Soft Comput
DOI 10.1007/s00500-017-2624-5

METHODOLOGIES AND APPLICATION

A multiple time series-based recurrent neural network for short-term load forecasting

Bing Zhang[1] · Jhen-Long Wu[2] · Pei-Chann Chang[3,4,5]

Abstract Electricity, an indispensable resource in daily life and industrial production, is hard to store, so accurate short-term load forecasting (STLF) plays a vital role in resource allocation, capital budgeting of power companies, energy deployment and government control. In recent decades, the strong dependency relationships of time series have been considered in many researches, but the discrete information has not proven to be very useful in their experiments. In general, while discrete information is weak, it can provide macro trends compared to the micro trends of continuous information. In this research, we aim to combine macro and micro information by continuous and discrete time series to generate multiple time series (MTS). The MTS comprise four information sequences: short-term, cycle, long short-term and cross-long short-term. These MTS are used to build a STLF system using a recurrent neural network (RNN) model that can learn sequential information between continuous and discrete series. Therefore, the RNN model with MTS information can improve the forecasting performance for short-term load forecasting. The experimental results show that our proposed forecasting system outperforms the state-of-the-art approach.

Keywords Multiple time series · Deep learning · Short-term load forecasting · Recurrent neural networks · Long short-term memory · Gated recurrent units

Communicated by V. Loia.

✉ Pei-Chann Chang
iepchang@saturn.yzu.edu.tw

1 School of Information Science & Engineering, Yun-Nan University, Yun Zan, China

2 Institute of Information Science, Academia Sinica, Taipei, Taiwan, ROC

3 Innovation Center for Big Data and Digital Convergence, Department of Information Management, Yuan Ze University, Taoyuan, Taiwan, ROC

4 Software School, Nanchang University, Nanchang, China

5 Office of Academic Affairs, Zhuhai College of Beijing Institute of Technology, Zhuhai 519088, Guangdong, China

1 Introduction

Load forecasting has remained an important research issue for several decades. It is a fundamental demand for the control and planning of power systems. Electricity is a special form of energy that is hard to store, so short-term load forecasting (STLF) at the day-ahead timescale is an essential aspect of power system operations in the unit commitment process (Hodge et al. 2013). Load forecasting includes three kinds of methods: conventional mathematical (Badar 2011), advanced machine learning (Mateo et al. 2013) and deep learning (Dedinec et al. 2016). Load forecasting is a difficult task because load consumption at a present time and previous times has a very complex correlation. Past researches focused on continuous time series information during a set period, but the discrete time series of a cycle were not considered because the impact of forecasting was not considered to be significant. However, the discrete time series offer some valuable information that can provide the macro trend, which affects forecasting based on continuous time series information (Deng and Jirutitijaroen 2010). Continuous and discrete time series data generation can provide overall information for realizing improved forecasting performance.

Deep learning technology has proven very popular in regard to solving many regression and classification problems in recent years. It can be used in automatic learning and to retrieve desired knowledge from huge amounts of training

Published online: 04 May 2017

data. The key aspect of deep learning is that the resultant layers of features are not designed by human engineers, but rather are learned from data using a general-purpose learning procedure (LeCun et al. 2015). In particular, the recurrent neural network (RNN) is a very powerful dynamic system and an important implementation mechanism of deep learning. It can process sequential data and multiple time steps, fulfilling requirements such as speech, language and time series forecasting. RNN processes an input sequence one element at a time, maintaining hidden units in their state vector that implicitly contain information about the history of all the past elements of the sequence. A special unit, called the memory cell, acts like an accumulator or a gated leaky neuron. This unit has a connection to itself at the next time step that has a weight of one, so it copies its own real-value state and accumulates external signals. This self-connection is multiplicatively gated by another unit that learns to decide when to clear the content of the memory. The RNN method can find the dependencies relationship of time series that provide more effective ways for time memory to operate. Loop memory can extract the valuable information from the history data through memory cell execution and other control mechanisms. The long short-term memory (LSTM) and gated recurrent units (GRUs) are two kinds of special memory cell of RNN that employ different memory cell mechanisms. LSTM and GRU networks use special hidden units whose natural function is remembering inputs for a long time (Hochreiter and Schmidhuber 1997a). However, regarding the load of power data with obvious characteristics of time series and cycles, the load forecasting can take advantage of history information via the LSTM and the GRU cell. The conventional machine learning methods, utilizing the mechanism of end-to-end to input and output, cannot effectively tap the history information and generate further processing from the sequential data

In this paper, we propose a novel short-term load forecasting system based on multiple time series (MTS) with an RNN model; the system has two main stages: In the first stage, MTS sequence datasets that have short-term, cycle, long short-term and cross-long short-term features are generated. In the second stage, we built the load forecasting system using the RNN model with deep learning methods. We have experimented in different ways to combine multiple time series dataset. The main contributions of this paper are as follows:

(1) We propose the novel concept of generating time series datasets with multiple time series datasets possessing two features: continuous and discrete sequences. The four different time series datasets: short-term, cycle, long short-term and cross-long short-term series can provide enough time series information for accurate load forecasting.
(2) The recurrent neural network model is used to handle the MTS datasets and learn the load forecasting system because the RNN can remember whether sequence information is continuous or discrete.
(3) The MTS datasets in the RNN model are used to build an STLF system that has effective learning and accurate forecasting, thereby providing improved performance.

2 Related work

Time series is one of the most important and often considered mechanisms in load forecasting. During the past few decades, numerous efficient algorithms and techniques for the short-term load forecasting problem were presented. We have reviewed and will discuss the conventional mathematical approach, advanced machine learning methods and deep learning methods, respectively.

2.1 Conventional mathematical approach

Conventional mathematical techniques based on time series for short-term load forecasting have produced many important and effective research results. For example, Nataraja et al. (2012) used traditional time series analysis for Karnataka Demand and hence comparisons of different models, such as the autoregressive (AR) model, autoregressive moving average (ARMA) model and autoregressive integrated moving average (ARIMA) model. Bahrami et al. (2014) proposed a novel model based on the combination of the wavelet transform (WT) and Grey model (GM) for the STLF; in their research, the weather data are considered as the model inputs and include average temperature, average relative humidity, average wind speed and day-ahead load information. The wavelet transform is used to eliminate the high frequency components of the day-ahead load data, and the generation coefficient of GM is enhanced using the PSO algorithm; this model is used for New York's and Iran's load forecasting, and shows good performance. Other important works for several mathematical methods based on time series, and other important features were developed; for example, Taylor (2012) considered five developed exponentially weighted methods; they include several exponential smoothing formulations using discount weighted regression, cubic splines and singular value decomposition (SVD). In addition, many methods based on artificial immune system (AIS) for the STLF have surfaced in recent years. For example, Dudek (2008) first proposed a novel model based on an artificial immune system (AIS) to solve the problem of short-term load forecasting: an artificial immune system is trained to recognize antigens that encode sequences of load time series and generate immune memory. In the testing stage, the sequence is reconstructed from activated antibodies; their model shows good performance. Furthermore, Dudek (2011) focused on time series with multiple seasonal periods, using a new immune inspired method; a model was proposed for the STLF that generates immune memory cells with multiple seasonal series for computation and application to forecast construction. Recently, the focus has been on multiple seasonal cycles with time series. Dudek (2017) proposed a novel load forecasting model based on the artificial immune system. This approach takes the time series with multiple seasonal cycles into account as the model is used for STLF; the results exhibited good performance compared with other artificial immune system-based forecasting models as well as neural networks, ARIMA and exponential smoothing. While all of the above methods analyze the relationship between time series and multiple seasonal cycles, which can extract some useful features from the time series, they lack a more effective mechanism to find the stronger time dependence information between short term and long term.

