IET Generation, Transmission & Distribution

Research Article

IET Journals
The Institution of Engineering and Technology

Multi-GRU prediction system for electricity generation's planning and operation

ISSN 1751-8687
Received on 4th July 2018
Revised 30th September 2018
Accepted on 2nd November 2018
E-First on 4th April 2019
doi: 10.1049/iet-gtd.2018.6081
www.ietdl.org

Weixian Li[1] ✉, *Thillainathan Logenthiran*[1], *Wai Lok Woo*[1]

[1]*School of Electrical & Electronic Engineering, University of Newcastle upon Tyne, SIT Building at Nanyang Polytechnic, 172A Ang Mo Kio Avenue 8 #05-01, Singapore 567739*

✉ *E-mail: w.li17@newcastle.ac.uk*

Abstract: Electricity generation's planning and operation have been key factors for any economic development in the power industries but it can only be achieved if the generation was accurately forecasted. This made forecasting systems essential to planning and operation in the electricity market. In this study, a novel system called multi-GRU (gated recurrent unit) prediction system was developed based on GRU models. It has four level of prediction process which consists of data collection and pre-processed module, multi-features input model, multi-GRU forecast model and mean absolute percentage error. The data collection and pre-processed module collect and reorganise the real-time data using the window method. Multi-features input model uses single input feeding method, double input feeding method, and multiple feeding method for features input to the multi-GRU forecast model. Multi-GRU forecast model integrates GRU variation such as regression model, regression with time steps model, memory between batches model, and stacked model to predict the future electricity generation and uses mean absolute percentage error to evaluate the prediction accuracy. The proposed systems achieved high accuracy prediction results for electricity generation.

1 Introduction

Electric power is a driving force for economic development that resulted in forecasting electricity generation becoming of a great importance for the management of power systems for successful efficiency planning [1, 2]. Forecasting of peak electricity generation is needed for planning and scheduling in the industry. It helps to make an important decision on electricity purchasing, load shifting, and infrastructure developments, this would greatly increase the reliability of balancing the supply and demand between the electricity generators and suppliers to companies [3–6].

Forecasting in electricity generations reached a significant improvement in performance due to the restructuring of industries [7]. Forecasting techniques like auto regressive integrated moving average (ARIMA) is often used for operational purposes that considered end users and economical situations [8–12]. There are different forecasting methods such as fuzzy systems, support vector machine (SVM), multi-linear regression (MLR) models and artificial neural network (ANN) [13–18]. The research increases the accuracy of electricity forecasting due to different factors being enhanced in forecasting tools. The factors include pre-processing of data, feature selections, and settings of forecasting methods.

Time series forecasting methods, however, are popular in electricity forecasting. It is commonly used with ANN's feed forward neural network (FFNN) in the deep learning fields due to its high accuracy of prediction [19]. Due to technology advancements, recurrent neural network (RNN) was used as one of the forecasting tools for electricity market but a variation of RNN such as gated recurrent unit (GRU) has yet to be used for forecasting of electricity generation. It is known that GRU solves long dependency issues by keeping relevant information and computes at relatively fast speeds, thus proving to be more advantageous to RNN, ARIMA, and SVM [20]. This made the implementation of a variety of GRU models a novelty for electricity generation forecasting.

Hoiles and Krishnamurthy [21] evaluated a non-parametric test to detect if the demand behaviour of consumers is consistent with time-of-day electricity tariff initiatives. It included several tests on various forecasting algorithms, but the method is only for the detection of consumers who participate in time-of-use electricity pricing initiatives that do not require the utility function of the consumers to be known.

Shrivastava and Panigrahi [22] proposed a novel approach for the generation of prediction intervals using a differential evolution-based multi-objective approach. It focuses on optimising SVM to predict the electricity demands and prices. However, comparison with other machine learning methods did not experiment.

