|\hat{y}_t - y_t|}{\frac{|\hat{y}_t|+|y_t|}{2}} \times 100 \tag{7}$$

where $\hat{y}_t$ and y_t are the predicted and actual value at time point t.

4.4 Experiment setup

ARIMA, ETS, MLP, and the methods proposed in (Babu and Reddy 2014; Panigrahi and Behera 2017; Zhang 2003) are utilized as the comparison techniques. In this study, a single-step ahead forecasting strategy is adopted. LSTM,

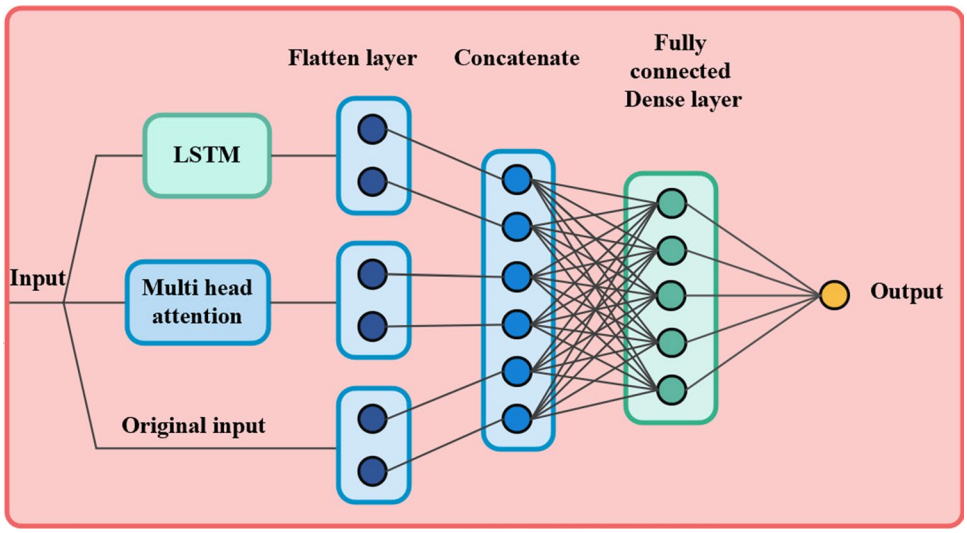

Fig. 3 The architecture of the proposed hybrid method

Table 3 Time series description and divisions (Panigrahi and Behera 2017)

Dataset	Description	Train size	Validation size	Test size
TS 1	Accidental Death	43	14	15
TS 2	IBM	221	74	74
TS 3	Lake	360	120	120
TS 4	Lynx	68	23	23
TS 5	Pollution	78	26	26
TS 6	Stock	59	20	20
TS 7	Sun Spot	172	58	58
TS 8	Colorado River	446	149	149
TS 9	Passenger	86	29	29
TS 10	UK Internet traffic	994	331	332
TS 11	Temperature	144	48	48
TS 12	Unemployment	144	48	48
TS 13	Milk	93	31	32
TS 14	Mumps	320	107	107
TS 15	Chickenpox	298	100	100
TS 16	Traffic	108	36	36

Multi-Head ATT, and the proposed method (namely, Multi-Head ATT-LSTM) are accordingly implemented using Keras library in Python (Chollet 2015).

The significant hyperparameters that must be set for the employed methods including the proposed hybrid model, Multi-Head ATT, and LSTM are as follows:

1. Lag size (L): Finding the optimal value for the lag is a significant task as the performance of time series forecasting is mainly dependent on the lag value.
2. Number of units in the LSTM layer (N_L): Selecting the optimal number of neurons for the LSTM layer is a crucial task. A small number of neurons can thus prevent LSTM from memorizing the required information, so the generated representation fails to be informative. As well, selecting a very high number of neurons may lead to overfitting (Reimers and Gurevych 2017).
3. Number of units in the dense layer (N_d): The final representation of the input data is fed into the dense layer to produce predictions. Selecting the optimal number of neurons in this layer can thus have a significant impact on forecasting.

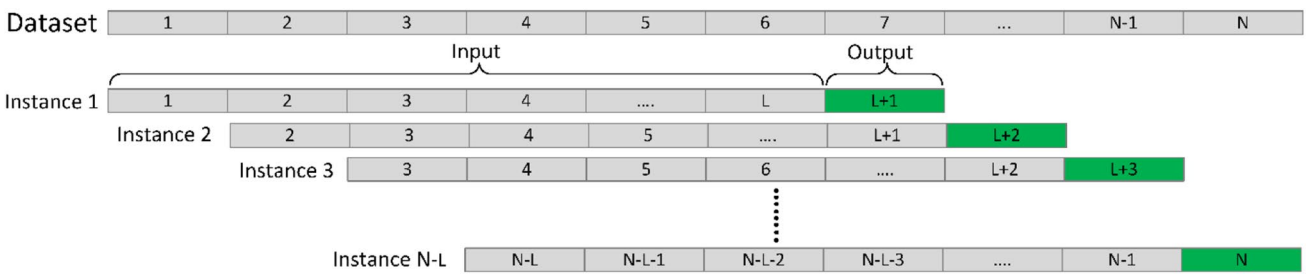

Fig. 4 Creation of instances for LSTM network

4. Learning rate (*LR*): Learning rate is an important hyperparameter that greatly affects the model performance. It adjusts the extent of changes to the model weights (Reimers and Gurevych 2017).
5. Epoch size (*epoch*): One training epoch is a complete pass through the entire training data. A smaller number of epochs can accordingly make the model incapable of capturing the patterns of the training data. In addition, a larger epoch size may cause the model be overfitted to the training data.

As mentioned before, adjusting epoch size is an important step to prevent a model from overfitting. In each experiment, the epoch size is accordingly specified using the early stopping (Prechelt 2012). The procedure exercised for early stopping is as follows:

A. The model is trained using the training data. At the end of each epoch, the performance of the model on the validation set is computed.
B. Based on each epoch in which the performance on the validation set increases, the trained model corresponding to that epoch is saved.
C. Training is stopped when the performance on the validation dataset decreases over a given number of epochs or once no change in performance is observed.
D. As the training terminates, the best model is returned.

In this study, for all methods, the Adam algorithm (Kingma and Ba 2014) was used as optimizer.

5 Results and analysis

Our experiments are divided into two parts. For Part 1, except for lag, a fixed value is set for all hyperparameters. Also, for Part 2, the BO algorithm (Brochu et al. 2010) is employed to find the optimal hyperparameter values. In the following subsections, the results of experiment Part 1 and Part 2 are provided.

5.1 Experiments—part 1

The important hyperparameters of the utilized methods were explained in the previous section. For each method presented in this study, the optimal lag size corresponding to each time series is obtained using the grid search. The best lag size is thus selected considering the validation set error of the corresponding model. Table 4 illustrates the optimal lag size acquired using the grid search for each time series.

For LSTM, the number of hidden units is set to128. For Multi-Head ATT, the number of heads is also set to lag size. The number of units in the dense layer for all methods is additionally set to 100. Likewise, the learning rate for all methods is set to 0.001. The epoch size is determined following the early stopping procedure that is described in the previous section. To employ the early stopping, the epoch limit is set to 500.

Table 4 Optimal lag size for each considering modeling techniques

Time series	Multi-Head ATT-LSTM/ Multi-Head ATT	LSTM
TS 1	15	18
TS 2	6	5
TS 3	17	10
TS 4	9	9
TS 5	15	19
TS 6	5	10
TS 7	9	9
TS 8	10	6
TS 9	14	15
TS 10	16	16
TS 11	13	13
TS 12	16	13
TS 13	15	15
TS 14	10	13
TS 15	10	19
TS 16	14	17

Table 5 illustrates the results of the methods in terms of SMAPE. For each time series, the best SMAPE is indicated in bold. As demonstrated, the proposed model obtains the lowest SMAPE in seven time series. Besides, the proposed method achieves better SMAPE than ETS in 12 cases, ARIMA model in 12 time series, MLP in 15 cases, Babu and Reddy's method (Babu and Reddy 2014) in 13 cases, Zhang's method (Zhang 2003) in 11 time series and ETS-ANN (Panigrahi and Behera 2017) in 9 cases. Besides, it performs better than Multi-Head ATT and LSTM in 11 and 13 time series respectively.