2.2 Advanced machine learning approach

The machine learning approach is a crucial aspect of artificial intelligence; it has been comprehensively applied in many fields. Machine learning technology has significantly improved the performance of STLF; in recent research, many efficient models were developed for STLF. For example, Shrivastava and Bhandakkar (2013) proposed a neural network approach for forecasting the load profile with a lead time of one to seven days from the Load Dispatch Centre Jabalpur. They took three types of neural network techniques into consideration: radial basis function neural network (RBFNN), feed forward neural network (FFNN) and cascade-forward neural network (CFNN); the performance exhibited significant improvement. Niu et al. (2012) presented a model based on Bayesian neural network (BNN) learned by the hybrid Monte Carlo (HMC) algorithm for forecasting the hourly load of 25 days of April (spring), August (summer), October (autumn) and January (winter), respectively. Their experimental results illustrate that the BNN learned by the HMC algorithm offers far better performance than the BNN learned by the Laplace algorithm; the neural network learned by the BP algorithm and the BNN learned by the HMC have powerful generalizing capability and can solve the overfitting problem well. However, some important approaches, such as data mining and big data technologies, exist for STLF as well (Shelke and Thakare 2014; Zhang et al. 2015). Besides, many combination methods have been presented to improve the performance and accuracy of load forecasting (Nie et al. 2012; Che and Wang 2014; Hu et al. 2014; Kouhi et al. 2014; Chaturvedi et al. 2015; Ghofrani et al. 2015; Ghayekhloo et al. 2015). Many other recent important works still deserve notice; for example, Dudek (2016) proposed a univariate model for STLF based on linear regression and patterns of daily cycles of load time series; the relationship between the patterns was modeled locally in the neighborhood of the current input using linear regression to analyze the holiday load consumptions and normal days. Çevik and Çunkaş (2016) presented different models using the fuzzy logic method without weather information. In their methods, holidays were classified according to their characteristics and historical load shapes; each fuzzy model had three inputs and one output. While historical data from many years, consumption data from last week and the type of holiday were all selected as inputs, the output was the hourly predicted holiday load. Hu et al. 2015 considered both filter and wrapper methods, first using the partial mutual information-based filter method to filter out most of the irrelevant and redundant features, and subsequently applying a wrapper method, implemented via a firefly algorithm; finally, they selected support vector regression (SVR) to implement the hybrid feature selection scheme. Hu et al. (2014) proposed a kernel-based SVR combination model by using a novel individual model selection algorithm; the combination model provided a new way to kernel function selection in the SVR model. The performance and electric load forecast error of the proposed model were assessed by means of real data from the Australia and California Power Grid, respectively; the results showed that the combination model decreased the electric load forecasting errors compared to the best individual kernel-based SVR model. Moreover, Cheepati and Prasad (2016) carried out a comparative study of these forecasting models; for example, they found that a combination of curve fitting, regression trees and neural network leads to less error by MAPE. All of the above methods considered the history load of power systems; however, other models take the weather, humidity and other features into consideration (Medeiros and Soares 2006; Oliveira et al. 2011). Besides, some advanced technology for long-term load forecasting was proposed as well. Hong et al. (2014) proposed an approach that takes advantage of hourly information to create more accurate and defensible forecasts, modernizing three key elements of long-term load forecasting: predictive modeling, scenario analysis and weather normalization. However, the above machine learning methods' performance is weak in relation to high-dimensional data and few researchers analyzed the relationship between short-term and long-term load.

2.3 Deep learning approach

In recent years, deep learning technologies have achieved remarkable results in STLF. For example, Vermaak and Botha (1998) proposed a recurrent neural network (RNN) model for the STLF of a South African utility. They utilized the inherent nonlinear dynamic nature of neural networks to represent the load as the output of some dynamic system, influenced by

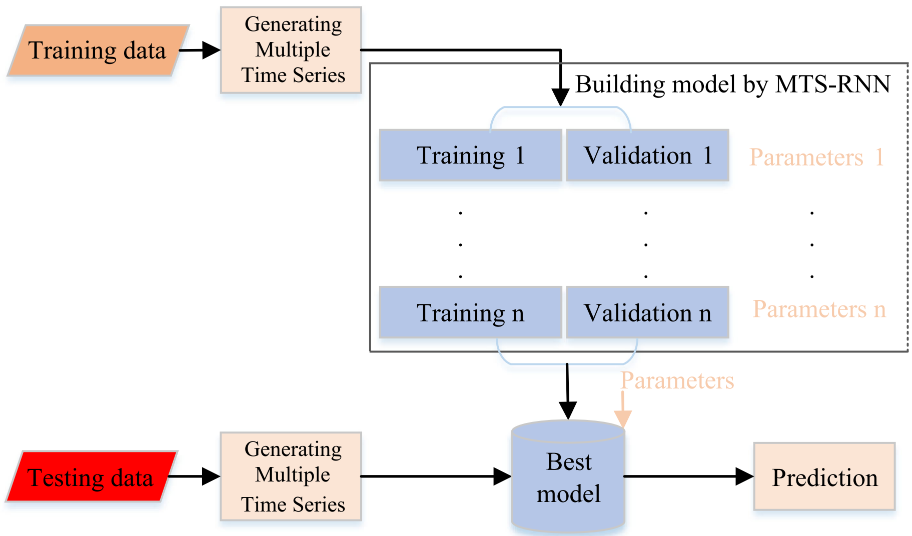

Fig. 1 The flowchart of the proposed MTS-RNN short-term load forecasting system

weather, time and environmental variables. Khan et al. (2013) utilized a neuro-evolutionary technique known as Cartesian genetic programming evolved from recurrent neural network (CGPRNN) to develop a load forecasting model for very short term, namely half an hour. The network is trained using historical data of one month on half-hourly basis to predict the next half hour load based on the 12 and 24 hourly data history. Dedinec et al. (2016) applied a deep belief network (DBN) model to short-term electricity load forecasting based on the Macedonian hourly electricity consumption data in the period 2008–2014. Hosein and Hosein (2016) compared deep neural network (DNN) methods with traditional methods for load forecasting; the results indicated that the DNN methods outperformed most traditional methods. However, the above deep learning methods have rarely considered the relationship between time jump sequence and single time series.

