Neurocomputing 388 (2020) 269–279

Contents lists available at ScienceDirect

Neurocomputing

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

Multivariate time series forecasting via attention-based encoder–decoder framework

Shengdong Du[a], Tianrui Li[a,*], Yan Yang[a], Shi-Jinn Horng[b,*]

[a] School of Information Science and Technology, Southwest Jiaotong University, Chengdu 611756, China
[b] Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei 10607, Taiwan

ARTICLE INFO

Article history:
Received 22 March 2019
Revised 9 December 2019
Accepted 31 December 2019
Available online 15 January 2020

Communicated by Dr. Wojciech Froelich

Keywords:
Multivariate time series
Temporal attention
Multi-step forecasting
Encoder–decoder
Deep learning
Long short-term memory networks

ABSTRACT

Time series forecasting is an important technique to study the behavior of temporal data and forecast future values, which is widely applied in many fields, e.g. air quality forecasting, power load forecasting, medical monitoring, and intrusion detection. In this paper, we firstly propose a novel temporal attention encoder–decoder model to deal with the multivariate time series forecasting problem. It is an end-to-end deep learning structure that integrates the traditional encode context vector and temporal attention vector for jointly temporal representation learning, which is based on bi-directional long short-term memory networks (Bi-LSTM) layers with temporal attention mechanism as the encoder network to adaptively learning long-term dependency and hidden correlation features of multivariate temporal data. Extensive experimental results on five typical multivariate time series datasets showed that our model has the best forecasting performance compared with baseline methods.

1. Introduction

Time series forecasting has received much attention in recent decades due to its important applications in many fields [1], including traffic flow forecasting [2], air pollution forecasting [3], time series anomaly detection [4], medical monitoring analysis [5], network intrusion detection, etc. Generally speaking, time series data can be described as a set of observations in chronological order. Its type can be divided into univariate time series and multivariate time series, which have the characteristics of a large scale in data size, high dimensionality, and constant change. Many scholars have been studying the problem of time series prediction, especially based on classical statistical model such as ARIMA [6], and typical machine learning models like HMM [8], SVR [7], and ANN [9–11]. But many traditional methods mostly employ statistical model to research the evolution of temporal data. For example, Zhang proposed a hybrid method that combines both ARIMA and ANN for linear and nonlinear time series modeling [9]. Pai et al. presented a hybrid SSVR method for forecasting time series by employing SARIMA and SVR models [7]. Sapankevych and Sankar provided a survey of time series prediction applications using a classical machine learning model: Support Vector Machines (SVM), which provides us with insight into the advantages and challenges using SVM for time series forecasting [12].

With the arrival of the big data era, multivariate and multichannel massive time series data are increasing explosively. In many cases, multivariate time series data has high dimensional and spatial-temporal dependency characteristics, or contains noisy data, which makes it difficult to be modeled effectively by classical statistical methods [10]. In addition, those traditional methods face difficulties in processing big data, especially massive and complex multivariate temporal data. Therefore, data-driven time series prediction methods are increasingly favored by researchers, and some researchers have made progress in many fields, e.g. air pollution forecasting [3,13], traffic flow prediction [14], anomaly detection [4], and urban crowd flow prediction [15]. Recently, deep learning [16,17] has been used in a lot of areas and achieved the best results on many benchmark data sets, e.g. image [18] and video processing [19], natural language understanding [20]. Although traditional methods can still be used for time series modeling, deep learning-based forecasting methods are becoming more popular [21–23].

In addition, the traditional time series forecasting methods also face the following two challenges: firstly, many classical methods mostly solve the single-step prediction problem, which is limited in practical monitoring and early warning applications based on time series modeling. The single-step time series forecasting usually does not help a lot in a real early-warning application since it is difficult to forecast what is going to happen after a

* Corresponding authors.
E-mail address: trli@swjtu.edu.cn (T. Li).

https://doi.org/10.1016/j.neucom.2019.12.118
0925-2312/

multi-step condition. Moreover, multi-step forecasting is more complicated than single-step forecasting, which must consider more conditions, e.g. accumulation of errors and degradation of forecasting performance [24]. Secondly, in some cases, there are correlations between variables in multivariate time series data, and a better understanding is often obtained by modeling all related variables together than just by modeling one variable. So it is significant to study the prediction model based on multivariable time series data.

In response to the above two issues, a Bi-LSTM based encoder–decoder with attention mechanism is proposed for the first time, which aims to achieve the purpose of adaptively learning the implicit temporal dependency features of multivariate time series data. And the contributions of this paper include the following aspects:

1) Firstly, we propose a novel temporal attention-based encoder–decoder model and apply it to the multivariate time series multi-step forecasting tasks. The Bi-LSTM with attention mechanism is used to encode the hidden representations of multivariate time series data as the temporal context vector, and the other LSTM is used to decode the hidden representation for prediction. Through this end-to-end process, hidden long-term dependent features and non-linear correlation features can be learned from the raw multivariate time series data.
2) Secondly, we introduce the temporal attention mechanism between the encoder network and decoder network, which can select relevant encoder hidden states across all time-steps for multi-step forecasting more accurately, so as to improve our model's representation ability of dynamic multivariate time series data.
3) Finally, we demonstrate the effectiveness of our model by testing it on five actual multivariate time series datasets, and the experiment results showed that our model has the best forecasting performance compared with baseline methods.

The rest of the paper is arranged as follows: Section 2 gives an overview of related works on time series modeling and forecasting. Section 3 expounds the research motivation of the proposed model, analyzes the overall architecture of the attention encoder–decoder framework, and describes the relevant theory and process details. Section 4 is the experimental analysis content. A comparative experiment based on five multivariate time series data sets is performed, and the experimental results are analyzed in detail. Finally, we give the conclusions of the study and discuss future research priorities.

2. Related works

The key to time series modeling is to design effective feature representation methods for dynamic time series data, which is a challenging topic and faces issues such as high dimensional, dynamic and uncertainty, and it involves a wide range of aspects and has always been the focus of researchers. Traditional time series forecasting methods often use statistical models or typical machine learning methods [1], e.g. ARIMA [6], SVR [7], HMM [8], and ANN [9]. And Ahmed et al. made a comparative analysis and research on time series prediction methods based on traditional machine learning models [25]. Park et al. proposed a continuous HMM model to deal with the financial time series prediction task [8]. De Gooijer and Hyndman reviewed the past 25 years of research on time series forecasting and found that there are still a lot of problems in need of further research [1]. Hassan et al. presented a fusion method with a fuzzy logic mechanism to process non-linear temporal data [26]. Fu provided a comprehensive review of the existing time series data mining methods [10]. In terms of multi-step forward time series forecasting problems, Taieb et al. reviewed existing strategies for multi-step-ahead forecasting and compared them in theoretical and practical terms [27].

In recent years, with the explosive growth of spatio-temporal big data, many collected temporal data usually have massive volumes, spatial-temporal dependence, dynamic and nonlinear characteristics. Due to the complexity and uncertainty of dynamic multivariate time series representation, classical machine learning and statistical methods are difficult to deal with the evolutionary characteristics of big temporal data [28]. So many researchers try to propose data-driven prediction methods like deep learning to deal with the problem [17,29]. Deep learning is considered to be the most successful representation learning method and has made significant progress in a lot of areas, e.g. images, videos, audios and natural language understanding tasks [18–20]. Moreover, the deep learning-based time series forecasting model has become a research hotspot [21,30–33]. For example, Yao et al. developed a time series forecasting model that integrates CNN and RNN [35]. Zheng et al. proposed a novel CNN based deep learning model for multivariate time series classification [34].

The encoder–decoder is a popular sequence-to-sequence structure, which is a simplified and automatic method for sequential data modeling [20,36,37]. It usually uses a deep neural network (e.g. RNN or LSTM) to encode the model input to a fixed vector, and then employs another deep neural network to decode the target output sequence from the fixed vector for prediction [38]. Chorowski et al. improved the performance of the speech recognition model by extending the attention mechanism [37]. Sutskever et al. proposed an end-to-end approach for sequence learning and demonstrated that the simple structure can achieve good performance [38]. The encoder–decoder model also can be used for temporal data processing. For example, Malhotra et al. developed a novel multi-sensor anomaly detection model, which used LSTM as the encoder and decoder component [48]. Park et al. employed the encoder–decoder model to analyze the implicit pattern of trajectory data [49]. Kuznetsov and Mariet firstly conducted a deep theoretical analysis of seq2seq deep learning model and used for time series forecasting [39].