According to the results of the utilized models (Table 5), the values are ranked from 1 (the lowest SMAPE) to K = 9 (the highest SMAPE) for each time series. Note that K is the number of comparison methods. The results of ranking are also illustrated in Table 6. In addition, the AR of each method is computed and given in Table 6. As indicated, the proposed method achieves the best AR (AR = 3). The proposed method also benefits from using two representations that extract the informative features from time series. The LSTM model additionally learns representations by considering long-term dependency. In addition, Multi-Head ATT is able to give higher weights to the most important input features.

In addition, the second-best AR is obtained using Multi-Head ATT, which demonstrates the usefulness of attention

Table 5 Comparison of methods in terms of SMAPE

Time series	ETS(Panigrahi and Behera 2017)	ARIMA(Panigrahi and Behera 2017)	MLP(Panigrahi and Behera 2017)	Babu and Reddy (2014)	Zhang (2003)	ETS-ANN(Panigrahi and Behera 2017)	Multi-Head ATT	LSTM	Proposed Method
Accidental death	7.0366	5.4168	4.6924	5.4511	3.6836	3.4941	**3.2258**	5.181	3.5718
IBM	1.5427	1.6486	1.9525	2.489	1.6592	1.5955	**1.5369**	1.9411	1.5545
Lake	2.8143	1.7767	1.8321	**1.7132**	1.7185	1.7607	1.8264	2.7376	1.8042
Lynx	**48.781**	50.714	63.445	51.249	53.269	56.674	55.1854	51.4051	52.8623
Pollution	21.329	19.295	25.445	18.318	17.68	18.537	16.5226	18.3555	**16.1661**
Stock	**9.9226**	11.811	41.747	37.84	12.552	10.609	15.2748	21.2092	18.1205
Sun spot	49.039	37.141	35.005	37.141	35.537	**29.453**	35.1682	29.6602	33.0126
Colorado river	**28.822**	37.023	39.228	43.85	36.758	36.059	33.6573	42.3836	42.3374
Passenger	9.9966	6.8829	5.3276	4.7027	5.0594	3.5187	3.2897	3.8609	**3.0606**
UK Internet traffic	4.9155	4.8438	2.8952	2.943	3.122	3.0218	2.6158	6.4831	**2.5026**
Temperature	8.3699	4.8149	4.1199	4.2225	4.4404	4.4404	3.9857	4.4203	**3.9705**
Unemployment	6.5909	7.3117	6.0006	6.2131	6.0949	4.4097	3.8629	5.2619	**3.8557**
Milk	2.0752	1.9558	1.3403	1.0848	0.7258	**0.6175**	1.3751	3.4715	1.2552
Mumps	29.112	31.333	23.912	21.714	24.494	22.449	20.6951	19.5662	**19.2959**
Chickenpox	54.297	54.43	34.568	38.393	37.818	28.391	23.8511	22.7060	**22.2735**
Traffic	18.138	17.669	13.036	11.63	11.406	11.053	11.2386	**10.9980**	11.4085

(The lowest SMAPE for each time series is indicated in bold)

Table 6 The rank of methods

Time series	ETS(Panigrahi and Behera 2017)	ARIMA(Panigrahi and Behera 2017)	MLP(Panigrahi and Behera 2017)	Babu and Reddy (2014)	Zhang (2003)	ETS-ANN(Panigrahi and Behera 2017)	Multi-Head ATT	LSTM	Proposed method
TS 1	9	7	5	8	4	2	**1**	6	3
TS 2	2	5	8	9	6	4	**1**	7	3
TS 3	9	4	7	**1**	2	3	6	8	5
TS 4	**1**	2	9	3	6	8	7	4	5
TS 5	8	7	9	4	3	6	2	5	**1**
TS 6	**1**	3	9	8	4	2	5	7	6
TS 7	9	7	4	8	6	**1**	5	2	3
TS 8	**1**	5	6	9	4	3	2	8	7
TS 9	9	8	7	5	6	3	2	4	**1**
TS 10	8	7	3	4	6	5	2	9	**1**
TS 11	9	8	3	4	6	7	2	5	**1**
TS 12	8	9	5	7	6	3	2	4	**1**
TS 13	8	7	5	3	2	**1**	6	9	4
TS 14	8	9	6	4	7	5	3	2	**1**
TS 15	8	9	5	7	6	4	3	2	**1**
TS 16	9	8	7	6	4	2	3	**1**	5
AR	6.6875	6.5625	6.125	5.625	4.875	3.6875	3.25	5.1875	**3**

Bold values indicate the best ranks

mechanisms in capturing patterns existing in time series. The hybrid forecasting technique developed in (Panigrahi and Behera 2017) obtains the third-best AR. The worst performing method in terms of AR is the ETS, which achieves the best SMAPE value for TS 4, TS 6, and TS 8, and its overall rank is the worst among all methods. Furthermore, ARIMA is another statistical method that performs worse among the employed cases. This may be due to the fact that the statistical methods are suitable modeling linear time series Also, the performance of the statistical methods is sensitive to parameter selection. Comparing the results of the LSTM model indicates that it achieves a better AR than the standard forecasting methods such as ETS, ARIMA, and MLP. This demonstrates the suitability of LSTM for sequence data processing.

5.2 Experiments—part 2

In the previous section, fixed values were set for hyperparameters except for lag hyperparameter which was obtained using the grid search. To gain a better understanding of performances associated with the models, the whole hyperparameter space needs to be investigated in order to find the best hyperparameter configuration in which a model has the best forecasting performance. Finding the best hyperparameter configuration is thus called fine-tuning (Law and Shawe-Taylor 2017). The common method for hyperparameter fine-tuning is the grid search in which all possible hyperparameter configurations must be explored. As a grid search involves more computational complexity, metaheuristic-based methods such as the BO algorithm (Brochu et al. 2010) are often employed for selecting optimal hyperparameters.

In this section, further experiments are conducted using hyperparameters selected by the BO algorithm. The proposed method along with Multi-Head ATT and LSTM possess several hyperparameters affecting the forecasting performance.

To perform BO, the domain of each hyperparameter must be firstly specified. Then, BO finds the best values. Accordingly, BO is conducted for the proposed method as well as Multi-Head ATT and LSTM. The range of hyperparameters for all models is given in Table 7. Please note that the epoch size is determined following the procedure explained in the previous subsection.

After selecting the best hyperparameters, the proposed method along with the Multi-Head ATT, and LSTM are applied to the training data. We perform 10 experiments for each time series data described in Table 3, and compute the average SMAPE across 10 experiments. The results are illustrated in Table 8.

Considering the results of ETS, ARIMA, MLP, ETS-ANN (Panigrahi and Behera 2017), the hybrid method developed by Babu and Reddy (2014), and Zhang's method (Zhang 2003) provided in Table 5, and the results of experiments shown in Table 8, the proposed method outperforms ETS in 13 cases, ARIMA model in 15 time series, MLP in 15 cases, Babu and Reddy's method Babu and Reddy (2014) in 14 cases, Zhang's method (Zhang 2003) in 11 time series and ETS-ANN (Panigrahi and Behera 2017) in 11 cases. In addition, it performs better than Multi-Head ATT and LSTM in 11–15 time series, respectively. Besides, the proposed model achieves the best AR among all methods (as seen in Table 9). As well, the Multi-Head ATT obtains the second-best results in terms of AR. ETS and ARIMA, as the standard statistical methods, also gain the worst AR. The results of the experiment Part 2 are in line with the ones in Part 1, validating the effectiveness of attention-based methods in time series forecasting.

