A Short-Term Electricity Forecasting Scheme Based on Combined GRU Model with STL Decomposition

Y J Tian¹, S J Zhou², M Wen² and J G Li²

¹State Grid Shanghai Municipal Electrical Power Company, Shanghai, China
²Computer Science and Technology, Shanghai University of Electric Power, Shanghai, China

E-mail: 13901712348@163.com

Abstract. With development of smart grid, the stable operation of grid has put forward higher requirements for system dispatch. In particular, short-term load forecasting of power systems is a key factor of power grid management systems, which is related to the safety, economy, and stable operation of the smart grid. However, short-term electricity forecasting is affected by many external factors. It has complex characteristics, especially non-linear relationships, so it cannot be accurately predicted. Recently, Recurrent Neural Network based models have good performance in electricity forecasting because of their excellent ability to capture non-linear relationships. However, they cannot fully capture historical information, especially local historical information, which has an impact on prediction accuracy. In order to address these problems, we propose a scheme by combining STL decomposition and GRU model. Specifically, we first decompose the original time series into three different components by STL. The decomposition results are separately imported into the main prediction module, which uses two GRU networks with different structures to obtain the local and global dependencies of the data. We also add an autoregressive method to make the model more robust. The proposed scheme is validated based on real-world data, and the simulation results show that our proposed method can perfectly capture local and global information and achieve higher prediction accuracy than traditional models.

1. Introduction

Nowadays, electric vehicles, distributed generations and other technologies and applications have developed rapidly. As an important part of smart cities, smart grids use many smart devices and advanced management systems, which are more efficient and reliable than traditional power grids [1]. Accurate short-term electricity forecasting is beneficial to power dispatching, power system security, and energy transfer scheduling. It can fully optimize the resource allocation of the power system. In addition, it can also reduce the impact caused by the integration of new energy into the grid [2].

Electricity forecasting belongs to the time series forecasting paradigm and can be divided into three categories according to the forecast range: short-term (predicting hours to days), medium-term (predicting weeks to months), and long-term (future years). Because of various external factors, such as weather conditions, economic fluctuations, electricity data often have highly complex non-linear patterns, which makes short-term electricity forecasting the most challenging problem [3].

In recent years, many statistical-based methods have been proposed for time series prediction problems. There are some successful methods to achieve good accuracy, such as autoregressive integrated moving average (ARIMA) [4] and support vector machine (SVM) [5]. However, these
statistical-based methods often cannot fully capture complex non-linear relationships because they often take the form of a time series distribution or function. Recently, Artificial neural networks are widely used because of their ability to capture non-linear relationships. It can also solve the classification and regression problems in the field of power system management. In particular, the Recurrent Neural Network (RNN) [6] is considered as a new starting point for sequence modeling. But the traditional RNN has the problem of gradient disappearance during the training phase. Therefore, Long Short-Term Memory network (LSTM) [7] and the Gated Recurrent Units (GRU) [8] have been proposed, and they have proven to be very efficient. A GRU network short-term load forecasting model based on a deep learning framework is proposed [9]. However, due to the use a pure GRU model, the proposed model is not robust enough, and it ignores the local correlations of historical data, and cannot capture all historical information.

In order to accurately and robustly predict electricity data, we propose a short-term electricity forecasting scheme, base a combined network for efficient time series prediction. The main contributions of this paper are three-fold.

- We use the STL decomposition method to decompose the original data into three parts to reduce the interaction between different components.
- Each component independently inputs two GRU networks with different structures in parallel, which we call GlobalGRU and LocalGRU respectively, ensuring better capture of global and local dependencies.
- To improve the robustness of the forecasting scheme, we also add a linear autoregressive model in parallel with the GRU networks. Simulation results show that the proposed scheme can perfectly capture local and global information and achieve higher prediction accuracy.

The rest of the paper is organized as follows. Section 2 introduces the model of the prediction system. Section 3 explains the prediction method one by one. Section 4 is our experiment and analysis. Finally, we conclude the paper in Section 5.

2. System model

In the system model, electricity dataset can be seen as a time series \( x = \langle x_1, x_2, ..., x_T \rangle \), which is a sequence of recording measurements arranged in chronological order and with a constant time interval. \( x_T \) represents the record corresponding to the time stamp \( T \).