Nevertheless, the encoder–decoder structure has not yet been well researched for temporal data modeling. Existing research methods rarely consider effective modeling of long-term temporal sequences from the perspective of attention mechanism, which is the reason to motivate us to investigate it in this paper. In addition, multi-step forward forecasting is more difficult than single-step forecasting of multivariate time series data, and sequence-to-sequence learning architecture can handle this problem well. In this paper, different from classical statistical and machine learning methods, we propose a multivariate time series multi-step forecasting model via an attention-based sequence-to-sequence learning structure. Experiments on five multivariate time series datasets showed that the proposed model can effectively forecast multi-step time series value under different conditions.

3. Methodology

3.1. Problem and definitions

Time series data are usually sequences of values (in discrete or continuous form) measured over time. The dynamic updating, uncertainty and high dimensionality of time series make it different from other data such as image, text and so on. Time series forecasting has always been a very important research area of data mining tasks, whose goal is to predict the change of future temporal values. And the observation time interval of different types of time series data is often different and decided by the sensor specifications, e.g. the observation time interval of traffic flow time

series data maybe 5 min, 15 min, 60 min, etc. and the observation time interval of a PM2.5 time series usually is one hour.

A multivariate time series typically contains multiple variable values at each observation time-step. Each variable of multivariate time series depends on not only its past value but also other variables value in some cases. These potential correlations are crucial for modeling multivariate time series data. How to learn the correlation features of the monitored multiple variables of multivariate time series datasets is important for modeling the temporal data, that is why we focus on multivariate time series forecasting problem in this paper.

Then we give a general definition of multivariate time series forecasting task [41,42], which goal is to anticipate the future temporal value f_{t+1} at time $t+1$ or f_{t+p} at time $t+p$ based on the historical time series dataset. The model input includes f_i itself and the other variables of multivariate time series. Taking traffic flow forecasting as an example, it not only includes flow variable, but also includes other variables, e.g. traffic speed, traffic density, and road length.

$$\left[MD = \begin{pmatrix} md_{1,t-l} & \cdots & md_{1,t} \\ \vdots & \ddots & \vdots \\ md_{n,t-l} & \cdots & md_{n,t} \end{pmatrix} \right] \overset{model}{\rightarrow} \begin{pmatrix} f_t \\ \vdots \\ f_{t+p} \end{pmatrix} \tag{1}$$

As shown in the above formula, multivariate time series multi-step forecasting task in this paper is represented as predicting the next p values of time series data $[f_t, f_{t+1}, \ldots, f_{t+p}]$ given a history multivariate time series dataset $MD = \{md_{i,j} \mid i = 1, 2.., n;\ j = t-l, \ldots, t-1, t$ in the past$\}$, where l denotes the lookup size of history data, n represents a variable number of input data, and p denotes the multi-step forward prediction size.

3.2. Overview of the proposed framework

A multivariate time series multi-step forecasting framework via attention-based encoder–decoder structure is proposed in this paper (as shown in Fig. 1), which has three components: Bi-LSTM as the encoder component, an LSTM as the decoder component and a temporal attention context layer as the attention component. The Bi-LSTM is used to learn the hidden representation of input data with arbitrary lengths, which extracts the deep temporal dependency features from the multivariate time series and then uses the temporal attention layer to construct latent space variables (temporal attention context vectors). The LSTM decoder is responsible for forecasting the future time series value, which is based on the generated latent space variables. Fig. 1 shows the graphical illustration of the proposed framework, which can model the multivariate time series data in an end-to-end process.

3.3. The encoder–decoder learning structure

LSTM is a popular scheme for learning long-term temporal dependency features of raw time series data [43]. The typical LSTM unit contains five components: i_t, f_t, s_t, o_t, h_t as the input gate, forget gate, memory cell, output gate and hidden state of the LSTM unit, respectively. As the lower right corner of Fig. 1 shows, a typical LSTM cell block includes three gates and one memory cell unit, which has the ability to forget or memory information (determine how much information should be transferred to the next cell) based on these components.

$$i_t = \sigma\left(U^{(i)}x_t + W^{(i)}h_{t-1} + b_i\right) \tag{2}$$

$$f_t = \sigma\left(U^{(f)}x_t + W^{(f)}h_{t-1} + b_f\right) \tag{3}$$

$$o_t = \sigma\left(U^{(o)}x_t + W^{(o)}h_{t-1} + b_o\right) \tag{4}$$

$$\tilde{s}_t = tanh\left(U^{(c)}x_t + W^{(c)}h_{t-1} + b_c\right) \tag{5}$$

$$s_t = f_t \odot s_{t-1} + i_t \odot \tilde{s}_t \tag{6}$$

$$h_t = o_t \odot tanh(s_t) \tag{7}$$

As shown in the above formulas, for an input time series value at t time-step, the LSTM unit computes a hidden state h_t and a memory state s_t which is an encoding of input time series values until t time-step, σ is the activation function, and $\odot$ is the element-wise multiplication.

However, traditional LSTM has a weak point. It can only learn the previous context of time series data, and cannot learn the forward context of same sequence data. So we use Bi-LSTM as the encoder, which can simultaneously process sequential data in different directions through two interconnected hidden layers: one direction process uses a forward hidden layer from $t = 1$ to T; the other direction process uses a backward hidden layer from $t = T$ to 1. The classical Bi-LSTM computing process is as follows:

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

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

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

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

$$\vec{s}_t = \vec{f}_t \odot \vec{s}_{t-1} + \vec{i}_t \odot \overrightarrow{\tilde{s}_t} \tag{12}$$

$$\vec{h}_t = \vec{o}_t \odot tanh(\vec{s}_t) \tag{13}$$

$$\overleftarrow{i}_t = \sigma\left(\overleftarrow{U}^{(i)}\overleftarrow{x}_t + \overleftarrow{W}^{(i)}\overleftarrow{h}_{t-1} + \overleftarrow{b}_i\right) \tag{14}$$

$$\overleftarrow{f}_t = \sigma\left(\overleftarrow{U}^{(f)}\overleftarrow{x}_t + \overleftarrow{W}^{(f)}\overleftarrow{h}_{t-1} + \overleftarrow{b}_f\right) \tag{15}$$

$$\overleftarrow{o}_t = \sigma\left(\overleftarrow{U}^{(o)}\overleftarrow{x}_t + \overleftarrow{W}^{(o)}\overleftarrow{h}_{t-1} + \overleftarrow{b}_o\right) \tag{16}$$

$$\overleftarrow{\tilde{s}_t} = tanh\left(\overleftarrow{U}^{(c)}\overleftarrow{x}_t + \overleftarrow{W}^{(c)}\overleftarrow{h}_{t-1} + \overleftarrow{b}_c\right) \tag{17}$$

$$\overleftarrow{s}_t = \overleftarrow{f}_t \odot \overleftarrow{s}_{t-1} + \overleftarrow{i}_t \odot \overleftarrow{\tilde{s}_t} \tag{18}$$

$$\overleftarrow{h}_t = \overleftarrow{o}_t \odot tanh\left(\overleftarrow{s}_t\right) \tag{19}$$

$$h_t = \vec{h}_t \circ \overleftarrow{h}_t \tag{20}$$

As shown in the above formulas, the arrow represents the processing direction and h_t denotes the final hidden output of Bi-LSTM, which is concatenated by the forward output $\vec{h}_t$ and backward output $\overleftarrow{h}_t$. We use Bi-LSTM as the encoder component, and an LSTM as the decoder component, which forecasts multi-step time series value by generating target value conditioned on the previous temporal context and the previously generated value. In other words, the Bi-LSTM encoder is responsible for encoding the input data to a context generation C_T. It is a fixed vector as the temporal representation of the input data. The LSTM decoder then