Table 7 Hyperparameter range

Hyperparameter	(Initial value: step value: final value)
Lag size	$L = (5: 1: 20)$
Number of units in the LSTM layer	$N_L = (16: 16: 128)$
Number of units in dense layer	$N_d = (20: 20: 100)$
Learning rate	$LR = [0.0001, 0.001, 0.01]$

Table 8 Comparison of methods in terms of SMAPE- the lowest SMAPE for each time series is indicated in bold (The parameters of Multi-Head ATT, LSTM, and the Proposed method are selected using BO)

Time series	Multi-Head ATT	LSTM	Proposed Method
TS 1	**3.40**	5.6664	3.5407
TS 2	**1.495**	2.0394	1.5803
TS 3	1.9199	2.06	1.7581
TS 4	48.6108	46.6450	**43.8696**
TS 5	**17.5385**	17.707	17.9335
TS 6	30.5964	49.4288	25.0625
TS 7	33.5282	32.1709	32.1039
TS 8	32.6737	44.8583	36.8639
TS 9	3.2977	4.30	**3.2731**
TS 10	2.7921	6.1614	**2.7655**
TS 11	3.9583	4.1907	**3.9071**
TS 12	**4.034**	4.8431	4.0818
TS 13	1.6886	2.4082	1.3902
TS 14	19.5385	17.8756	**17.3091**
TS 15	24.2341	23.9568	**22.44**
TS 16	11.5548	10.6973	**10.5722**

5.3 Performing statistical test

To assess the results of the utilized methods, the non-parametric Friedman test with Hochberg's method (Demšar

Table 9 The mean rank

Method	*AR*
Proposed method	2.375
Multi-Head ATT	3.0625
LSTM	5.3125
Zhang method (Panigrahi and Behera 2017)	4.84375
ETS-ANN(Panigrahi and Behera 2017)	3.71875
Babu(Panigrahi and Behera 2017)	5.65625
MLP(Panigrahi and Behera 2017)	5.9375
ARIMA(Panigrahi and Behera 2017)	6.78125
ETS(Panigrahi and Behera 2017)	6.875

Table 10 Statistical test on the results of the utilized methods. The proposed method performs best, Multi-Head ATT and ETS-ANN do not perform significantly worse

Method	p value of Hochberg
Proposed method	–
Multi-Head ATT	0.796
LSTM	0.024
Zhang method (Panigrahi and Behera 2017)	0.049
ETS-ANN(Panigrahi and Behera 2017)	0.498
Babu(Panigrahi and Behera 2017)	7×10^{-3}
MLP(Panigrahi and Behera 2017)	1.25×10^{-3}
ARIMA(Panigrahi and Behera 2017)	2.06×10^{-4}
ETS(Panigrahi and Behera 2017)	1.45×10^{-4}

2006) is performed on the results of the experiment Part 1. The statistical test contains two steps. At first, whether there are statistically significant differences among methods is investigated using the Friedman test. Afterwards, the Hochberg's test is utilized to compare the proposed model with other benchmark methods. The statistical test is accordingly performed using the AR presented in Table 6. Also, the significance level is set to $\alpha = 0.05$.

The p value corresponding to the computed Friedman statistic is 4.1×10^{-5}, which indicates that the differences among all methods (a total number of nine methods utilized in this study) are significant.

Table 10 illustrates the results of Hochberg's method. As the proposed method achieves the best results, we use it as the control method. According to Table 10, six comparison methods including ETS, ARIMA, MLP, Babu, and Zhang methods (Panigrahi and Behera 2017) and LSTM perform significantly worse than the proposed method. Besides, methods including ETS-ANN (Panigrahi and Behera 2017) and Multi-Head ATT do not perform significantly worse.

5.4 Visual representation of the actual and predicted values

As mentioned in the previous section, the proposed method achieved the top rank among all other methods. To further illustrate the predictive power of the proposed method and Multi-Head ATT on the utilized time series, in Figs. 5, 20 we also visualize the actual and predicted values for each time series with the results of these models. The black line represents the actual data while the dashed green and red lines represent the predicted values of the proposed method and Multi-Head ATT, respectively. In the following, we provide more analysis of the performances of the proposed method and the Multi-Head ATT.

For TS 1, as Fig. 5 shows, for both methods, the predicted values are close to the actual values. However, the Multi-Head ATT forecast accurately the peak values compared to the proposed method. For TS 2, as Fig. 6 shows, the values predicted by both methods are close to the actual values.

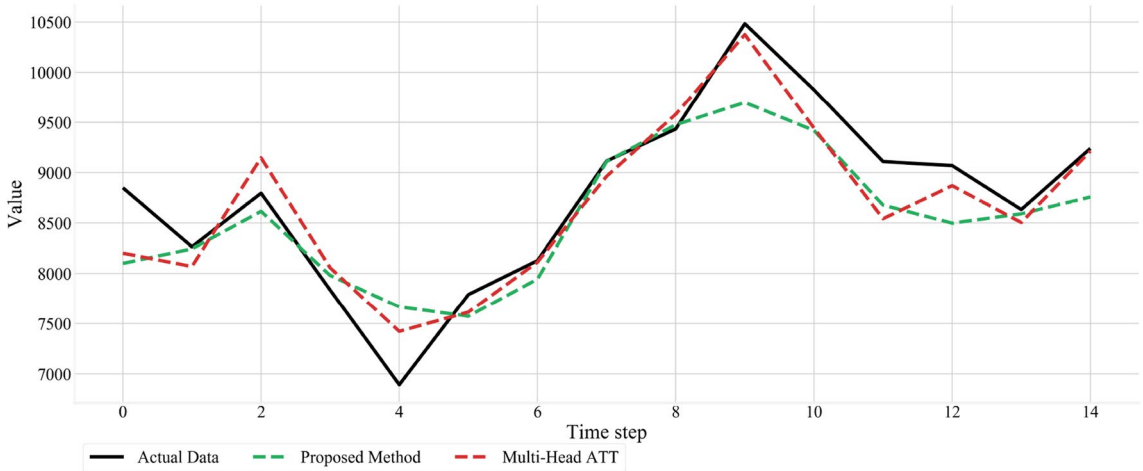

Fig. 5 TS 1- Accidental death time series

Although at some point the Multi-Head ATT accurate forecasts, there is not any significant difference between them at all. Also, considering the Fig. 7, which plots TS 3, the predicted values of both methods are completely overlapping the actual values.

The plot for TS 4, which is illustrated in Fig. 8, indicates that except for the first 5 points, in which the predicted values are not accurate, at the remaining points the difference between the predicted values of the proposed method and the actual values is lower than the Multi-Head ATT. Also, the plot of TS 5 (as illustrated in Fig. 9) indicates that although there are overlaps between the predicted values and actual values in some points, both methods do not exhibit good performance. For TS 6, as illustrated in Fig. 10, Multi-Head ATT performs better than the proposed method. The reason behind this performance is that this time series contains a sudden change immediately at the point the test set begins. As the proposed method incorporated the LSTM model, it considers the dependency between data, so the generality of the model is reduced when sudden changes appear. Besides, both models use the validation error in tuning their weights and assume that the error on the test set will be similar to the validation set one.

Figure 11, which plots TS 7, indicates that although the proposed method predicts accurate values compared to the Multi-Head attention at the majority of time points, the difference is slight. In addition, for TS 8, as shown in Fig. 12, the Multi-Head ATT performs accurate forecasting compared to the proposed method. For TS 9, TS 10, TS 11, TS 12, TS 13, as shown in Figs. 13, 14, 15, 16 and 17, respectively, both methods exhibit accurate forecasting, however, the overlaps between the forecasted values by the proposed method with the actual values are slightly greater than that of Multi-Head ATT.