3 Multiple time series-based recurrent neural network (MTS-RNN)

Most time series data are considered as continuous sequence, so they lose some important information in such a cycle. In order to effectively learn and accurately forecast, we propose a multiple time series based on a recurrent neural network model called MTS-RNN. The flowchart of MTS-RNN is shown in Fig. 1. The main processes can be divided into four stages: First, we split the original data into the training and testing data. Second, the training and testing data are transformed into the MTS datasets, respectively. Third, we build the load forecasting model by RNN for short-term load forecasting; the model has four different time series dataset inputs according to the MTS generation. We take the validation data from the training data by random selection for obtaining the best forecasting model selection in the training phase. Finally, we evaluate our proposed MTS-RNN system by testing the data.

3.1 Multiple time series data generation

According to different time steps, we adopt different time sequences in our model. In this section, we propose multiple time series: short-term series, cycle series, long short-term series and cross-long short-term series. Each sequence has different sequence information. The detailed generation is described as follows:

3.1.1 Short-term series

The short-term series is the most intuitive feature of power systems, generally the characteristic of time series that can reflect the load change of the subtle trends at one stage in a time interval. They are related to the load of power in the next period; also, it is a common and effective method when we need to predict the next time load value through the history data.

Figure 2a represents the short-term series; as our target is to forecast the load value on day d hour h, we need to take "$t-n$" before the load information into account. Each time interval is 15 min, where n is a positive integer, the number before the predicted target $P(d, h)$ (in our experiment, the $n = 371$).

3.1.2 Cycle series

Cyclical analysis is a common method for determining the features of the power system; generally, the cycle characteristics that can reflect the load change of the significant trend

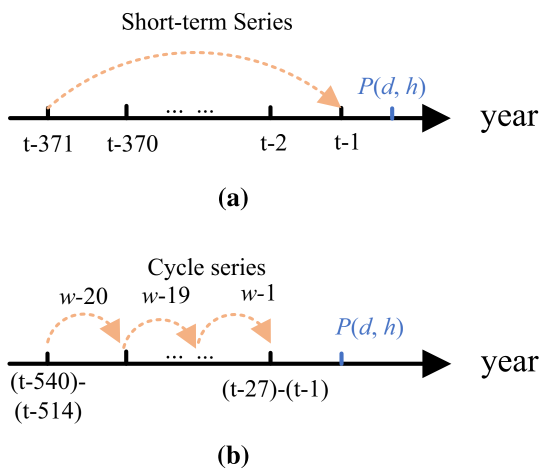

Fig. 2 The simplified views of short-term series (**a**) and cycle series (**b**) sequence

at one stage in a time interval. The features are still related to the load of power in the next period; also, it is an effective method when we need to predict the next time load value through the history information.

Figure 2b represents the cycle series; the load of power will show the significant cycle by the change of weeks and other period characteristics. According to the history load of power systems, we choose a week as the minimum cycle. We handle the load of " *week-m*" before the cycle series, where m is a positive integer (in our experiment, the $m = 20$).

In Fig. 2, each step represents the order of input sequence; the model will reference the history information in light of the different step sequences. For example, from "$t - n$" to "$t - 1$" is like the label from "$w - 20$" to "$w - 1$," signifying a period from before 20 weeks to before 1 week.

3.1.3 Long short-term series

Long short-term series is a novel and important concept in our model; similar to the short-term series and cycle series, there are different time steps and vector dimensions. As shown in Fig. 3a, the long short-term series means that we refer to all the load information of last year first (from "$w - 2$" to "d" by "year-1"). After that, it refers to all the load information of this year (from "$w - 2$" to "d" by "year") step by step, where "year-1," "year," "w-2" and "d" mean last year, this year, the same times before two weeks and today, respectively.

3.1.4 Cross-long short-term series

Cross-long short-term series are still a novel and important concept in our model; similar to the short-term series and cycle series, there are different time steps and vector dimensions. As shown in Fig. 3b, the greatest difference of long

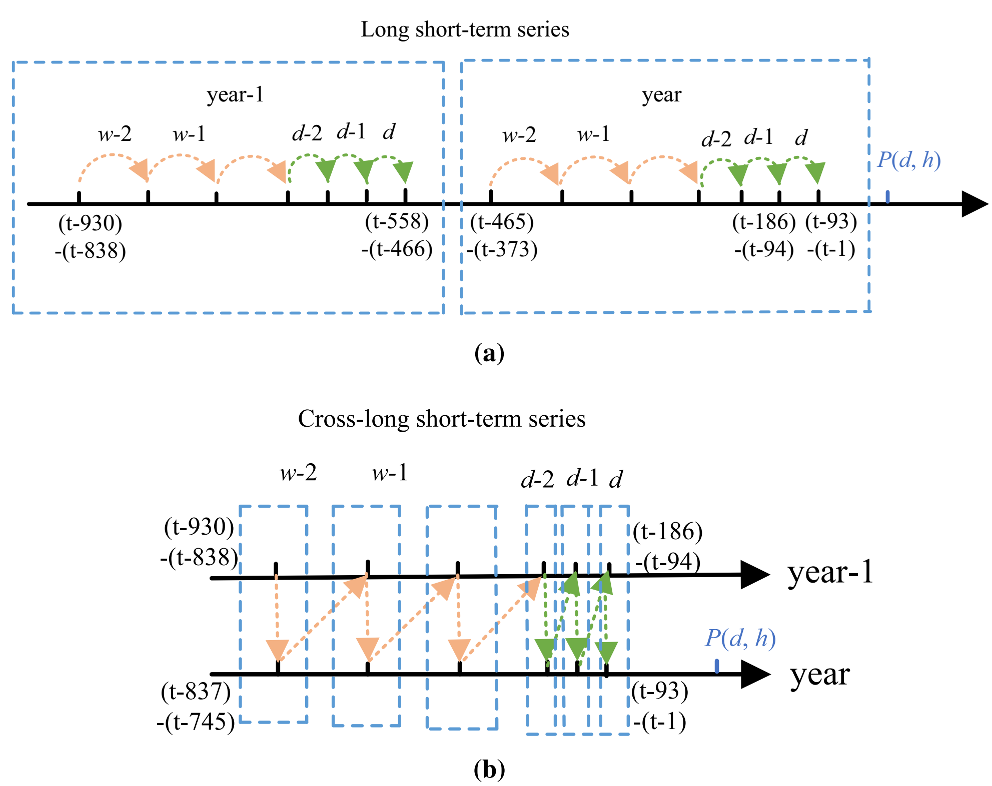

Fig. 3 The simplified views of long short-term series (**a**) and cross-long short-term series (**b**) sequence

short-term series and cross-long short-term series is just the input sequence that refers to history load data having a different order. The cross-long short-term series means that we refer to the load information of one step ("$w-2$") last year first, then to one step ("$w-2$") this year, followed by one step ("$w-1$") last year, then one step ("$w-1$") this year, and so forth, until the step before that our forecasting target $P(d,h)$, for instance, from "$w-2$" to "d" sequence in Fig. 3. Whether long short-term series or cross-long short-term series, they can exert different memory effects on load forecasting through a step-by-step sequence.