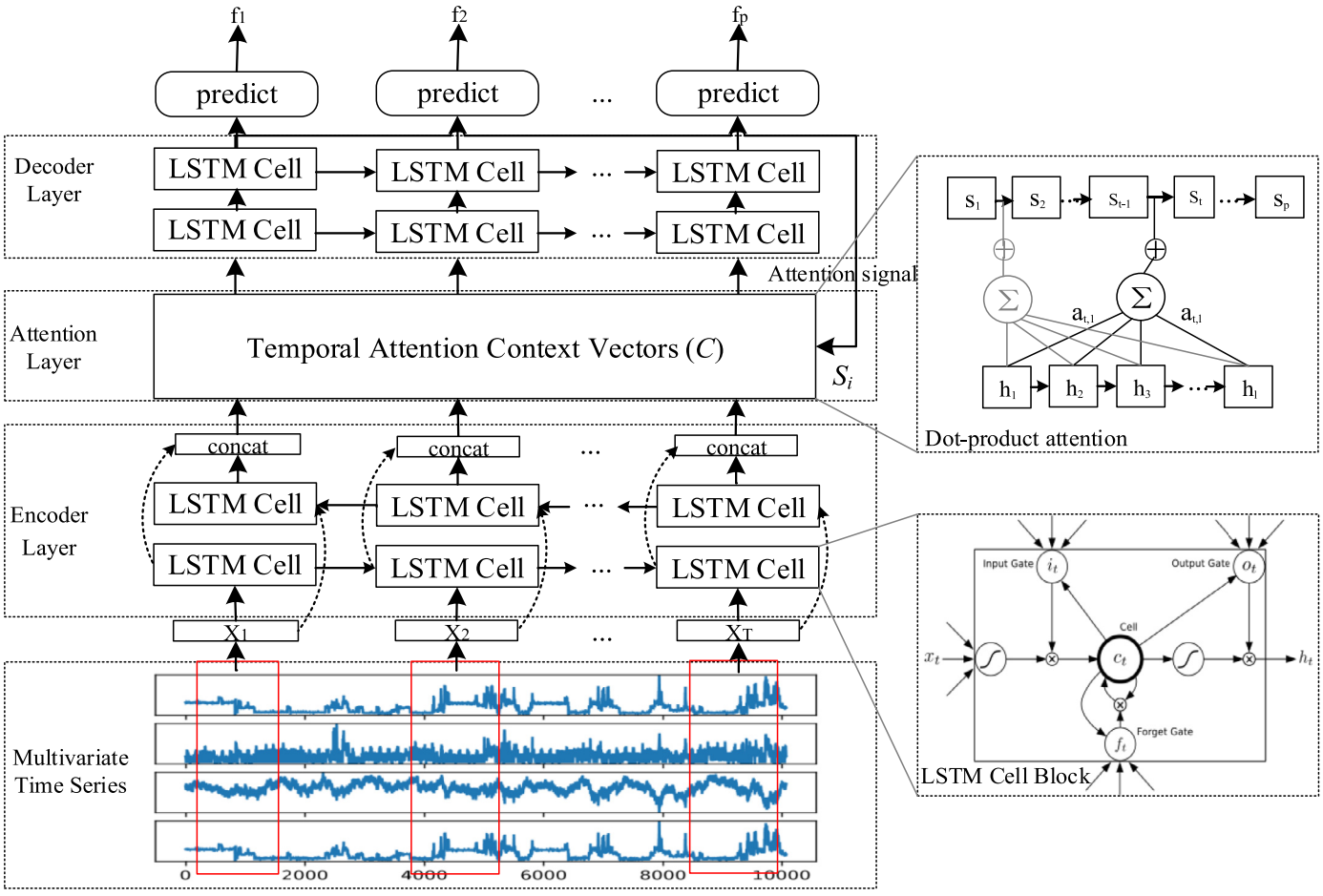

Fig. 1. Graphical illustration of the multivariate time series forecasting framework via a temporal attention-based encoder–decoder model (MTSMFF for short), where the temporal attention context layer is designed to select the best frames of the encoder hidden representation and make the decoder attend to these frames for more accurate forecasting.

decodes C_T and generates a target sequence $(f_1, f_2, \ldots, f_P)$ as the multi-step forward prediction value. The probability of the target sequence can be computed as follows:

$$p(f_1, f_2, \ldots, f_P \mid X_1, X_2, \ldots, X_T) = \prod_{t=1}^{P} p(f_t \mid C_T, f_1, f_2, \ldots, f_{t-1}) \tag{21}$$

where $(X_1, X_2, \ldots, X_T)$ is the model input of history multivariate time series, X_i denotes the ith time-step observation variables of temporal data, and $(f_1, f_2, \ldots, f_P)$ is the multi-step forecasting target value whose length P can differ from the input length T (called lookup size or window size). As mentioned above, we present the calculation process of a classic encoder–decoder [20], which can be used to model multivariate time series data and complete a multi-step forecasting task.

3.4. Temporal attention mechanism

However, the classic encoder–decoder deep learning structure has defects that the encoder must compress all hidden representation of the past temporal information into a fixed-length context vector C_T. Therefore, the prediction ability will gradually degrade as the length of input time series increases [44]. In view of this difficulty, this paper first proposes a temporal attention mechanism for processing multivariate time series forecasting tasks. A temporal attention layer is designed and marked as the interface between the encoder and the decoder. We use Bi-LSTM as the encoder, which can take a variable-length time series sequence $x = (x_1, x_{2,\ldots}, x_T)$ as input by recursively processing and maintaining its internal hidden state h. At each time step t, the LSTM reads x_t and updates its hidden state h_t as follows:

$$\vec{h}_t = LSTM\left(\mathbf{x}_t, \vec{h}_{t-1}\right); \overleftarrow{h}_t = LSTM\left(\mathbf{x}_t, \overleftarrow{h}_{t-1}\right); h_t = \vec{h}_t \oplus \overleftarrow{h}_t \tag{22}$$

where the arrow represents the processing direction and h_t denotes the output of Bi-LSTM, which is concatenated by the forward output $\vec{h}_t$ and backward output $\overleftarrow{h}_t$. Then temporal attention context vectors are generated as a weighted sum of the hidden states of the encoder network based on Bi-LSTM, which is used to select the best frames of the encoder hidden representation and make the decoder attend to these frames. The temporal attention layer computing process can be described as follows:

$$e_{i,t} = \text{Align}(s_{i-1}, h_t) = s_{i-1}^T \odot h'_t \tag{23}$$

$$\alpha_{i,t} = \frac{\exp(e_{i,t})}{\sum_{k=1}^{T} \exp\left(e_{i,k}\right)} \tag{24}$$

$$h_a = \sum_{t=1}^{T} \alpha_{i,t} h_t \tag{25}$$

As shown in the above formulas, formula (23) represents the soft align computation between the hidden state s_{i-1} of the decoder layer and hidden state h_t of the encoder layer. Formula