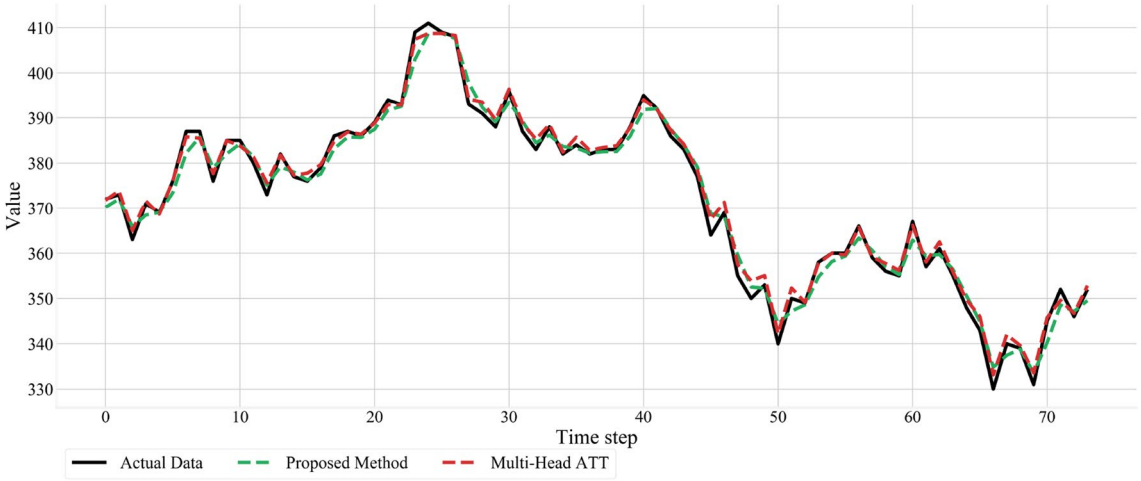

Fig. 6 TS 2- IBM time series

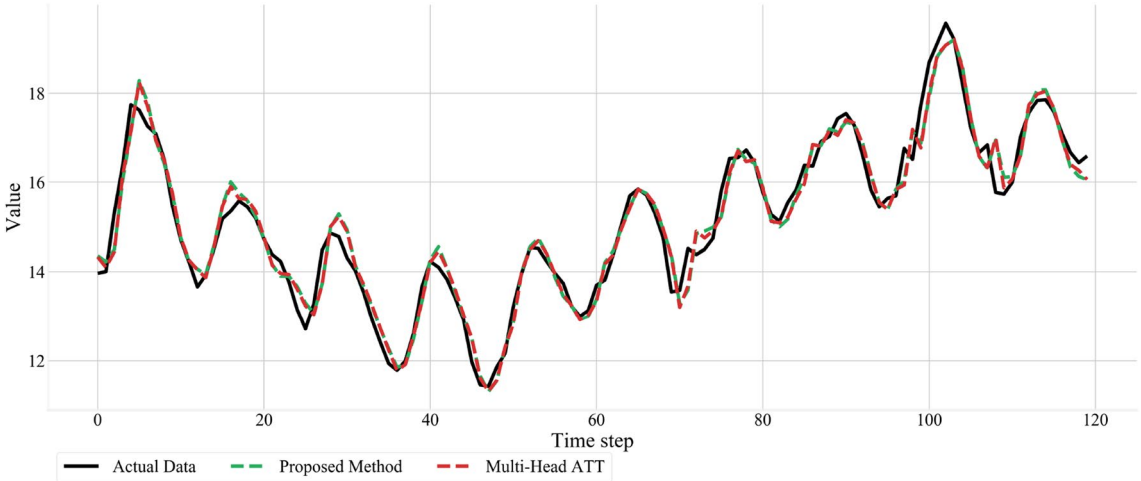

Fig. 7 TS 3—Lake time series

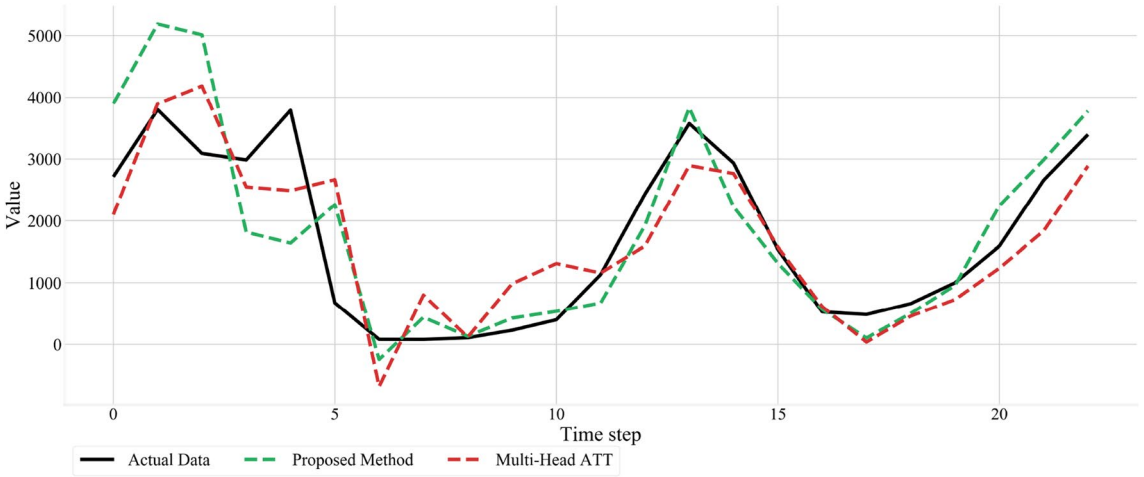

Fig. 8 TS 4- Lynx time series

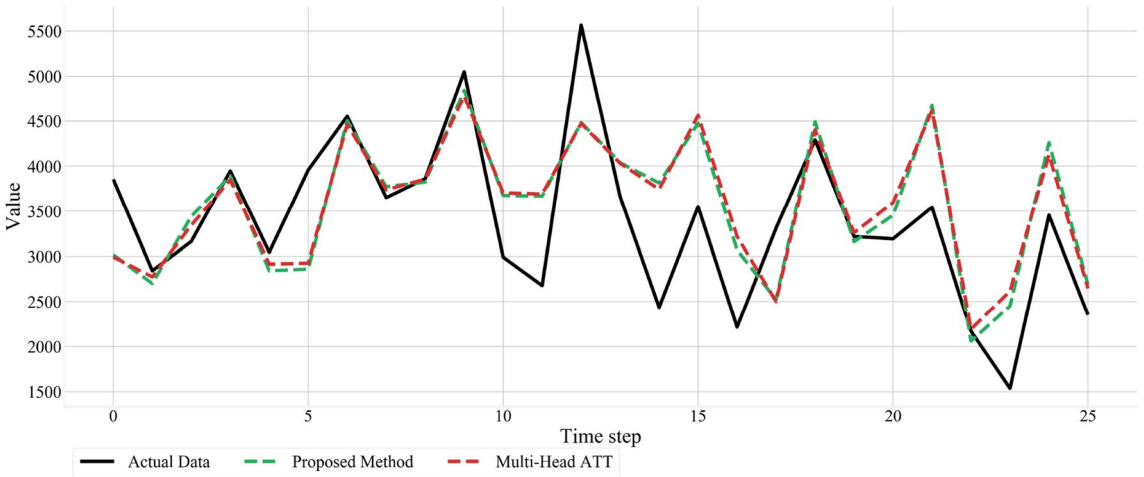

Fig. 9 TS 5—Pollution time series

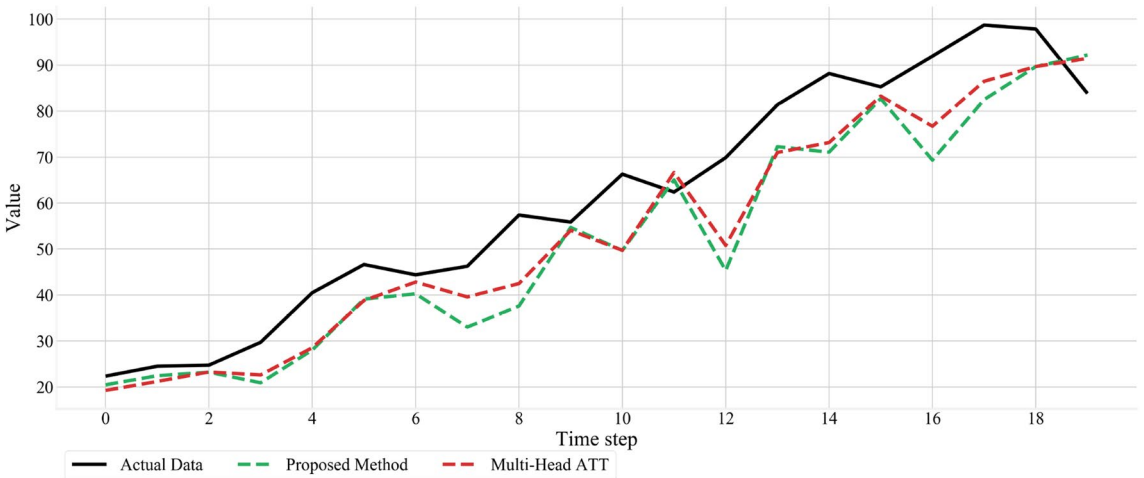

Fig. 10 TS 6—Stock time series

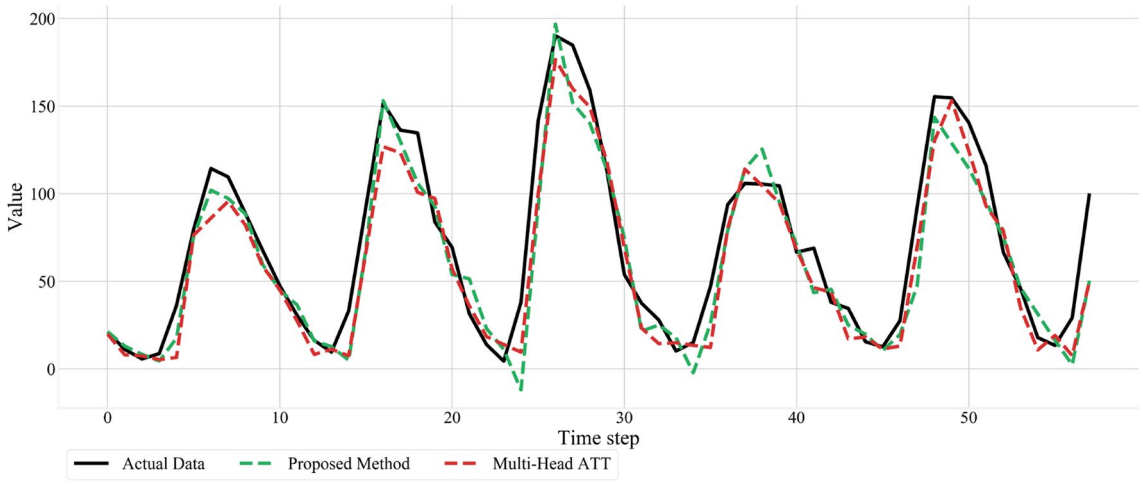

Fig. 11 TS 7—Sun Spot time series

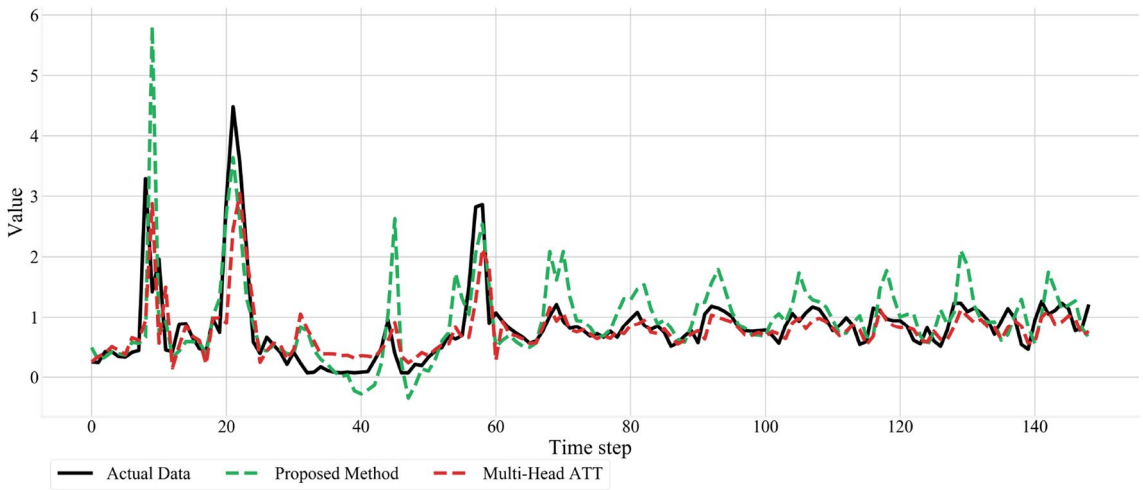

Fig. 12 TS 8—Colorado river time series

Figure 18, which illustrates TS 14, indicates that the overall performance of the proposed method is better than the Multi-Head ATT for TS 14. Besides, the forecasts are very close to the actual data in the majority of time points. Also, for TS 15, as shown in Fig. 19, except for the time points with peak values, in which the predicted values by the proposed method are more accurate than that of Multi-Head ATT, in other time points, the forecasting is similar. Finally, for TS 16, which is plotted in Fig. 20, both methods exhibit similar performance, and the difference is not significant.

6 Discussion

In this study, we proposed a hybrid method based on LSTM and attention mechanism. The results on 16 time series indicate the predictive power of the proposed method. It reaches the best performance. The results demonstrated the predictive power of the attention mechanism in time series forecasting. Also, the two sets of experiments, Part 1 and Part 2, indicated that the proposed method of attention mechanism