Figure 1 shows the sliding window prediction method we used. In particular, given a set of time series data \( x = \langle x_1, x_2, ..., x_T \rangle \), where \( T \) is the length of the input window size, we predict \( x_{T+h} \) based on the known \( \langle x_1, x_2, ..., x_T \rangle \), where \( h \) is the forecast length from the current time stamp. Moreover, we predict the future electricity value \( x_{T+h+n} \) based on historical data \( \langle x_{1+n}, x_{2+n}, ..., x_{T+n} \rangle \), \( n \in \mathbb{R}^+ \). Finally, we summarize the sliding prediction method that for the target \( x_{T+h} \in \mathbb{R} \), and the input vector at \( T \) is \( x = \langle x_1, x_2, ..., x_T \rangle \in \mathbb{R}^{T \times 1} \).

![Figure 1. Sliding window forecasting method.](image-url)
3. The proposed scheme

Figure 2 presents the structure of our proposed scheme. The model consists of a STL decomposition and a dual recurrent neural network, and the dual recurrent neural network is combined by GlobalGRU Layer and LocalGRU Layer. The raw data is decomposed by the STL to generate three components, which are input into the forecasting model respectively. GlobalGRU layer is used to capture long-term global dependencies, LocalGRU layer can capture local dependencies, and then merge the two parts of the results. Then, combine autoregressive (AR) components to improve the robustness and accuracy of the forecasting scheme. Finally, the forecasting module aggregates the forecast results of the GRU network and AR model to generate the final forecasting results. We will describe the proposed scheme in detail in the following paragraphs.

![Figure 2](image)

**Figure 2.** The structure of proposed scheme.

3.1. Time Series Decomposition

Because the electricity has great fluctuation and instability, it is very difficult to directly predict the original data. Therefore, it is essential to analyse and process the original data. STL (Seasonal and Trend decomposition using Loess) [10] is an iterative nonparametric process, which is a time series decomposition method based on robust local weighted regression as a smoothing method.

In our scheme, the time series of electricity data is decomposed into three parts using STL decomposition: trend, seasonal, and residual (Figure 3), with the aim of reducing the interaction between different components. Predicting different components separately can also make the prediction structure more accurate. Concretely, seasonal components are found by smoothing the loess (local regression) of the original time series. Smooth the rest to find trends. The remainder of the last section represents the residuals for seasonal and trend fitting.

These time series data are input into the prediction module for the next phase. In the final prediction stage, the three components are linearly summed to restore the original time series.

![Figure 3](image)

**Figure 3.** Original data and decomposed data.
3.2. Forecasting Module
The prediction module consists of three parts. The deep neural network part is called GlobalGRU and LocalGRU. GlobalGRU layer is used to capture global dependencies, and LocalGRU layer is used to capture local dependencies. The linear regression part uses autoregressive method. These three parts are introduced below.

3.2.1. GlobalGRU Layer. Considering the gradient vanishing problem of classical RNN in long sequence, we use the Gated Recurrent Units (GRU) [8] network as the main prediction structure for this layer. Figure 4 shows the structure of GRU. The model combines the input and forgot gate into update gate \( z_t \) based on LSTM, and replacing the output gate with reset gate \( r_t \). In addition, the unit hidden states and output are combined into one state parameter \( S \). Therefore, the number of gates is changed from 3 to 2. As the training parameters decrease, the training speed of the network will increase. The formula of GRU is as follows:

\[
\begin{align*}
    z_t &= \sigma(W_z \cdot [h_{t-1}, x_t]) \\
    r_t &= \sigma(W_r \cdot [h_{t-1}, x_t]) \\
    \tilde{h}_t &= \tanh(W \cdot [r_t \cdot h_{t-1}, x_t]) \\
    h_t &= (1 - z_t) \cdot h_{t-1} + z_t \cdot \tilde{h}_t
\end{align*}
\]

Where \( t \) is the current time step, \( x_t \) is the input data at time \( t \), \( W \) is the weight matrix. We use multiple GlobalGRU layers to obtain global nonlinear correlation and take the last hidden state \( h^G_t \) as the output of GlobalGRU.

![Figure 4. The structure of GRU.](image)

3.2.2. LocalGRU Layer. R-Transformer [11] model proposes a LocalRNN method to improve the transformer model [12] in natural language processing, which can obtain local dependencies information in long sequence modelling. Similarly, we use LocalRNN to capture the local dependencies in the time series, and specifically use the GRU network, which we call LocalGRU. Specifically, the original long sequence is reconstructed into a short sequence that only contain local information and are processed independently and identically by the shared GRU. A local window of size \( N \) is constructed for the target position in the sequence, which contains \( N \) consecutive sequences and ends at the target position. Thus, each local window forms a local short sequence from which the potential representation is learned using the shared GRU. Figure 5 shows the different between GlobalGRU and LocalGRU operations. Given the positions \( x_{t-N-1}, x_{t-N-2}, \ldots, x_t \) of local short sequences of length \( N \), the LocalGRU processes them sequentially and output the last hidden state:

\[
h_t = \text{LocalGRU}(x_{t-N-1}, x_{t-N-2}, \ldots, x_t)
\]