Xu *et al.* [23] proposed an intelligent model with a focus on the prediction of distributed power generation and the analysis of regional power supply structure. SVM was implemented to forecast the utilisation coefficient of the distributed generation. The parameters of SVM were optimised using immune algorithm and fruit fly optimisation algorithm (IAFOA). The proposed IAFOA-SVM had the best mean absolute percentage error (MAPE) when benchmarked against conventional methods such as MLR, genetic algorithm optimised SVM, and an auto regressive moving average. However, the results were based on only a month data and it was further split into four quarters for the simulation. In addition, it did not include machine learning comparison in the proposed method. This shows the lack of forecasting systems that are developed for electricity generation's planning and operation.

This paper proposes a novel idea of multi-GRU prediction system (MPS) for electricity generation's planning and operation. This forecasting system is more accurate compared to previous methods. As the electricity market generation is a time series data, various advantageous predictive GRU models such as regression model (RM), regression with time steps model (RTSM), memory between batches model (MBBM), and stacked model (SM) are used for accurate predictions. Compared to most forecasting approaches in time series electricity data, its ability to retain relevant information and link it with the latest data point proves to be beneficial. The simulations of RNN, long short term memory (LSTM), GRU, and MPS were benchmarked against to find the highest accuracy system based on MAPE.

The major contributions of this paper are as follows:

- The proposed system is an adaptive prediction system that integrates machine learning and mathematic equations to achieve high accuracy results in multiple different environments.

- The proposed system modifies the preprocess data and combining the GRU with optimisation methods for better accuracy.
- Due to the proposed system high accuracy results, it will provide optimised planning of workers and resources for electricity generation. Additionally, it will avoid under or over generations.

The remaining paper is organised as follows: Section 2 shares the importance of forecasting in electricity generation's planning and operation. Section 3 presents the proposed MPS and algorithms. Section 4 provides the simulation results of all the different systems. Section 5 discusses the results obtained from all the implemented simulation. Finally, the paper is concluded in Section 6.

2 Importance of forecasting in electricity generation's planning and operation

In electricity markets, the importance of accurate forecasting is essential for the planning and operations of utility companies [24]. Overestimation of electricity generation will lead to supplying excessive energy and adds unnecessary operational cost. Underestimation results in the sudden deployment of additional peak capacity and purchases of necessity for generating at a higher cost [10]. As a result of this hostile trade-off, it was reported that a forecast error of an estimated 1% increased the operating cost of a UK power utility by £10 million in 1985 [25].

Erdener *et al.* [26] analysing the impacts of interdependencies between electricity and natural gas systems in terms of security of energy supply. The failure of adequately reloading fuel to the electricity system has an effect on the entire grid. This problem leads to research on a more accurate forecast system, it will increase the reliability of power supply and delivery system while saves a substantial amount of operating and maintenance costs. Additionally, it is a key requirement for the planning, economic, and secure operation of future power systems.

There are a few advantages to accurate electricity generation forecasting [27–30]:

- Enables the utility companies to plan and prepare well.
- Minimise the risks for the utility company.
- Understanding the future long-term electricity generation helps the company to plan and make economically viable decisions in regard to future generation and transmission investments.
- Determines the required resources such as fuels required to operate the generating plants as well as other resources that are needed to ensure uninterrupted but yet economical generation and distribution of the power to the consumers. This is important for short, medium, and long-term planning.
- The electricity generation forecasting helps in planning the future in terms of the size, location, and type of the future generating plant. By determining areas or regions with high or growing demand, the utilities will most likely generate the power near the load. This minimises the transmission and distribution infrastructures as well as the associated losses.
- Decides and plan for maintenance of the power systems. By understanding the generation, the utility knows when to carry out the maintenance and ensure that it has a minimum impact on the consumers. For example, they may decide to do maintenance in residential areas during the day when most people are at work and generation is very low.
- Maximum utilisation of power generating plants. The forecasting avoids under or over generations.

Hence, an accurate forecasting system is needed for efficient planning and operations in electricity generation. The forecasting system will reduce the failure of electricity supply and the cost of preparation for electricity generations.

3 Proposed MPS

Fig. 1 shows the overall architecture of the proposed MPS. MPS is designed for forecasting electricity generation. It collects real-time electricity generation data to predict the future of electricity production and pick the most accurate prediction model. The overall architecture comprises the following modules:

i. Data collection and pre-processed module.
ii. Multi-features input mode.
iii. Multi-GRU forecast model.
iv. MAPE.