Recurrent neural networks attach importance to the features and benefits of multiple time series and time memory. Different input sequences that can memorize the useful information by the multiple time series and the time jump steps feature that combines situations may be a good choice, such as combining short-term series with cycle series, long short-term series with cross-long short-term series, and mutual combinations with them.

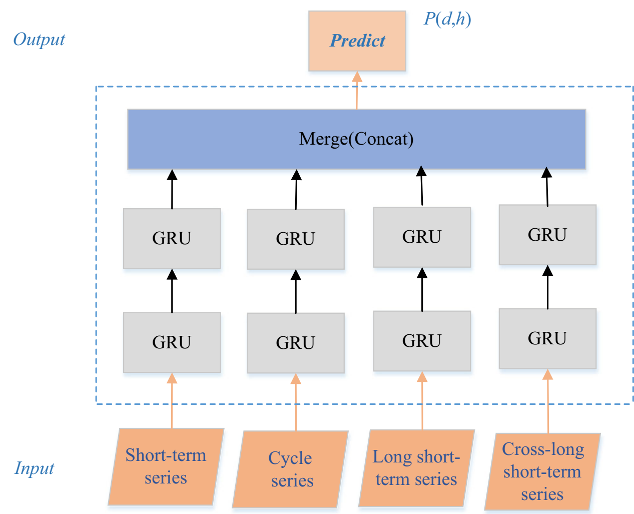

Fig. 4 The main framework of our model for STLF

3.2 Recurrent neural network model for STLF

In this section, we build the RNN model to handle and learn from MTS information for the STLF system. The model is divided into three layers: the input layer, hidden layer and output layer, respectively. We adopt four input sequence mechanisms in the input layer: the short-term series, the cycle series, the long short-term series and the cross-long short-term series. Complementarily, we use two layer units of long short-term memory (LSTM) and gated recurrent unit (GRU) in the hidden layer. We describe the LSTM and GRU unit principle and mechanism in Sects. 3.2.1 and 3.2.2 in detail; after that, we add a merged layer combination of the four sequence results by concatenation. The finally layer is the output; predictably, the output is $p(d,\ h)$ of our predicted target, scilicet, the load value of time h of day d (Fig. 4).

3.2.1 Long short-term memory (LSTM)

Hochreiter and Schmidhuber (1997b) introduced the LSTM model primarily in order to overcome the problem of vanishing gradients. An LSTM RNN is composed of one input layer, one recurrent hidden layer and one output layer. Differing from the traditional NN, the basic unit of the hidden layer is the memory block. The memory block contains memory cells with self-connections for memorizing the temporal state, and a pair of adaptive, multiplicative gating units to control information flow in the block. Two additional gates, named the input gate and output gate, respectively, control the input and output activation in the block. The core of the memory cell is a recurrently self-connected linear unit Constant Error Carousel (CEC), and the activation of the CEC represents the cell state. Due to the presence of the CEC, multiplicative gates can learn to open and close; thus, LSTM RNN can solve the vanishing error problem by remaining the network error constant. To prevent the internal cell values from growing without bounds when processing continual time series that were not previously segmented, a forget gate was added to the memory block. This treatment enables the memory blocks to reset by themselves once the information flow is out of date and to replace the CEC weight with the multiplicative forget gate activation. It has been demonstrated that the special unit structure is very effective regarding long-term sequence dependency (Zhang et al. 2015).

These equations give the full algorithm for a modern LSTM. The input gate i is computed as follows

$$i_t = \sigma(W_{\mathrm{i}}x_t + U_{\mathrm{i}}h_{t-1} + b_{\mathrm{i}}) \tag{1}$$

where the i_t is the input gate at time t. The W_i is the weight matrix of input gate for input vector. The X_t is input vector to memory cell layer at time t. The U_i is the weight matrix of input gate for hidden vector. The h_{t-1} denotes the hidden state at time $t-1$. The b_i denotes the bias vectors of input gate. The σ is the logistic sigmoid activation function. The forget gate f is computed as follows

$$f_t = \sigma\,(W_{\mathrm{f}}x_t + U_{\mathrm{f}}h_{t-1} + b_{\mathrm{f}}) \tag{2}$$

where the f_t denotes the forget gate at time t. The W_f denotes the weight matrix of forget gate for input vector. The U_f denotes the weight matrix of forget gate for hidden vector. The b_f denotes the bias vectors of forget gate. The cell state c is computed as follows

$$c_t = f_t{*}c_{t-1} + i_t{*}\phi(W_{\mathrm{c}}x_t + U_{\mathrm{c}}h_{t-1} + b_{\mathrm{c}}) \tag{3}$$

where the c_t is the cell state at time t. The W_c denotes the weight matrix of cell state for input vector. The U_c denotes the weight matrix of cell state for hidden vector. The b_f denotes the bias vectors of cell state. The * is the element-wise multiplication operation. we use the *tanh* function ϕ for the input gate. The output o is computed as following

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{4}$$

where the o_t is the output at time t. The W_o is the weight matrix of output for input vector. The U_o is the weight matrix of output for hidden vector. The b_o denotes the bias vectors of output. The hidden vector h is computed as following

$$h_t = o_t * \phi(c_t) \tag{5}$$

where h_t is the hidden state at time t.

3.2.2 Gated recurrent unit (GRU)

The gated recurrent unit (GRU) proposed by Cho et al. (2014) aims to make each recurrent unit adaptively capture dependencies of different timescales and sequences. Similar to the LSTM unit, the GRU has gating units that modulate the flow of information inside the unit, however, without separate memory cells.

GRUs are a modified version of the general RNN, mainly improved by the following two aspects: On the one hand, the sequence has different effects on different objects at the position (example, words) to the current state of the hidden layer; the further the distance, the smaller the impact. On the other hand, when an error occurs, which is caused by one or several objects from the previous sequence, it should update the weights for the corresponding objects:

$$z_t = \sigma(W_z x_t + U_z h_{t-1}) \tag{6}$$
$$r_t = \sigma(W_r x_t + U_r h_{t-1}) \tag{7}$$
$$\overline{h_t} = \phi(W_h x_t + r_t * U_h h_{t-1}) \tag{8}$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \overline{h_t} \tag{9}$$

where the z_t, r_t, $\overline{h_t}$ and h_t are the update gate, reset gate, candidate activation and hidden state at time t, respectively. The W_z and U_z are the weight matrices of update gates for input vector and hidden state, respectively. The W_r and U_r are the weight matrices of reset gates for input vector and hidden state, respectively. The * is the element-wise multiplication operation. The ϕ is the sigmoid activation function.

The update mechanism helps the GRU capture long-term dependencies. Whenever a previously detected feature or the memory content is considered to be important for later use, the update gate will be closed to carry the current memory content across multiple time steps. The reset gate mechanism helps the GRU to efficiently use the model capacity by allowing it to reset whenever the detected feature is no longer necessary.