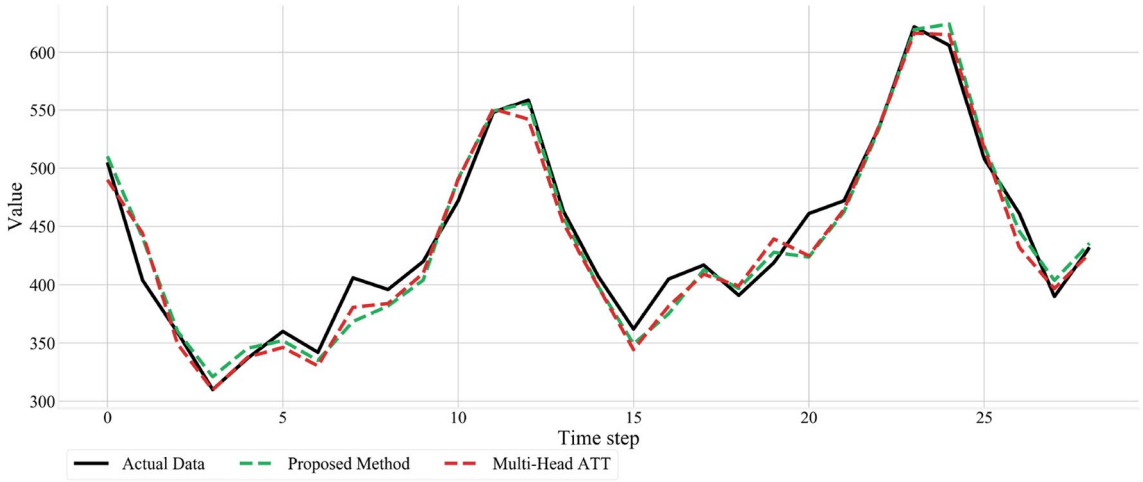

Fig. 13 TS 9—Passenger time series

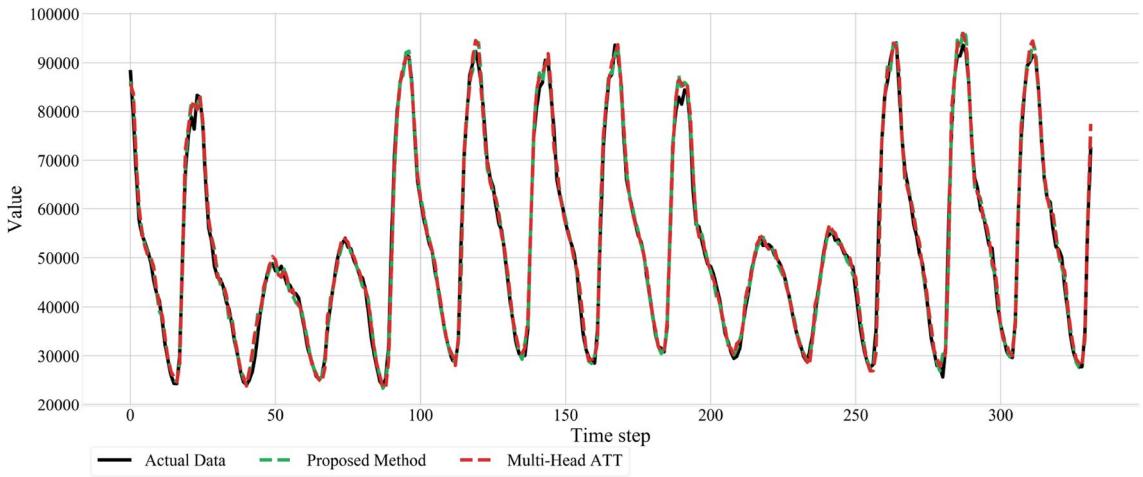

Fig. 14 TS 10—Internet traffic data of UK time series

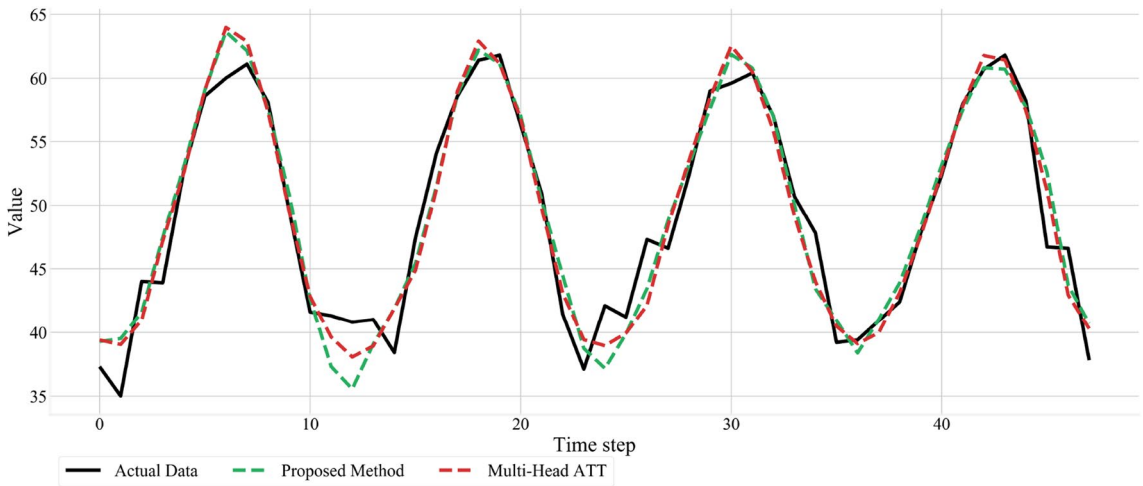

Fig. 15 TS 11—Temperature time series

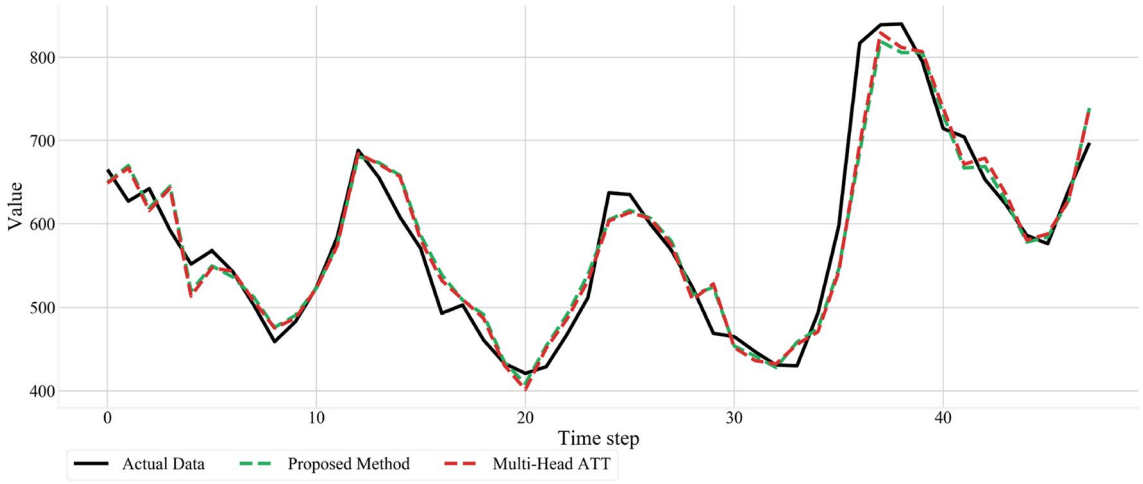

Fig. 16 TS 12—Unemployment time series

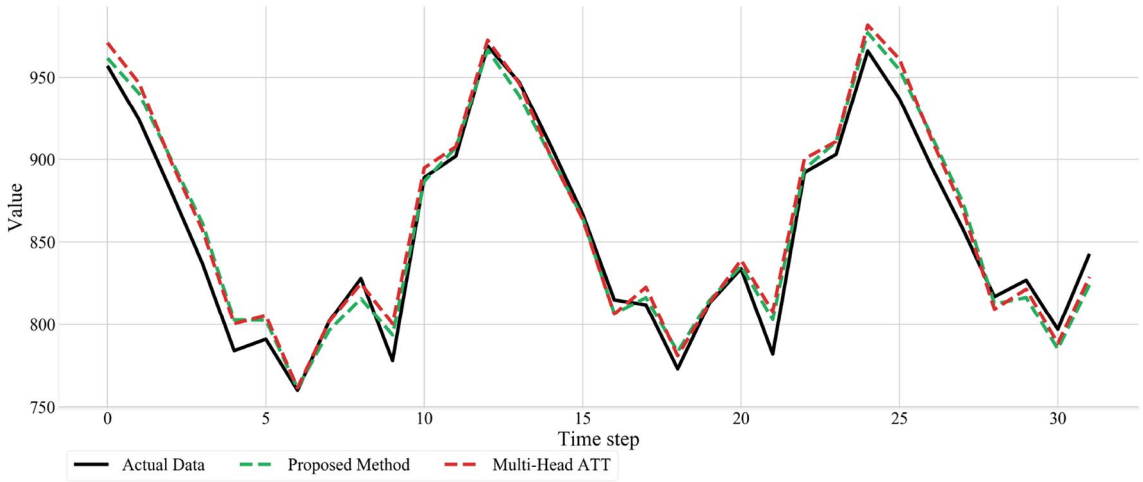

Fig. 17 TS 13—Milk time series

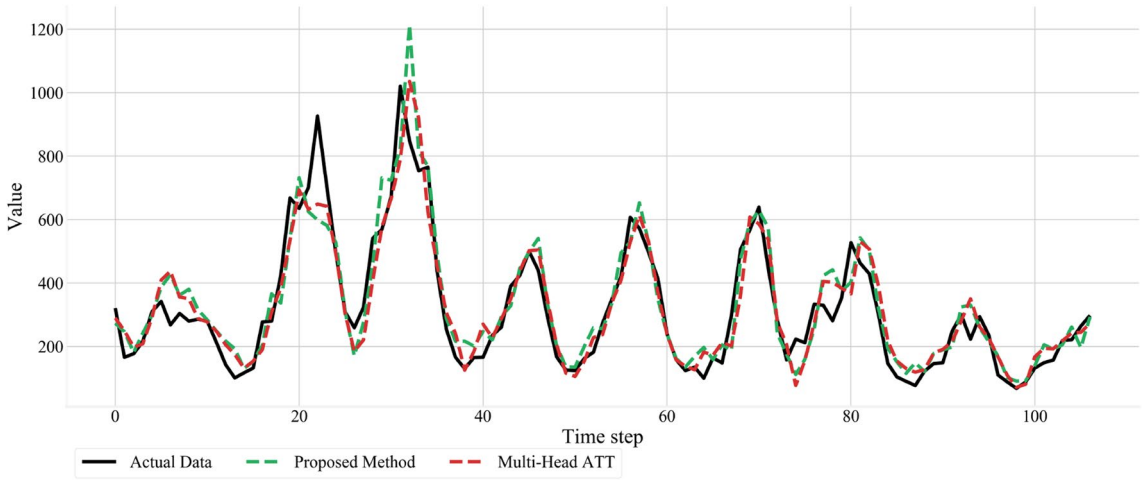

Fig. 18 TS 14—Mumps time series

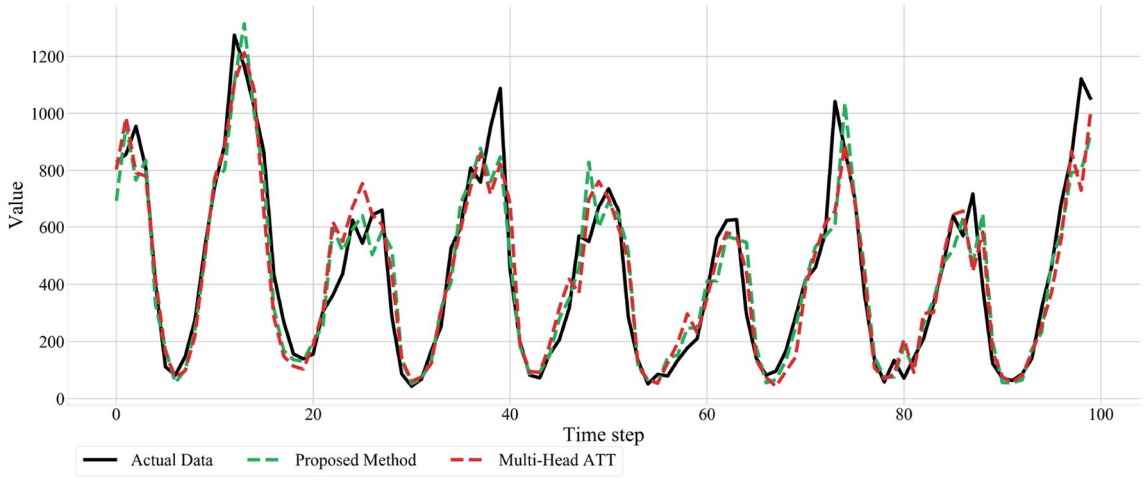

Fig. 19 TS 15—Chickenpox time series

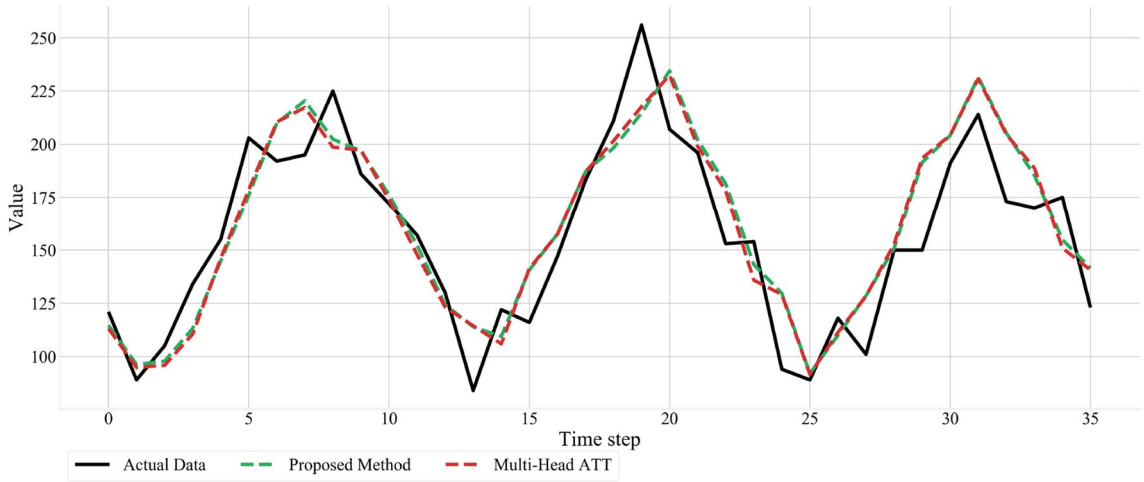

Fig. 20 TS 16—Traffic time series

and LSTM outperformed state-of-the-art methods in time series forecasting including ARIMA, ETS, and MLP. Besides, the proposed method archived better results compared to some hybrid methods introduced in literature e.g. (Babu and Reddy 2014; Panigrahi and Behera 2017; Zhang 2003). Our experiments demonstrate that although models based on the attention mechanism improves the time series forecasting accuracy. Their combination with the LSTM network increases the time series forecasting accuracy.

6.1 Assumptions

Similar to any neural network training task, in this study, we use a validation set and monitor the model building process using the error on the validation set. Therefore, our main assumption is that the error on the test set or unseen future data will be similar to the error on the validation set. Also, in this study, we assume that the optimal hyperparameters for the proposed method have been selected. In fact, achieving good results with the proposed method requires the optimization of some hyperparameters described in the experiment section.

6.2 Discussion on parameters

In order to gain a better understanding of the performance of the proposed method, the experiments are conducted in two parts. Part 1 uses a fixed hyperparameter value. Also, in Part 2, the BO algorithm is employed for finding optimal hyperparameters. In both Part 1 and Part 2, the proposed method achieves the best performance. Also, the results indicate that among many hyperparameters, the lag size and number of neurons have a crucial impact major on the performance of models. The results are in line with the previous studies

such as (Martínez et al. 2018; Panigrahi and Behera 2017) in which indicated that Lag size has a significant impact on forecasting accuracy. Also, the importance of adjusting the number of neurons for deep learning methods has been emphasized in past researches (Abbasimehr et al. 2020; Sagheer and Kotb 2019).

6.3 Implications of results

The results indicate that the models based on the attention mechanism outperform individual methods such as LSTM. Multi-head attention learns a representation by focusing on important features in time series. The reason behind the improvements obtained by the proposed method is that this method concatenates two representations learned from the multi-head attention mechanism and the LSTM network. By exploiting the multi-head attention mechanism, the proposed method learns important features from the input time series. Also, by employing the LSTM network, it considers the dependency among data.

6.4 Limitations