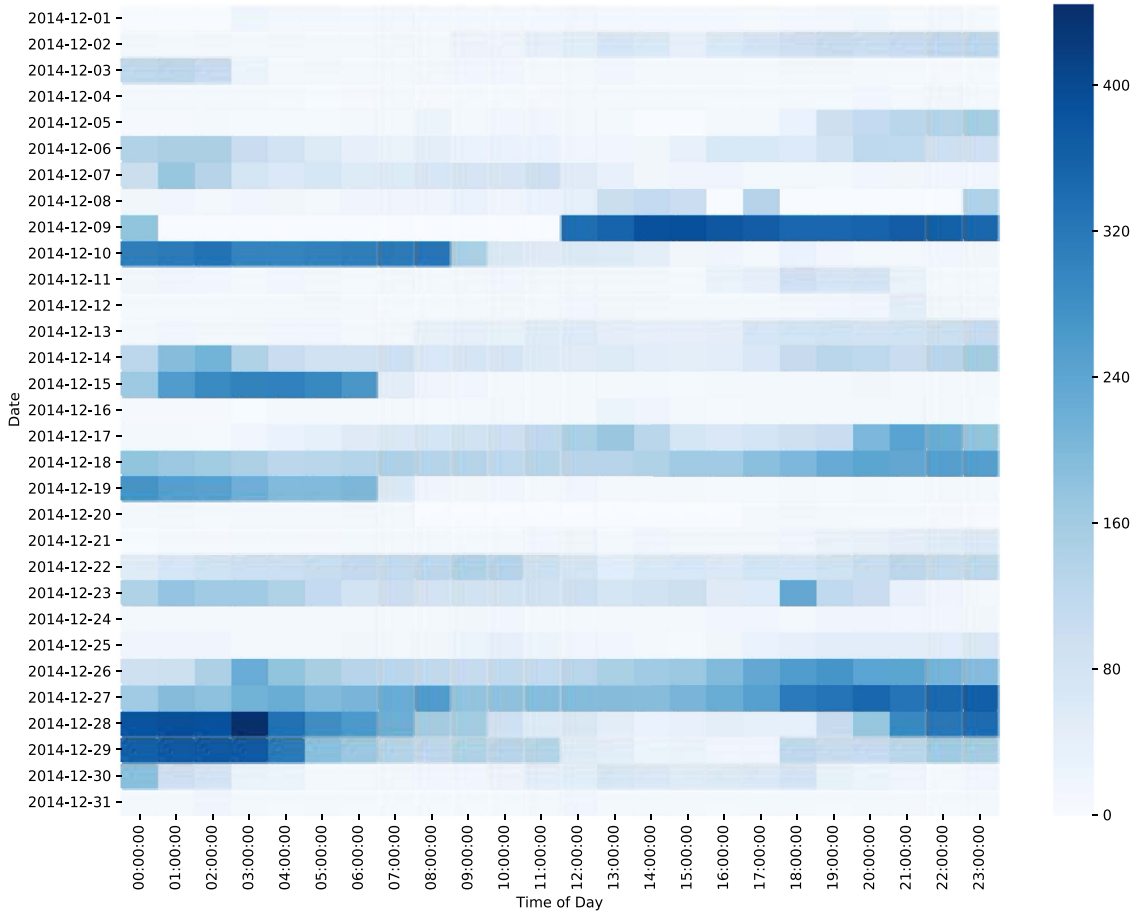

Fig. 2. Heatmap visualization of PM2.5 value during one month (01/12/2014–31/12/2014), from Beijing air quality dataset [40].

(24) indicates the attention weights that correspond to the importance of the input time series frame at time-step t to predict the output value at time-step i, which use the softmax function to normalize the vector e_i of length T as the attention mask over the input time series sequence. The h_a as formula (25) is the final state of the attention layer. The modal training problem is to minimize the negative log-likelihood of a historical training set $MD = \{X_i, f_i\}_{i=1}^{n}$, which can be described as follows:

$$\underset{\boldsymbol{\theta}}{\operatorname{argmin}} C = -\sum_{i=1}^{n} \log p(f_i \mid X_i; \boldsymbol{\theta}) + \lambda \left|\left|\boldsymbol{\theta}^2\right|\right| \tag{26}$$

where n represents the number of training samples, $\boldsymbol{\theta}$ is the parameter space of the model, which includes W and b of each layer, and λ controls the importance of the penalty or regularization term of the loss function.

4. Experiments

In this section, we evaluate the forecasting ability of the proposed model on five public multivariate time series datasets. Through comparison of the baseline shallow learning and deep learning models, the forecasting performance and effectiveness of the proposed model are validated.

4.1. Datasets

Details of the five real multivariate time series datasets (as shown in Table 1) are described as follows.

Beijing PM2.5 Data Set (Beijing PM25): It is a typical multivariate time series data that includes multiple variables, e.g. temperature, pressure, wind direction, wind speed, snow or rain conditions and PM2.5 value itself, which is published on the UCI machine learning repository website [40].

Table 1
Details of all experiment datasets.

Dataset	Instances number	Variables number	Time interval
Beijing PM25	43,824	9	1 h
Power consumption	2,075,259	7	1 min
Italian air quality	9358	15	1 h
Highways traffic	37,564	10	15 min
PeMS-Bay	52,117	14	15 min

Individual household electric power consumption Data Set (Power Consumption): The dataset contains 2,075,259 measurements of electric power consumption and is gathered with a one-minute sampling rate over a period of almost 4 years in a house of Sceaux [40].

Air Quality Data Set (Italian Air Quality): The dataset contains 9358 instances of hourly averaged responses from an array of 5 metal oxide chemical sensors, which was located in a significantly polluted area of Italian city [40].

Highways England Traffic Dataset (Highways Traffic): This dataset provides average journey time, traffic speed and traffic flow related variables for 15-min periods, which is derived from Highways England Traffic Data from Opening up Government of UK [45].

PeMS-Bay: This traffic dataset is collected by California Transportation Agencies Performance Measurement System (PeMS). We

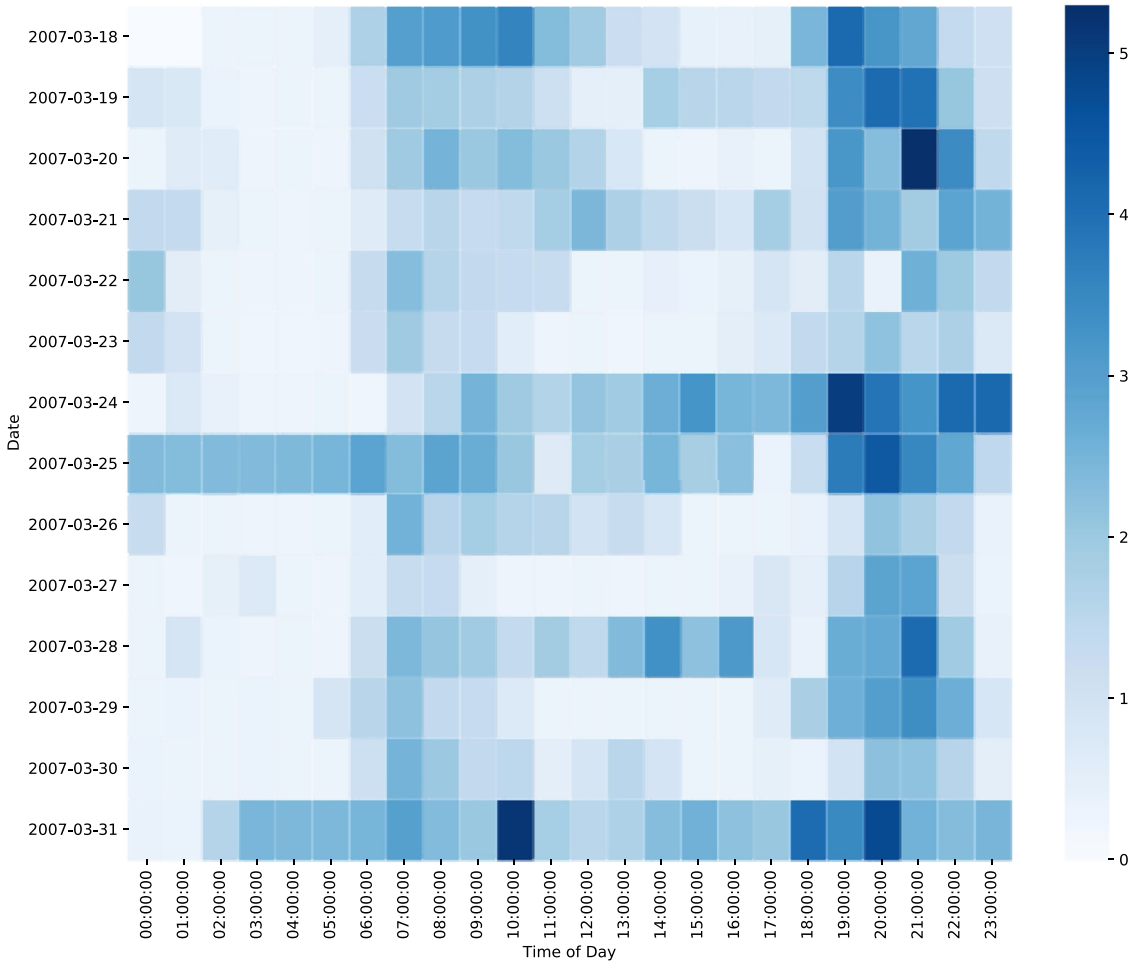

Fig. 3. Heatmap visualization of the global_active_power value of 14 days (18/03/2007–31/03/2007), from individual household electric power consumption dataset [40].

select 14 sensors in the Bay Area and collect 6 months of data ranging from 01/01/2017 to 05/31/2017 for the experiment [46].