3.3 RMSProp optimization

One of the challenging aspects of deep learning is the optimization of the training criteria over millions of parameters; it is derived from the size of the neural networks and also because the training objective is non-convex in regard to the parameters. However, RMSprop is a biased estimator proposed in neural networks for machine learning; it is also an optimization method based on gradient (like SGD). RMSProp has recently remained the method of choice for most recurrent neural networks (Dauphin et al. 2015). In this paper, we utilize the RMSProp as our parameters update optimizer, with the parameter update rules according to the following formula:

$$E[g^2]_t = \eta E[g^2]_{t-1} + (1-\eta) g_t^2 \tag{10}$$
$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \varepsilon}} g_t \tag{11}$$

where $t \in \Re$ is the iteration times. The θ is the parameters of neural network. The E is weighted sum operation. The g^2 is the vectors of gradient square. The $\eta \in (0, 1)$ is the learning rate. The ε is a smoothing term that avoids division by zero.

4 Experiment results

In this section, we illustrate the proposed multiple time series model through experiments of different sequences and analyze their performance. In the first experiment, we train the models individually for the targets of hourly load forecasting with one day in different sequences from the training data. Then, we study the combination of the models to change the different input sequences and compare it to other popular short-term load forecasting (STLF) methods, such as neural network (NN), support vector regression (SVR) and artificial immune system (AIS), via the same forecasting tasks on several load time series. Finally, we discuss concentration on the effects of different sequences for each kind of RNN.

4.1 Dataset

In the analysis, we use a dataset based on the load of the Polish power system (in Polish: Krajowy System Energetyczny or KSE). The original data consisted of 15 min sequences; temporary values (in MW) were downloaded from the PSE

Table 1 The detailed features of MTS

Input tensor	Load value	
Short-term series	$p(d-3,h-23)$, $p(d-3,h-22.75),\ldots,p(d-3,h)$,	
	$p(d-2,h-23)$, $p(d-2,h-22.75),\ldots,p(d,h-0.5)$,	
	$p(d,h-0.25)$	$(t-371)-(t-1)$
	$p(w-20,h-23)$, $p(w-20,h-22),\ldots,p(w-20,h)$,	
	$p(w-20,min)$, $p(w-20,\text{average})$, $p(w-20,\max)$	$(t-540)-(t-514)$
Cycle series	....	
	....	
	$p(w-1,h-23)$, $p(w-1,h-22),\ldots,p(w-1,h)$,	
	$p(w-1,min)$, $p(w-1,\text{average})$, $p(w-1,\max)$,	$(t-27)-(t-1)$
	$p(y-1,w-2,h-23)$, $p(y-1,w-2,h-22.75),\ldots,p(y-1,w-2,h)$	$(t-930)-(t-838)$
	$p(y-1,w-2,h-23)$, $p(y-1,w-2,h-22.75),\ldots,p(y-1,w-2,h)$	
Long short-term series	....	
	....	
	$p(y-1,d,h-23)$, $p(y-1,d,h-22.75),\ldots,p(y-1,d,h)$	$(t-558)-(t-466)$
	$p(w-2,h-23)$, $p(w-2,h-22.75),\ldots,p(w-2,h)$	$(t-465)-(t-373)$
	....	
	....	
	$p(d,h-23)$, $p(d,h-22.75),\ldots,p(d,h-0.25)$, $p(d,\text{average})$	$(t-93)-(t-1)$
	$p(y-1,w-2,h-23)$, $p(y-1,w-2,h-22.75),\ldots,p(y-1,w-2,h)$	$(t-930)-(t-838)$
	$p(w-2,h-23)$, $p(w-2,h-22.75),\ldots,p(w-2,h)$	$(t-837)-(t-745)$
Cross-long short-term series	$p(y-1,w-1,h-23)$, $p(y-1,w-1,h-22.75),\ldots,p(y-1,w-1,h)$	
	$p(w-1,h-23)$, $p(w-1,h-22.75),\ldots,p(w-1,h)$	
	....	
	....	
	$p(y-1,d,h-23)$, $p(y-1,d,h-22.75),\ldots,p(y-1,d,h)$	$(t-186)-(t-94)$
	$p(d,h-23)$, $p(d,h-22.75),\ldots,p(d,h-0.25)$, $p(d,\text{average})$	$(t-93)-(t-1)$

SA Web site.[1] The aim was to forecast the hourly load at hour $t = 1, 6, 12, 18, 24$ for the next day. The time series was divided into training data and testing data. The training data included the period February 1, 2003, to December 31, 2003, for a total of 1670 samples. The testing data included January 2004 (except for January 1) and July 2004, for a total of 305 samples. The features of different input sequences are presented in Table 1 in detail:

The $p(d, h)$ represents the load value of day d hour h; "$t-i$" is the target time before i quarter of an hour; "$w-j$" and "$y-1$" refer to before j week and last year's load information; "$h-k$," "min," "max" and "average" refer to k hour before h hour; minimum, maximum and average load value of a day, respectively.

4.2 Evaluation

Error measurement statistics plays an important role in analyzing forecasting performance, observing exceptions and benchmark methods. In this paper, we consider three kinds of error analysis methods for evaluating our model prediction results with actual values. We first use the mean absolute percentage error (MAPE) which measures the size of the error in percentage terms and is a common means of estimating error in forecasting problems. MAE and MSLE are defined as follows:

$$\text{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_i - x_i}{x_i}\right|^{*} 100 \tag{12}$$

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - x_i| \tag{13}$$

$$\text{MSLE} = \frac{1}{N}\sum_{i=1}^{N} y \left(\log\frac{y_i+1}{x_i+1}\right)^2 \tag{14}$$

where the N is the total number of predicted values. The x_i is the actual value, and y_i is the forecasting value derived by the model.

[1] http://www.pse.pl/index.php?dzid=78.

Table 2 Performance of the short-term series dataset

Method	January		July		Average	
	$MAPE_{val}$	$MAPE_{test}$	$MAPE_{val}$	$MAPE_{test}$	$MAPE_{val}$	$MAPE_{test}$
SimpleRNN	0.94	0.62	0.79	0.86	0.86	0.74
LSTM	0.80	0.67	0.76	0.80	0.78	0.73
GRU	0.79	0.65	0.78	0.83	0.78	0.74

Table 3 Performance of the cycle series dataset

Method	January		July		Average	
	$MAPE_{val}$	$MAPE_{test}$	$MAPE_{val}$	$MAPE_{test}$	$MAPE_{val}$	$MAPE_{test}$
SimpleRNN	5.10	5.59	5.13	1.70	5.11	3.64
LSTM	5.27	5.87	3.03	2.32	4.14	4.09
GRU	4.30	5.27	1.59	1.88	2.94	3.57

4.3 Results on single time series

In this section, we show all the results on different input sequences individually. The results of short-term series, cycle series, long short-term series and cross-long short-term series sequence are shown in Sects. 4.3.1–4.3.4, respectively; we tested the MAPE of January and July in 2004 by three methods.