Generally, the deep learning methods contain many hyperparameters that should be specified optimally. As a deep learning method, the proposed method also needs hyperparameter tuning. Since testing every combination of hyperparameters may not be possible, so we have to use metaheuristic-based algorithms such as Bayesian optimization. It should be noted that the optimization algorithms may fall into local sub-optimal solutions.

7 Conclusion

Time series forecasting is a challenging task as time series contain both linear and non-linear patterns. Various time series forecasting approaches have been thus far developed in the related literature. In this respect, deep learning methods have the ability to learn useful features from data and enhance the accuracy of time series forecasting. Therefore, in this study, a hybrid model, was developed based on LSTM and the multi-head attention mechanism. The given model benefitted from two representations produced using LSTM and multi-head attention to enhance forecasting accuracy. The presented model also compared with some traditional methods including ETS, ARIMA, MLP, and some hybrid techniques proposed in the literature. For comprehensive compression, LSTM and multi-head attention were also implemented. The performance of all methods was then evaluated using 16 public time series. The results of the experiments indicated that the proposed hybrid model outperformed all other methods utilized in most time series datasets. In addition, the proposed model achieved the best AR among all comparison methods. Besides, the second-best AR was obtained using multi-head attention which demonstrated the usefulness of attention mechanisms in capturing patterns existing in time series. Furthermore, the standard time series forecasting techniques such as ETS, ARIMA, and MLP showed poor performance compared with other utilized methods. For future work, we aim to take into account the time series characteristics in optimizing the model parameters.

References

Abbasimehr H, Sabani M (2020) A new framework for predicting customer behavior in terms of RFM by considering the temporal aspect based on time series techniques. J Ambient Intell Human Comput. https://doi.org/10.1007/s12652-020-02015-wh

Abbasimehr H, Shabani M, Yousefi M (2020) An optimized model using LSTM network for demand forecasting. Comput Ind Eng 143:106435. https://doi.org/10.1016/j.cie.2020.106435

Atsalakis GS (2016) Using computational intelligence to forecast carbon prices. Appl Soft Comput 43:107–116

Babu CN, Reddy BE (2014) A moving-average filter based hybrid ARIMA–ANN model for forecasting time series data. Appl Soft Comput 23:27–38

Bahdanau D, Cho K, Bengio Y (2014) Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv: 14090473

Bandara K, Bergmeir C, Smyl S (2020) Forecasting across time series databases using recurrent neural networks on groups of similar series: a clustering approach. Expert Syst Appl 140:112896. https://doi.org/10.1016/j.eswa.2019.112896

Bedi J, Toshniwal D (2019) Deep learning framework to forecast electricity demand. Appl Energy 238:1312–1326

Bengio Y, Simard P, Frasconi P (1994) Learning long-term dependencies with gradient descent is difficult. IEEE Trans Neural Networks 5:157–166

Brochu E, Cora VM, De Freitas N (2010) A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. arXiv preprint arXiv: 10122599