We select two of the datasets for visual exploration to determine that the distribution characteristics of the selected experiments datasets are not the same. We visually explore the two datasets through a heat map. A heat map can visually display the density relationships through two dimensions, i.e. date and observed time-step. It helps us obtain instant insight into the distribution of time series data for each observed time-step of each day. As shown in Figs. 2 and 3, the darker the color, the higher the time-series value density during the time period.

Fig. 2 is the heat map visualization of PM2.5 value during one month (01/12/2014–31/12/2014) from Beijing air quality data set, which contains PM2.5 readings taken every hour. The *X*-axis shows observed time-steps of each day (include 24 points) and the *Y*-axis represents one month period. As Fig. 2 shows, the daily density color of PM2.5 value is the deepest during the morning (0:00–4:00) and evening (19:00–24:00) time period, which is also in line with the actual air pollution situation. However, during the whole month of December, there were no obvious rules on which days were polluted seriously and which days were polluted lightly.

Fig. 3 is heat map visualization of global active power value during half-month (18/03/2007–31/03/2007), which contains one household electric power consumption readings taken every minute [40]. The *X*-axis shows observed time-steps (sampling every one hour) of each day and the *Y*-axis represents 14 days period. As can be seen from Fig. 3, the daily density color of electric power consumption value is the deepest during the morning (6:00–9:00) and evening (16:00–22:00) time period, which is also in line with the actual situation of household life. In addition, the distribution of power consumption data during half a month indicates that the power consumption time series is more regular than the air pollution time series. The daily consumption is not much different (except for the weekend, e.g. day 21 and day 25), which is relatively average.

4.2. Experimental setup

The experiment environments are as follows: we use open-source machine learning library Scikit-learn and deep learning framework Keras to implement the baseline models and our model. The experimental hardware environment is configured with Intel (R) CPU E5-2623 3.00 GHz, each of the 4 GPUs is 12G NVIDIA Tesla K80C, and the memory is 128 GB.

In the experiment, several benchmark models are used to compare the multi-step forecasting performance of five real multivariate time series datasets, including typical classic shallow learning models SVR (with different kernel functions), four baseline deep learning models (RNN, CNN, LSTM, GRU), and the baseline sequence-to-sequence (seq2seq for short) deep learning model. The detailed descriptions of the above models are as follows.

ARIMA and VARMA are the classical statistical models for analyzing and forecasting time series data. SVR is a classical kernel method, which includes three kinds of kernel functions as RBF, POLY, and LINEAR.

Table 2
Hyperparameter settings of the experimental models.

Model	Parameter	Value
Baseline deep learning models	Default hidden layers	1
	Units in hidden layers	100
	Training epochs	100
	Dropout	0.3
	Batch size	96
	Loss function	mse
	Optimizer	adagrad
seq2seq model and our model	Default hidden layers (encoder)	1
	Default hidden layers (decoder)	1
	Bidirectional-lstm merge mode	sum
	The active function of attention layer	softmax
	Units in hidden layers	100
	Training epochs	100
	Dropout	0.3
	Batch size	96
	Loss function	mse
	Optimizer	adagrad

CNN is a useful type of deep neural networks for image and video processing, which can also be used for time series forecasting (such as one dimensional CNN, or convert the sequence data into a 2D image for processing).

RNN is a classical deep neural network. LSTM and GRU are variants of the classical RNN model, which are often used to deal with sequential learning tasks.

SEQ2SEQ is a classic sequence-to-sequence deep learning model for sequential data processing.

SEQ2SEQ-BI is a simple variant of our model, using Bi-LSTM as the encoder component.

SEQ2SEQ-ATT is a simple variant of our model, using LSTM with attention mechanism as the encoder component.

In terms of model parameter settings, we try to keep the benchmark consistent so the same values of most parameters are used for baseline and our model.

For the baseline deep learning models (e.g. RNN and LSTM), we use the same parameters of neural network architecture configuration (Table 2), which use one hidden recurrent layer with 100 neural units, use mean square error (MSE) as the loss function, and use Adagrad function as the model optimizer [47]. The batch size sets to 96 and drop out rate sets to 0.3. The training process is done for 100 epochs. For the seq2seq and our model, we use one Bi-LSTM as a hidden layer of the encoder and use an LSTM as the hidden layer of the decoder. We select sum function as the Bi-LSTM merge mode and softmax as the activating function of the temporal attention context layer. We train the proposed model with the same parameter values of baseline deep learning models.

Additionally, all experiment datasets are split by retaining the first 80% of each dataset for training-validation and the last 20% for testing. We use the min–max function to normalize the input data to [0, 1]. Finally, we use RMSE (Root Mean Square Error) and MAE (Mean Absolute Error) as the model error evaluation metrics.

4.3. Results analysis

In this section, we summarize the results of extensive experiments to analyze the forecasting performance of baseline models and our model. Table 3 shows the results of multi-step prediction experiments for five actual multivariate time series datasets and gives a comparative analysis of the average RMSE and MAE of baseline models and our model MTSMFF. The model error is the average of the prediction errors (RMSE and MAE) for each time-step from the 1st time step to the 6th time step ($t1$–$t6$). As shown in Table 3, the experimental results show that our model has the smallest prediction error on all data sets, and significantly improves the forecasting accuracy compared to traditional methods. Further analysis shows that RMSE and MAE of the baseline deep neural network models are similar and lower than that of shallow learning models (e.g. SVR and ARIMA) in most cases. In addition, in some cases, we find that the traditional statistical model and the baseline deep learning model have different prediction performance on different data sets. For example, VARMA has a larger test error than LSTM in the experiment on the Beijing PM25 dataset. However, the result is the opposite in the experiment on the Highways Traffic dataset.

Next, we analyze the results of multi-step prediction experiments based on two data sets (Beijing PM2.5 and Power Consumption). As shown in Table 4, compared with the baseline models, the model proposed in this paper has better prediction performance, no matter under the condition of short-term time-step prediction or long-term time-step prediction condition. It is found that when the forward prediction size is 1, our model's error on the Beijing PM2.5 dataset is reduced to 23.52. When the forward prediction size is 6, the model error is 46.15 and still kept to the minimum. In addition, our model has the lowest test error on another power consumption dataset compared to other baseline methods. Experimental results show that our model improves the multi-step forecasting accuracy compared with baseline models, which performance is also better than the classic encoder–decoder deep learning (SEQ2SEQ) model and it's variant models

Table 3
A comparison of the average prediction error (RMSE and MAE) of our model MTSMFF with other baseline models on five datasets

Models	Average model prediction error on different datasets									
	Beijing PM25		*Power consumption*		*Italian air quality*		*Highways traffic*		*PeMS-Bay*	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
ARIMA	53.41	41.06	0.87	0.69	59.07	50.66	19.92	19.04	5.44	4.37
VARMA	51.78	38.95	0.79	0.66	58.44	47.32	15.10	10.79	5.42	4.21
SVR-LINEAR	88.35	71.48	1.45	1.23	72.39	65.48	36.01	28.85	6.51	5.53
SVR-RBF	57.64	43.47	0.76	0.71	59.02	48.49	25.20	19.52	5.59	4.43
RNN	44.42	27.99	0.46	0.23	53.35	40.16	17.27	12.29	3.95	1.98
CNN	42.98	26.89	0.46	0.24	54.13	41.07	17.05	12.11	3.86	1.94
LSTM	43.12	27.04	0.44	0.23	55.19	41.60	15.82	10.85	3.87	1.96
GRU	43.04	26.95	0.47	0.25	55.02	41.36	16.64	11.67	3.84	1.92
SEQ2SEQ	42.53	26.42	0.46	0.24	54.89	41.25	17.80	12.45	3.85	1.95
SEQ2SEQ-BI	42.29	26.31	0.44	0.23	54.45	41.21	16.33	12.01	3.77	1.86
SEQ2SEQ -ATT	42.15	26.29	0.44	0.23	54.22	41.18	16.27	11.95	3.76	1.84
MTSMFF(Ours)	**39.83**	**25.45**	**0.42**	**0.21**	**53.28**	**40.25**	**15.07**	**10.38**	**3.09**	**1.72**

Note: The model error is the average of the prediction errors (RMSE and MAE) for each time-step from the 1st time step to the 6th time step (t1–t6). And the main parameters of deep learning models are set as epochs-100, batch size-96, dropout-0.3, optimizer-Adagrad, loss function-MSE.