4.3.1 Results on short-term series dataset

The results of the short-term series are shown in Table 2. We compared the results among them by the three methods in our model architecture; the best results of validation and testing in January are 0.79 by GRU and 0.62 by SimpleRNN and 0.76 by LSTM and 0.80 by LSTM in July, respectively. However, the averages of the best results are 0.78 by LSTM and GRU and 0.73 by LSTM. In Table 2, the validation error is higher than the test error in January, while the situation of July is the opposite. The average error of testing is better than the validation results.

4.3.2 Results on the cycle series

The results of the cycle series are shown in Table 3. Among them, the best results of validation and testing in January are 4.30 by GRU and 5.27 by GRU and 1.59 by GRU and 1.70 by SimpleRNN in July, respectively. However, the averages of best results are 2.94 and 3.57 by GRU.

According to the above experiment results, we can see that the results of the short-term series MAPE are better overall than those of the cycle series. Compared with the different methods, the LSTM method performs slightly better than the other two methods by the average results of short-term series, and the GRU method is slightly better than the other two methods in regard to the average results of the cycle series. However, the results of the short-term series are better than those of the cycle series. Anyway, the three methods have similar results by the same series, based on 10,000 epochs, the same batch size and learning rate, the same number of neurons in each layer and the same dimensions.

4.3.3 Results on long short-term series

According to the results of the long short-term series in Table 4, compared to the three methods in our model architecture, the best results of validation and testing in January are 0.73 by LSTM and 0.64 by SimpleRNN and 0.77 by LSTM and 0.78 by LSTM in July, respectively; the averages are 0.75 and 0.73 by LSTM and GRU.

4.3.4 Results on the cross-long short-term series

The results of the cross-long short-term series are shown in Table 5. A comparison of the results by the three methods in our model architecture shows that the best results of validation and testing in January are 1.27 by LSTM and 1.05 by GRU and 1.14 by GRU and 1.08 by GRU in July, respectively. However, the averages of best results are 1.29 and 1.06 by GRU.

From the above experiment results, we can see that the results of long short-term series MAPE are better overall than those of the cross-long short-term series. Comparisons of the different methods in Table 4 show that the LSTM and GRU methods are slightly better than the other method according to the average results. In Table 5, the GRU method is shown to be slightly better than other two methods according to the average results. However, the results of the long short-term series are better than those of the cross-long short-term series. As the above shows, the results of both the short-term series sequence and long short-term series sequences are clearly better than the cycle series and cross-long short-term series completely based on the same situation. In Table 6,

Table 4 Performance of the long short-term series dataset

Method	January		July		Average	
	$MAPE_{val}$	$MAPE_{test}$	$MAPE_{val}$	$MAPE_{test}$	$MAPE_{val}$	$MAPE_{test}$
SimpleRNN	0.94	0.64	0.79	0.84	0.86	0.74
LSTM	0.73	0.68	0.77	0.78	0.75	0.73
GRU	0.78	0.67	0.79	0.80	0.78	0.73

Table 5 Performance of the cross short-term series dataset

Method	January		July		Average	
	$MAPE_{val}$	$MAPE_{test}$	$MAPE_{val}$	$MAPE_{test}$	$MAPE_{val}$	$MAPE_{test}$
SimpleRNN	2.10	1.59	2.13	1.73	2.11	1.66
LSTM	1.27	1.07	2.03	1.12	1.65	1.09
GRU	1.46	1.05	1.14	1.08	1.29	1.06

Table 6 Performance comparison of four-time series datasets with best model

Method	January	July	Average
	$MAPE_{test}$	$MAPE_{test}$	$MAPE_{test}$
T1 (LSTM)	0.67	0.80	0.73
T2 (GRU)	5.27	1.88	3.57
T3 (LSTM)	0.68	0.78	0.73
T4 (GRU)	1.05	1.08	1.06

we summarize the best results from four different sequence data by three methods: short-term series (T1), cycle series (T2), long short-term series (T3) and cross-long short-term series (T4). These three methods show no much difference in performance measure based on the same series sequence. In fact, the combination model can bring into play the advantages of each method and a comparison of the effects of different sequences becomes even clearer. We discuss the combination situations and analyze these results in the next section.

4.4 Results of different multiple time series combinations

In this section, we discuss the combination results from different sequences: (1) the short-term series combined with the cycle series (T1 + T2); (2) the long short-term series combined with the cross-long short-term series (T3 + T4); (3) the short-term series combined with the long short-term series (T1 + T3); (4) the short-term series combined with the long short-term series, adding up the cross-long short-term series (T1 + T3 + T3); and (5) finally combining the four series sequences (T1 + T2 + T3 + T4) together. The detailed results of these different models are shown in Tables 7, 8, 9, 10, 11 and 12.

4.4.1 Results on the short-term series and cycle series combination (T1 + T2)

The results of combining the short-term series and cycle series are shown in Table 7. In a comparison of the results by the three methods in our model architecture, the best results among them regarding validation and testing in January are 0.80 by GRU and 0.64 by GRU and 0.81 by GRU and 0.78 by LSTM in July, respectively. The best results of the average situation are 0.80 and 0.73 by GRU.

4.4.2 Results of the long short-term series and cross-long short-term series combination (T3 + T4)

The results of combining the long short-term series and the cross-long short-term series in three different methods in Table 8 show that the best testing results of January are 0.67 by GRU and 0.75 of July by GRU, while the average best of GRU is 0.71. Besides, compared to the results of the long short-term series and cross-long short-term series, the combination results are better than the long short-term series and the cross-long short-term series individually by the average value of MAPE.

The following shows the combination results of the short-term series and long short-term series. In fact, for both the short-term series and the long short-term series, these results are better than for the other two kinds of sequence, respectively. The detailed results of the combinations of the three methods are as follows.

4.4.3 Results of the short-term series and long short-term series combinations (T1 + T3)

As Table 9 shows the results of combining the short-term series and the long short-term series in three different methods, the best testing result of January by GRU is 0.63, and of July by GRU it is 0.77, while the average best result of GRU

Table 7 Performance of the short-term series and cycle series datasets

Method	January		July		Average	
	$MAPE_{val}$	$MAPE_{test}$	$MAPE_{val}$	$MAPE_{test}$	$MAPE_{val}$	$MAPE_{test}$
SimpleRNN	0.92	0.67	0.80	1.02	0.86	0.84
LSTM	0.84	0.68	0.99	0.78	0.91	0.73
GRU	0.80	0.64	0.81	0.83	0.80	0.73

Table 8 Performance of the long short-term series and cross-long short-term series datasets

Method	January		July		Average	
	$MAPE_{val}$	$MAPE_{test}$	$MAPE_{val}$	$MAPE_{test}$	$MAPE_{val}$	$MAPE_{test}$
SimpleRNN	0.82	0.77	0.91	0.82	0.86	0.79
LSTM	0.86	0.69	0.89	0.77	0.87	0.73
GRU	0.79	0.67	0.77	0.75	0.78	0.71