The following steps are taken as shown in Fig. 1:

1. Collect real-time data.
2. Preprocess the data using the window method.
3. Separate the preprocessed data to single input feeding method (SIFM), double input feeding method (DIFM), and multiple feeding method (MFM).
4. Use step 3 as the input for the GRU with RM, RTSM, MBBM, and SM.
5. Lastly, calculate the MAPE for the prediction systems.

The data collection and pre-processed module retrieve the real-time data and pre-process using window method for MPS. The multi-features input model function inputs the number of features for multi-GRU forecast model to achieve different prediction results. It contains various methods such as SIFM, DIFM, and MFM. The multi-GRU forecast model function predicts the future electricity generations. It consists of different GRU variations such as RM, RTSM, MBBM, and SM.

3.1 Data collection and pre-processed module

3.1.1 Data collection and pre-processed module: real-time data: Real-time data was taken from Energy Market Company (EMC) historical data (3-5-2015 to 24-4-2016) for power generations by the electricity market in Singapore [31]. There are 48 periods each day which amounts to 30 min per period in the electricity generation for demand and each row of the data represents a period.

The collected data is pre-processed by using

$$X1_t + X2_t = Y_t \tag{1}$$

where t: time step, $X1_t$: combined-cycle gas turbine, cogeneration, trigeneration power, $X2_t$: steam turbine generation power, and Y_t: actual total generation power.

$X1_t$, $X2_t$, and Y_t are used for prediction of the next time step prediction output Y_{t+1}.