Table 4
A comparison of RMSE of our model MTSMFF with other baseline models for the multivariate time series multi-step forecasting task.

Models	*RMSE*											
	Beijing PM25 *data set*						Power Consumption *data set*					
	t+1	*t*+2	*t*+3	*t*+4	*t*+5	*t*+6	*t*+1	*t*+2	*t*+3	*t*+4	*t*+5	*t*+6
ARIMA	24.21	41.17	56.43	62.96	65.28	70.38	0.75	0.82	0.89	0.90	0.94	0.96
VARMA	24.10	40.05	55.27	60.36	62.46	68.47	0.71	0.73	0.79	0.82	0.84	0.85
SVR-LINEAR	38.73	47.39	66.3	163.5	120.34	93.89	0.97	1.44	2.52	0.88	1.49	1.40
SVR-RBF	45.02	49.97	55.29	61.66	64.52	69.41	0.67	0.74	0.78	0.77	0.78	0.83
RNN	24.93	35.23	43.27	50.11	54.49	58.48	0.29	0.41	0.48	0.50	0.54	0.57
CNN	24.57	34.26	41.98	48.02	52.68	56.37	0.28	0.40	0.46	0.51	0.53	0.56
LSTM	24.16	34.35	42.06	48.14	53.04	57.01	0.27	0.38	0.44	0.48	0.52	0.54
GRU	23.99	34.34	41.95	48.08	52.94	56.97	0.29	0.40	0.48	0.52	0.54	0.57
SEQ2SEQ	24.42	33.36	41.83	48.07	52.17	55.34	0.29	0.40	0.46	0.50	0.53	0.56
SEQ2SEQ-BI	23.86	33.42	42.05	48.13	52.11	54.20	0.28	0.39	0.46	0.48	0.51	0.55
SEQ2SEQ -ATT	23.94	33.87	42.33	48.09	51.25	53.41	0.28	0.40	0.45	0.47	0.50	0.54
MTSMFF(Ours)	**23.52**	**33.06**	**41.09**	**47.12**	**48.05**	**46.15**	**0.26**	**0.37**	**0.42**	**0.45**	**0.50**	**0.52**

Note: The multi-step prediction size is 6, and the main parameters of deep learning models are set as epochs-100, batch size-96, dropout-0.3, optimizer-Adagrad, loss function-MSE. SVR parameters are set as: C-2.0, epsilon-0.1, gamma-0.5.

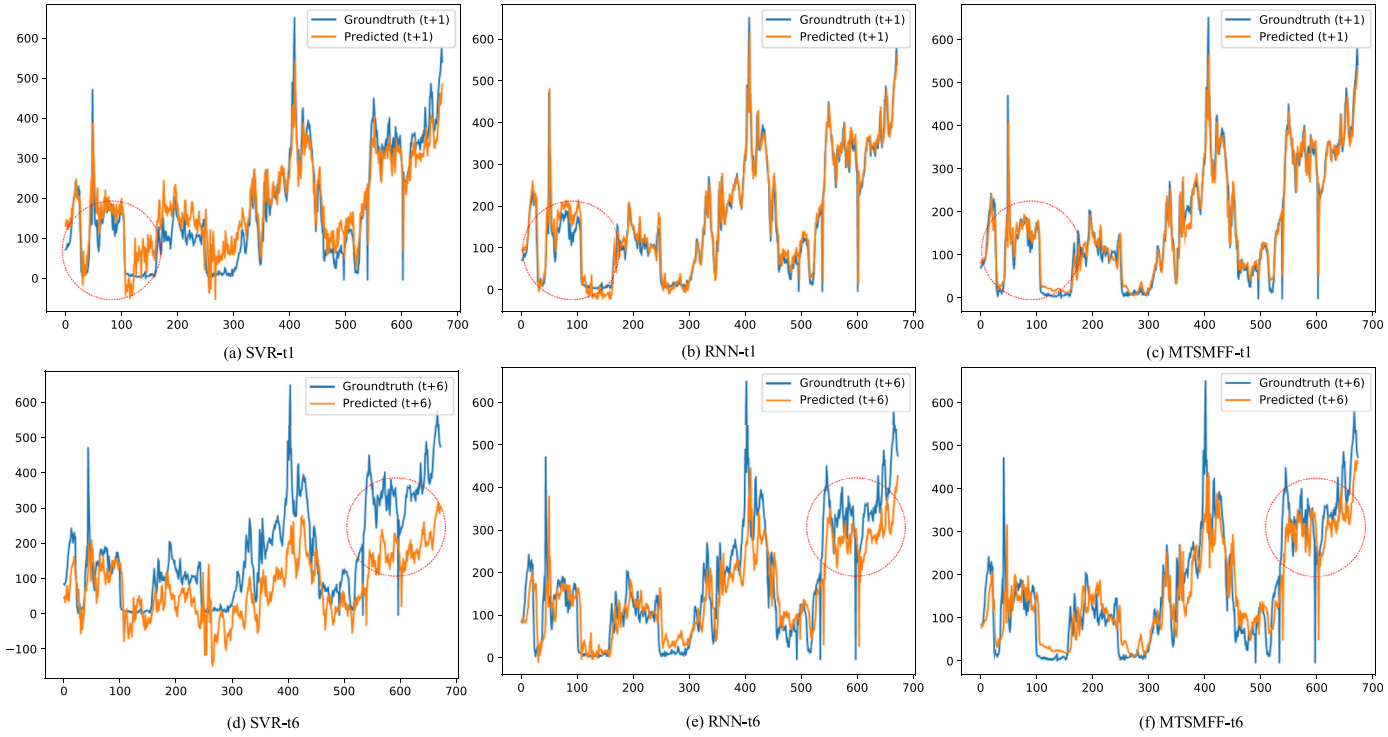

Fig. 4. In the experiment on the Beijing PM2.5 dataset, a comparison of the ground truth value and the multi-step forward predicted value of different methods under different prediction sizes (one timestep vs. 6 timesteps) conditions. (a) SVR model for one-timestep forward prediction; (b) RNN model for one-timestep forward prediction; (c) our model MTSMFF for one-timestep forward prediction; (d) SVR model for 6-timestep forward prediction; (e) RNN model for 6-timestep forward prediction; (f) our model MTSMFF for 6-timestep forward prediction.

(SEQ2SEQ-BI and SEQ2SEQ-ATT). Furthermore, the prediction ability of our model can maintain stable on different data sets.

We also discover a common phenomenon through experiments: When the prediction time step increases, the prediction ability of all models decreases gradually, which shows that the cumulative error of the multi-step forecasting model is increasing (except for SVR-LINEAR, which prediction performance is unstable). This also reflects the real problems faced by multi-step time series forecasting models. The longer the time step in the future, the more difficult it is to predict. Moreover, with the growth of prediction time step, our model has a greater performance advantage than the baseline models in Beijing PM2.5 dataset experiments. The experimental results show that our model can maintain the lowest prediction error compared with baseline methods under the condition of short-term and long-term time-step prediction.

For the comparison of the prediction performance of the deep learning methods and the shallow learning methods, we find an interesting phenomenon, that is, for short-term time-step forecasting of multivariate time series task, the baseline deep learning method has no obvious advantage (even get lower performance) compared with the performance of the shallow learning method. For example, as shown in Table 4, in the case of single-step prediction (*t*1), the prediction error of VARMA is smaller than that of LSTM and SEQ2SEQ models. However, in the case of long-term time-step prediction (e.g. from *t*4 to *t*6), we find that the forecasting performance of the baseline deep learning method exceeds the shallow learning method. But the proposed method

Fig. 5. A comparison of the ground truth value and six time-step ($t6$) predicted power consumption value of different methods during 24 h on individual household electric power consumption dataset, and the error is the absolute value of the predicted value minus the ground-truth value. (a) SVR-RBF model for six-timestep forward prediction; (b) RNN model for six-timestep forward prediction; (c) our model MTSMFF for six-timestep forward prediction.

can avoid this problem, and the experimental data shows that the prediction ability of MTSMFF is better than the baseline methods under different time-step forecasting conditions.