Table 9 Performance of the short-term series and long short-term series datasets

Method	January		July		Average	
	$MAPE_{val}$	$MAPE_{test}$	$MAPE_{val}$	$MAPE_{test}$	$MAPE_{val}$	$MAPE_{test}$
SimpleRNN	1.22	0.92	0.80	0.91	1.01	0.91
LSTM	0.84	0.66	0.75	0.78	0.79	0.72
GRU	0.74	0.63	0.75	0.77	0.74	0.70

Table 10 Performance of the three series dataset combinations

Method	January		July		Average	
	$MAPE_{val}$	$MAPE_{test}$	$MAPE_{val}$	$MAPE_{test}$	$MAPE_{val}$	$MAPE_{test}$
SimpleRNN	0.79	0.87	0.98	0.87	0.88	0.87
LSTM	0.77	0.65	0.96	0.78	0.86	0.71
GRU	0.78	0.61	0.75	0.74	0.76	0.67

Table 11 Performance of the four series dataset combinations

Method	January		July		Average	
	$MAPE_{val}$	$MAPE_{test}$	$MAPE_{val}$	$MAPE_{test}$	$MAPE_{val}$	$MAPE_{test}$
SimpleRNN	0.89	0.74	0.92	0.76	0.90	0.75
LSTM	0.82	0.68	0.88	0.74	0.85	0.71
GRU	0.79	0.63	0.73	0.70	0.76	0.66

is 0.70. Besides, compared to the results of combining the short-term series and long short-term series, the long short-term series and the cross-long short-term series are slightly better by the average MAPE.

All of the above are combinations of two kinds of sequence results; however, the time series relevance of different sequences had many other combination situations in our model, such as combining the short-term series and long short-term series, adding up the cross-long short-term series and combining the four series sequence. The results are shown in Tables 10 and 11, respectively.

Table 12 Performance comparison of the different MTS dataset combinations

Method	January	July	Average
	$MAPE_{test}$	$MAPE_{test}$	$MAPE_{test}$
T1 + T2 (LSTM)	0.68	0.78	0.73
T3 + T4 (GRU)	0.67	0.75	0.71
T1 + T3 (GRU)	0.63	0.77	0.70
T1 + T3 + T4 (GRU)	0.61	0.74	0.67
T1 + T2 + T3 + T4 (GRU)	0.63	0.70	0.66

4.4.4 Results of the three MTS and four MTS

As shown in the Tables 10 and 11, the January best result of GRU is 0.61, the July best result of GRU is 0.74, and the average best of GRU is 0.67 according to the MAPE results. The short-term series and long short-term series were combined and the cross-long short-term series were added up. Afterward, according to the combined results of the four series sequences, we can see that the January best result of GRU is 0.63 and the July best result of GRU is 0.70, while the average best result of GRU is 0.66 in Table 10. Besides, compared to the results of combining the four series sequences, the short-term series and long short-term series and added up cross-long short-term series results were slightly worse regarding the average MAPE.

Finally, in order to perform a comparison with other methods, we list the best results of the combination situations in Table 12. From the average perspective, the January best MAPE of the GRU method is 0.63 and the July best MAPE of the GRU method is 0.70, while the average best MAPE of the GRU method is 0.66.

According to all of the above results, we can make a simple summary. Firstly, from the points of view of different sequences independently, the short-term series and long short-term series are better than the cycle series and the cross-long short-term series based individually based on the same situation in all methods; this can reflect the major characteristics of the time series in the STLF. Then, from the points of view of the combined situations, the multiple time series combinations clearly can improve the performance. Finally, in a comparison of the three methods, the LSTM and GRU methods are slightly better than the SimpleRNN method in regard to the average MAPE. In the next section, we compare the results with other traditional methods for short-term load forecasting.

4.5 Comparison with other methods

In order to verify whether our proposed model based on multiple time series recurrent neural network units for short-term load forecasting exceeds the efficiency and performance of other methods, we compared it with other popular methods based on the same test tasks, for example neural network (NN) and support vector regression (SVR). Especially, Dudek (2015) proposed a hybrid model based on an artificial immune system (AIS) for the STLF; other important methods were compared in that paper, such as k-nearest neighbor (*k*-NN) algorithm, fuzzy Nelder–Mead (FNM), GM. As shown in Table 13, the *k*-NN results show that the average of January and July testing errors was 1.23. The average errors of FNM and GM were almost same, 1.09 and 1.08, respectively; the average errors of NN and SVR were almost the same as well as, 1.06 and 1.05, respectively. However, the January prediction errors based on the AIS model were 1.20 and 0.86 for July; the average, 1.03, was superior to the above methods. The best results of the model based on the recurrent neural network units that we proposed was 0.63 for January and 0.70 for July; the average was 0.66, which exceeded those of the above methods.

Table 13 Performance comparison of the different approaches

Approach	January	July	Average
	$MAPE_{test}$	$MAPE_{test}$	$MAPE_{test}$
k-NN	1.47	0.99	1.23
FNM	1.22	0.96	1.09
GM	1.21	0.96	1.08
NN	1.13	0.99	1.06
SVR	1.14	0.97	1.05
AIS	1.20	0.86	1.03
MTS-RNN (GRU)	0.63	0.70	0.66

As shown in Table 13, our methods-based multiple time series RNN model for STLF showed significant performance improvement. Our proposed model is superior to other models for short-term load forecasting. Besides, to illustrate the superiority of our model, we also evaluated the performance by MAE and MSLE. In Table 14, the MAE and MSLE show that the MAE best average was 1.29E+2 by the GRU method, while the MSLE best average was 1.21E-4 by the LSTM method.

4.6 Discussion

In our experiment results, we found a strange phenomenon: The prediction results are very good when we consider the short-term series and the long short-term series individually, while there is no significant improvement when we consider the cycle series and the cross-long short-term series individually. However, the results of combining them show some improvement compared to considering each sequence individually. This phenomenon can be explained as follows: The short-term series and the long short-term series sequences follow subtle changes in the history information, but the cycle series and the cross-long short-term series provide guidance on trend changes, while ignoring the influence of subtle changes. As a result, the prediction results were stable when we just considered the subtle changes in the information, such as with the short-term series and the long short-term series sequences. When we considered the trend changes in the history information, the prediction results were poor, such as with the cycle series and the cross-long short-term series. When the subtle changes in the information were combined with the trend changes, there were certain improvements in the results of the load forecasting.