Therefore, from a sequence perspective, LocalGRU takes a sequence of window length \( T \) and outputs a sequence containing a hidden representation of the local area information:
\[ h_{1}, h_{2}, \ldots, h_{T} = \text{LocalGRU}(x_{1}, x_{2}, \ldots, x_{T}) \]  
(6)

After learning from multiple LocalGRU layers, the last hidden state \( h_{T} \) is used as the output of LocalGRU part. Finally, we concatenate the output of the GlobalGRU and LocalGRU parts together and use a fully connected layer to obtain the prediction results \( h_{T}^{GGRU} \) of the deep learning part.

\[ h_{T}^{GGRU} = \text{Dense}(\text{Concatenate}(h_{T}^{C}, h_{T}^{L})) \]  
(7)

![Diagram of GRU and LocalGRU](image)

**Figure 5.** Different between GlobalGRU and LocalGRU.

3.2.3. Autoregressive Component. Due to the non-linearity of the recurrent neural network, the scale of the neural network output is not sensitive to the scale of the input [13]. To solve this shortcoming, we mix linear and non-linear components to get the final prediction result of our proposed scheme. We use the classic AR model to obtain the linear features of the original data. The formula of the AR model is as follows:

\[ h_{T}^{AR} = \sum_{k=0}^{T-1} W_{k}^{AR} x_{T-k} + b^{AR} \]  
(8)

Where \( T \) denotes the length of input sequence, \( W_{k}^{AR} \) and \( b^{AR} \) are the parameters matrix and the bias terms, respectively. The final prediction result consists of the GRU part and the AR part:

\[ \hat{y}_{T} = \text{Dense}(h_{T}^{GGRU} + h_{T}^{AR}) \]  
(9)

4. Performance evaluation

In this section, we will evaluate our proposed scheme using a real-world dataset from a region in Shanghai and compare it with popular time series prediction methods (SVM, GRU). All time intervals of the sampling points are equal, which is one day. And all data are reduced from 0 to 1 using the min-max normalization. For all tasks, all models are trained using the same optimizer and the learning rate is selected from the same set of values based on the validation performance.

4.1. Dataset

In the experiments, the electricity dataset is divided into a training set (80%), a validation set (10%), and a test set (10%). In each set, the data is further divided into the data required by the model using a sliding window, which means that in each set of data, we use a electricity sequence of length \( T \) as the input data of our proposed scheme, and the timestamp \( T + h \) value as the real data to be predicted.
4.2. Experimental settings

For the neural network model, we use the Adam method as the optimizer and the mean square error (MSE) as the loss function. For GRU and our scheme, the hidden layer dimension is selected from \{64, 128, 256\}, and the total number of network layers is selected from \{3, 5, 7\}. The best average result was selected through multiple experiments. For SVM, we use grid search to find its best parameters automatically.

All methods are implemented using Python 3.6, and deep learning methods use the PyTorch 1.3.0 framework. And all experiments are performed using a machine with Intel Core i9-9900K CPU, NVIDIA GTX2080 GPU, 32 GB RAM.

4.3. Experimental results and analysis

Three evaluation criteria are used as performance metrics: mean absolute error (MAE), mean absolute percentage error (MAPE), and root mean square error (RMSE). Lower values are better for them. For sliding window forecasting model, we set the use of historical data window = \{15, 25, 35\} and forecast future horizon = \{3, 5, 10\}, which means using the historical data from 15 to 35 days to predict electricity values from 3 to 10 days. And in this paper, due to the space limitations, we only list the results of window = 25.

Table 1 presents the evaluation results of the parameters and methods mentioned above in the test set. Each column in the table shows the results of different methods under different conditions, and the best performing data in each case is shown in bold. Figure 6 shows a comparison of experimental results with window size = 25 and horizon = 5.