Moreover, the experimental results show that the multi-step forecasting performance of multivariable time series based on SVR-LINEAR models is unstable. As Table 4 shows, in the Power Consumption dataset experiment, the forecasting performance of SVR-LINEAR has large fluctuations, e.g. the forecasting performance of the first time-step is very good, while the forecasting performance of the second time-step is very poor. Compared with the shallow learning models, deep learning models are more stable and there is no violent fluctuation, although the forecasting performance is gradually reduced as the forward prediction time-step increases.

We further analyze the multi-step prediction performance of the proposed model through a visual graphical display on the Beijing PM2.5 dataset. Fig. 4(a)–(f) shows the ground-truth value and predicted value curves of different models (SVR, RNN, and our model MTSMFF) at different prediction time steps (one time-step vs. six time-step). As shown in the deviation between the predicted value and the ground-truth value in these figures, in the case of single-step forecasting, the differences between three models are not too large. We can see more obvious deviations under the multi-step forecasting condition. Our model has the best prediction ability under both single-step and multi-step forecasting conditions.

In addition, we also analyze the multi-step prediction performance of the proposed model through a visual graphical display on the Power Consumption dataset experiment over the course of 24 h (1440 time-step points). Fig. 5(a)–(c) shows the ground-truth value and the predicted power consumption value curves of different models (SVR, RNN, and our model MTSMFF) at six time-step prediction conditions ($t6$), where x-coordinate indicates the observation time-step, and the error is the absolute value of the predicted value minus the ground-truth value. As shown in the deviation (error line) between the predicted value and the ground-truth value in these figures, we can see more obvious deviations of baseline models under the multi-step forecasting condition, and the prediction ability of our model is the best. In summary, the experimental results prove that under different conditions, the multi-step forecasting performance of the proposed model is more stable and effective than the other baseline models.

Finally, we compare the forecasting performance of different models on univariate input conditions and multivariate time series input conditions. Taking Beijing PM25 dataset for example, in

Table 5
Test error (RMSE) comparison between our model MTSMFF and other baseline models on univariate time series input condition and multivariate time series input condition.

Models	Beijing PM25 *dataset*		PeMS Bay *dataset*	
	Unvariate	Multivariate	Unvariate	Multivariate
ARIMA	53.41	–	5.44	–
VARMA	–	51.78	–	5.42
SVR-LIN	85.42	88.35	5.65	6.51
SVR-RBF	53.80	57.64	5.11	5.59
RNN	45.09	44.42	4.12	3.95
CNN	43.15	42.98	4.05	3.86
LSTM	44.63	43.12	3.94	3.87
GRU	44.62	43.04	3.92	3.84
SEQ2SEQ	44.56	42.53	3.97	3.85
SEQ2SEQ-BI	43.27	42.29	3.86	3.77
SEQ2SEQ-ATT	43.14	42.15	3.82	3.76
MTSMFF	43.09	39.83	3.28	3.09

the case of univariate input condition, only PM2.5 single variable sequence is input to the model; in the case of multivariate time series input condition, PM2.5 variable itself and the other variables (e.g. humidity and wind speed) are input the model together. As shown in Table 5, for the shallow learning model, the forecasting ability under univariate conditions is better than that under multivariate conditions, which indicates that it is difficult for the shallow learning model to effectively learn the hidden non-linear correlation features of multivariate time series data. However, for the baseline deep learning models and our model, the forecasting performance under the multivariate time series input condition is improved compared with the forecasting performance under univariate input condition, which indicates that the deep learning model can learn hidden non-linear correlation features of multivariate time series data to some extent.

All in all, through the comparative analysis of the real multivariate time series dataset experiments, which validates that our model has the best forecasting performance compared with other baseline methods. The reason that our model is more effective is because it can learn deep temporal representations and long-term temporal dependency features of multivariate time series data, which also can extract the interdependent correlation features of multiple variables of temporal data by using the temporal attention mechanism. The model of this paper can provide a valuable reference for the application of intelligent prediction system based on multivariate time series data.

5. Conclusion and future work

In this paper, we proposed an end-to-end deep learning framework for multivariate time series forecasting, which leverages the idea of the encoder–decoder learning structure with Bi-LSTM and is augmented with a temporal attention mechanism. It is a firstly proposed representation method of dynamic multivariate time series data, which can jointly learn the long-term temporal dependencies pattern and non-linear correlation features of multivariate temporal data. Experiments on five multivariate time series datasets showed that the proposed model can effectively forecast multi-step time series value, no matter under short-term time-step condition or long-term time-step conditions. In addition, due to the performance of deep learning models that are usually related to different parameters and dataset conditions in many cases, how to improve and stabilize the forecasting ability of our model for better supporting online time series forecasting and early warning applications is the focus of future research.

Declaration of Competing Interest

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

CRediT authorship contribution statement

Shengdong Du: Methodology, Investigation, Software, Writing - original draft. **Tianrui Li:** Supervision, Conceptualization, Resources, Writing - review & editing, Project administration. **Yan Yang:** Writing - review & editing. **Shi-Jinn Horng:** Supervision, Writing - review & editing.

Acknowledgments

This research was partially supported by the National Natural Science Foundation of China (Nos. 61773324 and 61976247), the “Center for Cyber-physical System Innovation” from The Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education (MOE) in Taiwan and MOST under 106-2221-E-011-149-MY2 and 108-2218-E-011-006.

References

[1] J.G. De Gooijer, R.J. Hyndman, 25 years of time series forecasting, Int. J. Forecast. 22 (3) (2006) 443–473.
[2] D. Xia, B. Wang, H. Li, et al., A distributed spatial–temporal weighted model on MapReduce for short-term traffic flow forecasting, Neurocomputing 179 (2016) 246–263.
[3] Qi Z., Wang T., Song G., et al. Deep air learning: Interpolation, prediction, and feature analysis of fine-grained air quality. IEEE Trans. Knowl. Data Eng., doi:10.1109/TKDE.2018.2823740, 2018
[4] S. Ahmad, A. Lavin, S. Purdy, et al., Unsupervised real-time anomaly detection for streaming data, Neurocomputing 262 (2017) 134–147.
[5] L. Guo, N. Li, F. Jia, Y. Lei, J. Lin, et al., A recurrent neural network based health indicator for remaining useful life prediction of bearings, Neurocomputing 240 (2017) 98–109.
[6] G.E.P. Box, D.A. Pierce, Distribution of residual autocorrelations in autoregressive-integrated moving average time series models, J. Am. Stat. Assoc. 65 (332) (1970) 1509–1526.
[7] P.F. Pai, K.P. Lin, C.S. Lin, et al., Time series forecasting by a seasonal support vector regression model, Exp. Syst. Appl. 37 (6) (2010) 4261–4265.
[8] S.H. Park, J.H. Lee, J.W. Song, et al., Forecasting change directions for financial time series using hidden Markov model, in: Proceedings of the International Conference on Rough Sets and Knowledge Technology, Berlin, Heidelberg, Springer, 2009, pp. 184–191.
[9] G.P. Zhang, Time series forecasting using a hybrid ARIMA and neural network model, Neurocomputing 50 (2003) 159–175.
[10] T. Fu, A review on time series data mining, Eng. Appl. Artif. Intell. 24 (1) (2011) 164–181.
[11] G.P. Zhang, M. Qi, Neural network forecasting for seasonal and trend time series, Eur. J. Oper. Res. 160 (2) (2005) 501–514.
[12] N.I. Sapankevych, R. Sankar, Time series prediction using support vector machines: a survey, IEEE Comput. Intell. Mag. 4 (2) (2009) 24–38.
[13] B.S. Freeman, G. Taylor, B. Gharabaghi, et al., Forecasting air quality time series using deep learning, J. Air Waste Manag. Assoc. 68 (8) (2018) 866–886.
[14] Y. Lv, Y. Duan, W. Kang, et al., Traffic flow prediction with big data: a deep learning approach, IEEE Trans. Intell. Transp. Syst. 16 (2) (2015) 865–873.
[15] J. Zhang, Y. Zheng, D. Qi, Deep spatio-temporal residual networks for citywide crowd flows prediction, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2017, pp. 1655–1661.
[16] A. Krizhevsky, I. Sutskever, E. Hinton G, Imagenet classification with deep convolutional neural networks, in: Proceedings of the International Conference on Neural Information Processing Systems, 2012, pp. 1097–1105.
[17] J. Schmidhuber, Deep learning in neural networks: an overview, Neural Netw. 61 (2015) 85–117.
[18] A. Karpathy, F.F. Li, Deep visual-semantic alignments for generating image descriptions, in: Proceedings of the International Conference on Computer Vision and Pattern Recognition, 2015, pp. 3128–3137.
[19] S. Venugopalan, M. Rohrbach, J. Donahue, et al., Sequence to sequence-video to text, in: Proceedings of the International Conference on on Computer Vision and Pattern Recognition, 2015, pp. 4534–4542.
[20] K. Cho, B. Van Merriënboer, C. Gulcehre, et al., Learning phrase representations using rnn encoder–decoder for statistical machine translation, in: Proceedings of the International Conference on Empirical Methods in Natural Language Processing, 2014, pp. 1724–1734.