Table 14 Performance evaluation for our proposed MTS-RNN system using T1 + T3 MTSs

Method	MAPE			MAE			MSLE		
	Jan.	July	Average	Jan.	July	Average	Jan.	July	Average
SimpleRNN	0.92	0.91	0.91	1.87E+2	1.69E+2	1.78E+2	2.50E−4	2.53E−4	2.51E−4
LSTM	0.66	0.78	0.72	1.82E+2	1.45E+2	1.63E+2	0.94E−4	1.49E−4	1.21E−4
GRU	0.63	0.77	0.70	1.34E+2	1.25E+2	1.29E+2	1.03E−4	1.54E−4	1.28E−4

5 Conclusions

The distribution and use of power resources in regard to economic regulation and resource allocation for power companies' budgets and government departments' control involve a crucial process. Less prediction error is not only a significant result for research work on resources, but also for actual power operation. In this paper, we proposed a framework based on a deep learning method for short-term load forecasting in order to verify whether the time series characteristics have a great impact on the history electric load data. We designed a model based on multiple time-steps RNN units such as SimpleRNN, LSTM and GRU units, taking four input sequences into account, with different time steps; the short-term series, the cycle series, the long short-term series and the cross-long short-term series, respectively. However, the multiple time series RNN units model has less prediction error compared with the other methods. The experiments attest that our model is very effective.

However, many aspects deserve future exploration. Our research in this field includes the following: Firstly, in order to effectively memorize the time series characteristics of history load information, we intend to use the bidirectional LSTM (BLSTM) for strengthening our model's LSTM layers. Secondly, it is important to explore the connection between weather conditions and electricity load consumption, such as the temperature and humidity. Finally, we are interested in applying our model to other prediction issues, for example stock price forecasting and film score prediction.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

Badar E (2011) Comparison of conventional and model load forecasting techniques based on artificial intelligence and expert system. Int J Comput Sci Issues 8(5):1694–1814

Bahrami S, Hooshmand RA, Parastegari M (2014) Short term electric load forecasting by wavelet transform and grey model improved by PSO (particle swarm optimization) algorithm. Energy 72:434–442

Çevik HH, Çunkaş M (2016) A fuzzy logic based short term load forecast for the holidays. Int J Mach Learn Comput 6(1):57–61

Chaturvedi DK, Sinha AP, Malik OP (2015) Short-term load forecast using fuzzy logic and wavelet transform integrated generalized neural network. Electr Power Energy Syst 67:230–237

Che JX, Wang JZ (2014) Short-term load forecasting using a kernel-based support vector regression combination model. Appl Energy 132:602–609

Cheepati KR, Prasad TN (2016) Performance comparison of short term load forecasting techniques. Int J Grid Distribut Comput 9(4):287–302

Cho K, Van Merrienboer B, Bahdanau D, Bengio Y (2014) On the properties of neural machine translation: encoder–decoder approaches. arXiv preprint arXiv:1409.1259

Dauphin YN, de Vries H, Chung J, Bengio Y (2015) RMSProp and equilibrated adaptive learning rates for non-convex optimization. arXiv:1502.04390v1

Dedinec A, Filiposka S, Dedinec A, Kocarev L (2016) Deep belief network based electricity load forecasting: an analysis of Macedonian case. Energy 115(13):1688–1700

Deng J, Jirutitijaroen P (2010) Short-term load forecasting using time series analysis: a case study for Singapore. In: Proceeding of the 4th IEEE CIS & RAM, pp 231–236

Dudek G (2008) Artificial immune system for short-term electric load forecasting. Proc CAISC 2008:1007–1017

Dudek G (2011) Artificial immune clustering algorithm to forecasting seasonal time series. Proc ICCCI 2011:468–477

Dudek G (2015) Pattern similarity-based methods for short-term load forecasting—part 2: models. Appl Soft Comput 36:422–441

Dudek G (2016) Pattern-based local linear regression models for short-term load forecasting. Electr Power Syst Res 130:139–147

Dudek G (2017) Artificial immune system with local feature selection for short-term load forecasting. IEEE Trans Evol Comput 21(1):116–130

Ghayekhloo M, Menhaj MB, Ghofrani M (2015) A hybrid short-term load forecasting with a new data preprocessing framework. Electr Power Syst Res 119:138–148

Ghofrani M, Ghayekhloo M, Arabali A, Ghayekhloo A (2015) A hybrid short-term load forecasting with a new input selection framework. Energy 81:777–786

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

Hochreiter S, Schmidhuber J (1997b) Long short-term memory. Neural Comput 9(8):1735–1780

Hodge BM, Lew D, Milligan M (2013) Short-term load forecasting error distributions and implications for renewable integration studies. In: Proceeding of the IEEE green technologies conference Denver, pp 1–8

Hong T, Wilson J, Xie J (2014) Long term probabilistic load forecasting and normalization with hourly information. IEEE Trans Smart Grid 5(1):456–462

Hosein S, Hosein P (2016) Improving power generation efficiency using deep neural networks. In: Proceeding of the 2016 ICML Workshop on #Data4Good: machine learning in social good applications, pp 1–5

Hu Z, Bao Y, Xiong T (2014) Comprehensive learning particle swarm optimization-based mimetic algorithm for model selection in short-term load forecasting using support vector regression. Appl Soft Comput 25:15–25

Hu Z, Bao Y, Xiong T (2015) Hybrid filter-wrapper feature selection for short-term load forecasting. Eng Appl Artif Intell 40:17–27

Khan GM, Zafari F, Mahmud SA (2013) Very short term load forecasting using Cartesian genetic programming evolved recurrent neural networks (CGPRNN). In: Proceeding of the 12th international conference on machine learning and applications, pp 152–155

Kouhi S, Keynia F, Ravadanegh SN (2014) A new short-term load forecast method based on neuro-evolutionary algorithm and chaotic feature selection. Electr Power Energy Syst 62:862–867

LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(28):436–444

Mateo F, Carrasco JJ, Milla'n-Giraldo M (2013) Machine learning techniques for short-term electric power demand prediction. Proc Comput Intell Mach Learn Bruges 2013:143–148

Medeiros MC, Soares LJ (2006) Robust statistical methods for electricity load forecasting. In: Proceeding of the RTE-VT workshop, pp 29–30

Nataraja C, Gorawar B, Shilpa GN et al (2012) Short term load forecasting using time series analysis: a case study for Karnataka, India. Int J Eng Sci Innov Technol 1(2):45–53

Nie H, Liu G, Liu X (2012) Hybrid of ARIMA and SVMs for short-term load forecasting. Energy Procedia 16:1455–1460

Niu DX, Shi HF, Wu DD (2012) Short-term load forecasting using bayesian neural networks learned by hybrid Monte Carlo algorithm. Appl Soft Comput 12(6):1822–1827

Oliveira MO, Marzec DP, Bordin G (2011) Climate change effect on very short-term electric load forecasting. In: Proceeding of the 2011 IEEE Trondheim power tech, pp 1–7

Shelke M, Thakare PD (2014) Short term load forecasting by using data mining techniques. Int J Sci Res 3(9):1363–1367

Shrivastava A, Bhandakkar A (2013) Short-term load forecasting using artificial neural network techniques. J Eng Res Appl 3(5):1524–1527

Taylor JW (2012) Short-term load forecasting with exponentially weighted methods. IEEE Trans Power Syst 27:458–464

Vermaak J, Botha EC (1998) Recurrent neural networks for short-term load forecasting. IEEE Trans Power Syst 13:126–132

Zhang P, Wu X, Wang X (2015) Short-term load forecasting based on big data technologies. CSEE J Power Energy Ssyst 1(3):59–67