**Table 1. Evaluation results of electricity forecasting**

| Window and horizon | Metrics       | SVM [5] | GRU [9] | Our Scheme |
|--------------------|--------------|---------|----------|------------|
| 25-3               | MAE (10^6)   | 7.75    | 6.23     | 5.02       |
|                    | RMSE (10^6)  | 8.64    | 8.23     | 6.19       |
|                    | MAPE (%)     | 8.05    | 7.78     | 6.14       |
| 25-5               | MAE (10^6)   | 7.51    | 7.15     | 4.72       |
|                    | RMSE (10^6)  | 9.69    | 9.37     | 5.98       |
|                    | MAPE (%)     | 9.07    | 9.94     | 5.86       |
| 25-10              | MAE (10^6)   | 8.34    | 7.75     | 6.54       |
|                    | RMSE (10^6)  | 10.93   | 10.39    | 8.28       |
|                    | MAPE (%)     | 8.34    | 9.52     | 8.04       |

**Figure 6.** Histograms of evaluation results between different forecasting methods in different prediction horizon.
As can be seen from Table 1 and Figure 6, GRU and our scheme using deep learning methods have more predictive capabilities than SVM models using linear regression method, which indicates that deep learning can learn complex non-linear dependencies in time series. Therefore, deep learning methods can better solve complex prediction tasks than traditional methods. Compared with GRU, our method can better fit local changes, such as some peaks and valleys, indicating that the LocalGRU module we use can capture local dependencies excellently. Moreover, we also found in the experimental results that almost all the models will decrease the prediction accuracy as the horizon becomes larger. It can be seen from Figure 7 that no matter whether it is a global prediction or a local prediction, our method always fits the original electricity consumption better than other methods. This is consistent with our theoretical analysis.

![Comparison of forecast results.](image)

Figure 7. Comparison of forecast results. (a) is the performance of the results on the entire test set. We also take a small part and show it in (b).

5. Conclusion

Electricity forecasting can not only improve the reliability of power systems, but also reduce production costs. Thus, it is of great significance to achieve efficient energy management. In this paper, we propose a novel short-term electricity forecasting scheme by using STL decomposition combined with GRU. Firstly, the proposed scheme decomposes the original sequence into multiple components by STL. Then then uses global and local GRU networks to predict the components separately. In order to improve the robustness of prediction results, we also added an autoregressive linear method. Finally, experiments on real data sets show that the proposed scheme has higher prediction accuracy than traditional models and pure GRU models, it can capture local and global correlations, and our scheme is more robust. In the future, we plan to use more accurate decomposition methods and further improve the model structure to achieve more accurate prediction results.
Acknowledgements
This work is supported by the State Grid Shanghai Municipal Electric Power Company Project of China (No.5209280002B).

References
[1] Zheng J, Xu C, Zhang Z and Li X 2017 Electric load forecasting in smart grids using long-short-term-memory based recurrent neural network 2017 51st Annual Conference on Information Sciences and Systems (CISS) (IEEE) 1–6
[2] Huang F, Zheng X, Yu Z, Yang G and Guo W 2019 Electric load forecasting based on sparse representation model Green, Pervasive, and Cloud Computing Lecture Notes in Computer Science ed S Li (Cham: Springer International Publishing) 357–69
[3] Zhu G, Chow T-T and Tse N 2018 Short-term load forecasting coupled with weather profile generation methodology Build. Serv. Eng. Res. Technol. 39 310–27
[4] de O. Santos Júnior D S, de Oliveira J F L and de Mattos Neto P S G 2019 An intelligent hybridization of ARIMA with machine learning models for time series forecasting Knowledge-Based Systems 175 72–86
[5] Khan R A, Dewangan C L, Srivastava S C and Chakrabarti S 2018 Short term load forecasting using SVM models 2018 IEEE 8th Power India International Conference (PIICON) 1–5
[6] Elman J L 1990 Finding structure in time Cogn. Sci. 14 179–211
[7] Hochreiter S and Schmidhuber J 1997 Long short-term memory Neural Comput. 9 1735–80
[8] Chung J, Gulcehre C, Cho K and Bengio Y 2014 Empirical evaluation of gated recurrent neural networks on sequence modeling ArXiv14123555 Cs
[9] Gao X Y, Wang Y, Gao Y, Sun C Z, Xiang W and Yue Y M 2018 Short-term load forecasting model of GRU network based on deep learning framework 2018 2nd IEEE Conference on Energy Internet and Energy System Integration (EI2) 1–4
[10] Cleveland R B, Cleveland W S, McRae J E and Terpenning I 1990 STL: a seasonal-trend decomposition J. Off. Stat. 6 3–73
[11] Wang Z, Ma Y, Liu Z and Tang J 2019 R-Transformer: recurrent neural network enhanced Transformer ArXiv190705572 Cs Eess
[12] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Kaiser Lukasz and Polosukhin I 2017 Attention is all you need Advances in neural information processing systems 5998–6008
[13] Lai G, Chang W-C, Yang Y and Liu H 2017 Modeling long- and short-term temporal patterns with deep neural networks ArXiv170307015 Cs