[21] Gamboa J. C. B. Deep learning for time-series analysis. arXiv preprint arXiv:1701.01887, 2017.
[22] T. Kuremoto, S. Kimura, K. Kobayashi, et al., Time series forecasting using a deep belief network with restricted Boltzmann machines, Neurocomputing 137 (2014) 47–56.
[23] Y. Tian, K. Zhang, J. Li, et al., LSTM-based traffic flow prediction with missing data, Neurocomputing 318 (2018) 297–305.
[24] Y. Bao, T. Xiong, Z. Hu, Multi-step-ahead time series prediction using multiple-output support vector regression, Neurocomputing 129 (2014) 482–493.
[25] N.K. Ahmed, A.F. Atiya, N.E. Gayar, et al., An empirical comparison of machine learning models for time series forecasting, Econom. Rev. 29 (5–6) (2010) 594–621.
[26] M.R. Hassan, B. Nath, M. Kirley, A fusion model of HMM, ANN and GA for stock market forecasting, Exp. Syst. Appl. 33 (1) (2007) 171–180.
[27] S.B. Taieb, G. Bontempi, A.F. Atiya, et al., A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition, Exp. Syst. Appl. 39 (8) (2012) 7067–7083.
[28] G.E.P. Box, G.M. Jenkins, G.C. Reinsel, et al., Time Series Analysis: Forecasting and Control, John Wiley & Sons, 2015.
[29] W. Liu, Z. Wang, X. Liu, et al., A survey of deep neural network architectures and their applications, Neurocomputing 234 (2017) 11–26.
[30] Chambon S., Galtier M. N., Arnal P. J., et al. A Deep Learning Architecture for Temporal Sleep Stage Classification Using Multivariate and Multimodal Time Series. arXiv:1707.03321 (2017).
[31] N. Laptev, J. Yosinski, L.E. Li, et al., Time-series extreme event forecasting with neural networks at uber, in: Proceedings of the International Conference on Machine Learning, 34, 2017, pp. 1–5.
[32] J. Yang, M.N. Nguyen, P.P. San, et al., Deep convolutional neural networks on multichannel time series for human activity recognition, in: Proceedings of the International Joint Conferences on Artificial Intelligence, 15, 2015, pp. 3995–4001.
[33] X. Ding, Y. Zhang, T. Liu, et al., Deep learning for event-driven stock prediction, in: Proceedings of the International Joint Conferences on Artificial Intelligence, 2015, pp. 2327–2333.
[34] Y. Zheng, Q. Liu, E. Chen, et al., Time series classification using multi-channels deep convolutional neural networks, in: Proceedings of the International Conference on Web-Age Information Management, Cham, Springer, 2014, pp. 298–310.
[35] S. Yao, S. Hu, Y. Zhao, et al., Deepsense: a unified deep learning framework for time-series mobile sensing data processing, in: Proceedings of the 26th International Conference on World Wide Web, 2017, pp. 351–360.
[36] N. Jaitly, Q.V. Le, O. Vinyals, et al., An online sequence-to-sequence model using partial conditioning, in: Proceedings of the International Conference on Neural Information Processing Systems, 2016, pp. 5067–5075.
[37] J.K. Chorowski, D. Bahdanau, D. Serdyuk, et al., Attention-based models for speech recognition, in: Proceedings of the International Conference on Neural Information Processing Systems, 2015, pp. 577–585.
[38] I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks, in: Proceedings of the Advances in Neural Information Processing Systems, 2014, pp. 3104–3112.
[39] Kuznetsov V., Mariet Z. Foundations of Sequence-to-Sequence Modeling for Time Series. arXiv preprint arXiv:1805.03714, 2018.
[40] UCI Machine Learning Repository, 2017 [Online]Available: http://archive.ics.uci.edu/ml/index.php.
[41] J. Du Preez, S.F. Witt, Univariate versus multivariate time series forecasting: an application to international tourism demand, Int. J. Forecast. 19 (3) (2003) 435–451.
[42] D.E.H. Zhuang, G.C.L. Li, A.K.C. Wong, Discovery of temporal associations in multivariate time series, IEEE Trans. Knowl. Data Eng. 26 (12) (2014) 2969–2982.
[43] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[44] Bahdanau D., Cho K., Bengio Y. Neural Machine Translation by Jointly Learning to Align and Translate. arXiv preprint arXiv:1409.0473, 2014.
[45] Highway England Traffic Data Set, 2013 [Online] Available: http://data.gov.uk/dataset/highways-england-network-journey-time-and-traffic-flow-data
[46] PeMS Traffic Flow Data Set, 2017 [Online] Available: http://pems.dot.ca.gov/.
[47] J. Duchi, E. Hazan, Y. Singer, Adaptive subgradient methods for online learning and stochastic optimization, J. Mach. Learn. Res. 12 (2011) 2121–2159.
[48] Malhotra P., Ramakrishnan A., Anand G., et al. LSTM-Based Encoder–Decoder for Multi-Sensor Anomaly Detection. arXiv preprint arXiv:1607.00148, 2016.
[49] S.H Park, B.D. Kim, C.M. Kang, et al., Sequence-to-sequence prediction of vehicle trajectory via LSTM encoder-decoder architecture, in: Proceedings of the IEEE Intelligent Vehicles Symposium, 2018, pp. 1672–1678.

Shengdong Du received the B.S. and M.S. degrees in Computer Science from Chongqing University in 2004 and 2007, respectively. He is currently a Ph.D. candidate in the School of Information Science and Technology, Southwest Jiaotong University. His research interests include data mining and machine learning.

Tianrui Li received the B.S., M.S., and Ph.D. degrees from the Southwest Jiaotong University, Chengdu, China in 1992, 1995, and 2002, respectively. He was a postdoctoral researcher with SCK•CEN, Belgium from 2005 to 2006, and a visiting professor with Hasselt University, Belgium, in 2008, the University of Technology, Sydney, Australia, in 2009, and the University of Regina, Canada, 2014. He is currently a Professor and the Director of the Key Laboratory of Cloud Computing and Intelligent Techniques, Southwest Jiaotong University. He has authored or coauthored more than 300 research papers in refereed journals and conferences. His research interests include big data, cloud computing, data mining, granular computing and rough sets. He is a fellow of IRSS and senior member of ACM and IEEE.

Yan Yang received the B.S. and M.S. degrees from Huazhong University of Science and Technology, Wuhan, China, in 1984 and 1987, respectively. She received Ph.D. degree from Southwest Jiaotong University, Chengdu, China in 2007. From 2002 to 2003 and 2004 to 2005, she was a visiting scholar with the University of Waterloo, Canada. She is currently a Professor and Vice Dean with the School of Information Science and Technology, Southwest Jiaotong University, Chengdu, China. She is an Academic and Technical Leader of Sichuan Province. Her research interests include artificial intelligence, big data analysis and mining, ensemble learning, cloud computing and service. She has participated in more than 10 high-level projects recently, authored and co-authored over 150 papers in journals and international conference proceedings. She also serves as the Vice Chair of ACM Chengdu Chapter, a distinguished member of CCF, a senior member of CAAI, and a member of IEEE and ACM.

Shi-Jinn Horng received the B.S. degree in electronics engineering from National Taiwan Institute of Technology, the M.S. degree in information engineering from National Central University, and the Ph.D. degree from National Tsing Hua University, in 1980, 1984, and 1989, respectively. He is currently a Chair Professor in the Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology. He has published more than 200 research papers and received many awards. Especially, the Distinguished Research Award got from the National Science Council in Taiwan in 2004. His research interests include deep learning, biometric recognition, image processing, and information security.