Büyükşahin ÜÇ, Ertekin Ş (2019) Improving forecasting accuracy of time series data using a new ARIMA-ANN hybrid method and empirical mode decomposition. Neurocomputing 361:151–163. https://doi.org/10.1016/j.neucom.2019.05.099

Chen W, Yeo CK, Lau CT, Lee BS (2018) Leveraging social media news to predict stock index movement using RNN-boost. Data Knowl Eng 118:14–24. https://doi.org/10.1016/j.datak.2018.08.003

Chollet F (2015) Keras. https://github.com/fchollet/keras. Accessed January 12, 2020

Chorowski J, Bahdanau D, Serdyuk D, Cho K, Bengio Y (2015) Attention-based models for speech recognition. In: The 28th international conference on neural information processing systems, Montreal, Canada. MIT Press, pp 577–585

de Oliveira JF, Ludermir TB (2016) A hybrid evolutionary decomposition system for time series forecasting. Neurocomputing 180:27–34

Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30

Farzad A, Mashayekhi H, Hassanpour H (2019) A comparative performance analysis of different activation functions in LSTM networks for classification. Neural Comput Applic 31:2507–2521. https://doi.org/10.1007/s00521-017-3210-6

Fischer T, Krauss C (2018) Deep learning with long short-term memory networks for financial market predictions. Eur J Oper Res 270:654–669

Fu X, Yang J, Li J, Fang M, Wang H (2018) Lexicon-enhanced LSTM with attention for general sentiment analysis. IEEE Access 6:71884–71891. https://doi.org/10.1109/ACCESS.2018.2878425

Gao W, Darvishan A, Toghani M, Mohammadi M, Abedinia O, Ghadimi N (2019) Different states of multi-block based forecast engine for price and load prediction. Int J Electr Power Energy Syst 104:423–435. https://doi.org/10.1016/j.ijepes.2018.07.014

Ghadimi N, Akbarimajd A, Shayeghi H, Abedinia O (2018a) A new prediction model based on multi-block forecast engine in smart grid. J Ambient Intell Human Comput 9:1873–1888. https://doi.org/10.1007/s12652-017-0648-4

Ghadimi N, Akbarimajd A, Shayeghi H, Abedinia O (2018b) Two stage forecast engine with feature selection technique and improved meta-heuristic algorithm for electricity load forecasting. Energy 161:130–142. https://doi.org/10.1016/j.energy.2018.07.088

Ghadimi N, Akbarimajd A, Shayeghi H, Abedinia O (2019) Application of a new hybrid forecast engine with feature selection algorithm in a power system. Int J Ambient Energy 40:494–503. https://doi.org/10.1080/01430750.2017.1412350

Graves A (2013) Generating sequences with recurrent neural networks. https://arxiv.org/. Accessed January 10, 2020

Gundu V, Simon SP (2020) PSO–LSTM for short term forecast of heterogeneous time series electricity price signals. J Ambient Intell Human Comput. https://doi.org/10.1007/s12652-020-02353-9

Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9:1735–1780

Hyndman R, Koehler AB, Ord JK, Snyder RD (2008) Forecasting with exponential smoothing: the state space approach. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-71918-2

Khandelwal I, Adhikari R, Verma G (2015) Time series forecasting using hybrid ARIMA and ANN models based on DWT decomposition. Procedia Comput Sci 48:173–179

Khashei M, Bijari M (2011) A novel hybridization of artificial neural networks and ARIMA models for time series forecasting. Appl Soft Comput 11:2664–2675

Kim J, Moon N (2019) BiLSTM model based on multivariate time series data in multiple field for forecasting trading area. J Ambient Intell Human Comput. https://doi.org/10.1007/s12652-019-01398-9

Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv preprint arXiv: 14126980

Kulshrestha A, Krishnaswamy V, Sharma M (2020) Bayesian BILSTM approach for tourism demand forecasting. Ann Tourism Res 83:102925. https://doi.org/10.1016/j.annals.2020.102925

Kumaresan K, Ganeshkumar P (2020) Software reliability prediction model with realistic assumption using time series (S)ARIMA model. J Ambient Intell Human Comput. https://doi.org/10.1007/s12652-020-01912-4

Law T, Shawe-Taylor J (2017) Practical Bayesian support vector regression for financial time series prediction and market condition change detection. Quant Financ 17:1403–1416

Law R, Li G, Fong DKC, Han X (2019) Tourism demand forecasting: a deep learning approach. Ann Tourism Res 75:410–423

Li J, Tu Z, Yang B, Lyu MR, Zhang T (2018) Multi-head attention with disagreement regularization. Paper presented at the 2018 conference on empirical methods in natural language processing. Belgium, Brussels

Martínez F, Frías MP, Pérez MD, Rivera AJ (2019) A methodology for applying *k*-nearest neighbor to time series forecasting. Artif Intell Rev 52:2019–2037. https://doi.org/10.1007/s10462-017-9593-z

Martínez F, Frías MP, Pérez-Godoy MD, Rivera AJ (2018) Dealing with seasonality by narrowing the training set in time series forecasting with kNN. Expert Syst Appl 103:38–48

Mir M, Shafieezadeh M, Heidari MA, Ghadimi N (2020) Application of hybrid forecast engine based intelligent algorithm and feature selection for wind signal prediction. Evolv Syst 11:559–573. https://doi.org/10.1007/s12530-019-09271-y

Murray PW, Agard B, Barajas MA (2018) Forecast of individual customer's demand from a large and noisy dataset. Comput Ind Eng 118:33–43

Nayak SC, Misra BB, Behera HS (2019) Efficient financial time series prediction with evolutionary virtual data position exploration. Neural Comput & Applic 31:1053–1074. https://doi.org/10.1007/s00521-017-3061-1

Olah C (2015) Understanding lstm networks. http://colah.github.io/posts/2015-08-Understanding-LSTMs. Accessed 20 Nov 2019

Panigrahi S, Behera HS (2017) A hybrid ETS–ANN model for time series forecasting. Eng Appl Artif Intell 66:49–59

Parmezan ARS, Souza VM, Batista GE (2019) Evaluation of statistical and machine learning models for time series prediction: identifying the state-of-the-art and the best conditions for the use of each model. Inform Sci 484:302–337

Prechelt L (2012) Early stopping: but When? In: Montavon G, Orr GB, Müller K-R (eds) Neural Networks: tricks of the trade: second edition. Springer Berlin Heidelberg, Berlin, Heidelberg 53–67 https://doi.org/10.1007/978-3-642-35289-8_5

Reimers N, Gurevych I (2017) Optimal hyperparameters for deep lstm-networks for sequence labeling tasks. arXiv preprint arXiv: 170706799

Sagheer A, Kotb M (2019) Time series forecasting of petroleum production using deep LSTM recurrent networks. Neurocomputing 323:203–213

Samet H, Reisi M, Marzbani F (2019) Evaluation of neural network-based methodologies for wind speed forecasting. Comput Electr Eng 78:356–372. https://doi.org/10.1016/j.compeleceng.2019.07.024

Sangeetha K, Prabha D (2020) Sentiment analysis of student feedback using multi-head attention fusion model of word and context embedding for LSTM. J Ambient Intell Human Comput. https://doi.org/10.1007/s12652-020-01791-9

Sengar S, Liu X (2020) Ensemble approach for short term load forecasting in wind energy system using hybrid algorithm. J Ambient Intell Human Comput. https://doi.org/10.1007/s12652-020-01866-7

Shankar S, Ilavarasan PV, Punia S, Singh Surya P (2019) Forecasting container throughput with long short-term memory networks. Ind Manage Data Syst 120:425–441. https://doi.org/10.1108/IMDS-07-2019-0370

Takahashi S, Chen Y, Tanaka-Ishii K (2019) Modeling financial time-series with generative adversarial networks. Phys A 527:121261. https://doi.org/10.1016/j.physa.2019.121261

Vaswani A et al (2017) Attention is all you need. In: 31st international conference on neural information processing systems, Long Beach, California, USA. Curran Associates Inc, pp 6000–6010

Zhang GP (2003) Time series forecasting using a hybrid ARIMA and neural network model. Neurocomputing 50:159–175. https://doi.org/10.1016/S0925-2312(01)00702-0