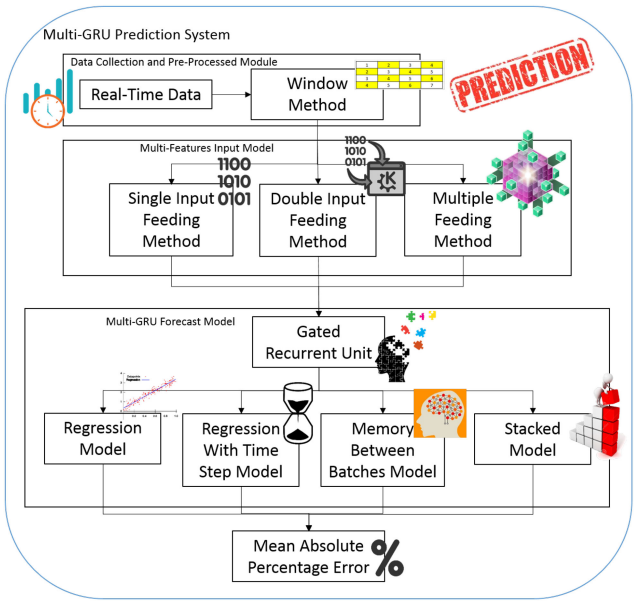

Fig. 1 *Overall MPS architecture*

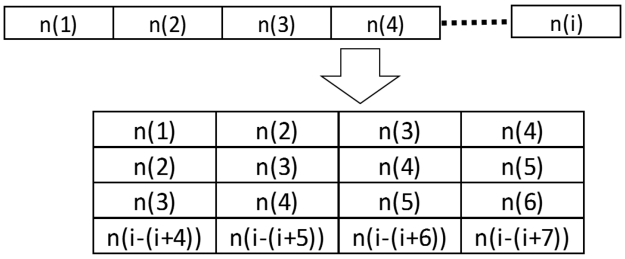

Fig. 2 *Window method with a window size of 4*

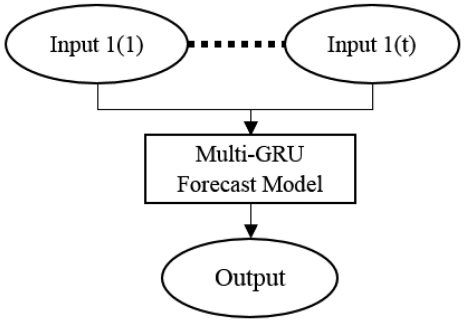

Fig. 3 *SIFM process diagram*

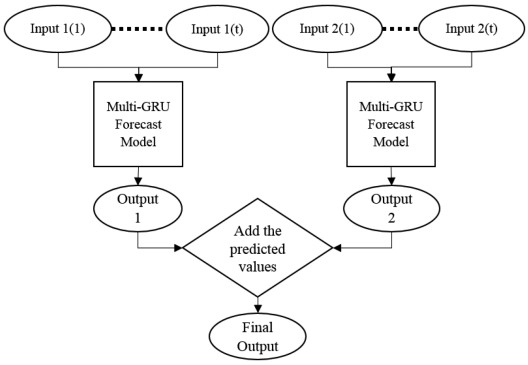

Fig. 4 *DIFM process diagram*

3.1.2 Data collection and pre-processed module: window method: Window method updates a sequence of values with the next value of the data within a fixed size of features. Fig. 2 shows an example of the window method with a window size of 4.

3.2 Multi-features input model

Multi-features input model process the features input to multi-GRU forecast model for different forecasting results. By using different methods, the possibility of distinct computation process results in accuracy differences. Thus, these methods will be tested to identify the most accurate prediction by the given data.

There are three methods for implementing the multi-features input model:

- SIFM
- DIFM
- MFM

These methods are further elaborated in the following sections.

3.2.1 Single input feeding method: SIFM utilises a single input feature to forecast the output results. An example is shown in Fig. 3.

3.2.2 Double input feeding method: DIFM utilises two input feature to forecast the output results. It is a method that uses two input individually to forecast the two output predicted values and add both to retrieve the final output. An example is shown in Fig. 4.

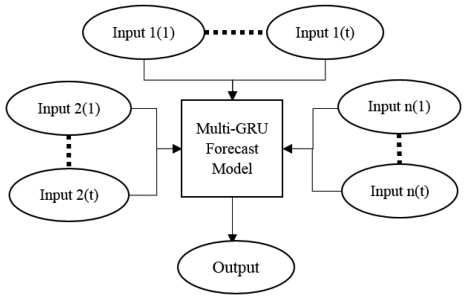

Fig. 5 *MFM process diagram*

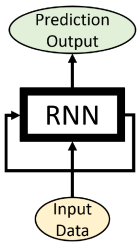

Fig. 6 *RNN diagram*

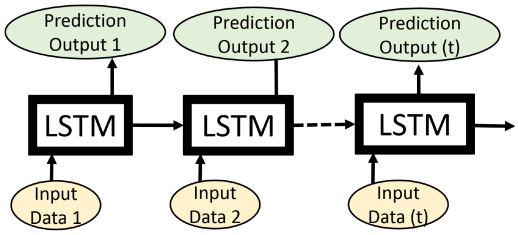

Fig. 7 *LSTM network diagram*

3.2.3 Multiple feeding method: MFM utilises multiple input feature to forecast the output results. It is a method that uses multiple input features to get the predicted output. An example is shown in Fig. 5.

3.3 Multi-GRU forecast model: GRU

Multi-GRU forecast model uses GRU variations to forecast future electricity generation.

A class of ANN known as RNN was designed to aim at the image, language text, and time sequence data recognition. It is a powerful neural network that has been used in data analytics, pattern prediction, and classifications for various industries [32].

Fig. 6 shows an RNN with loops which allows information to be preserved in the next process.

The strongest capability of RNN is using previous computations to compute the latest tasks. It allows RNN to use past information for learning if a situation where relevant computations and the latest tasks were separated into small gaps [33]. However, if the gap of the relevant computations is too far from the latest tasks then RNN will not be able to link up with the past information for the learning progression. A variation type of RNN called long short term memory (LSTM) network was introduced by Hochreiter and Schmidhuber [34] to solve the long-term dependency problem. It was popularised by industries because it works wonders on a different range of problems which made LSTM networks further refined for different purposes. Fig. 7 shows an illustration of how LSTM networks interact with each other [33].

There are three main gates in the LSTM block that manages the blocks state and output:

- Forget gate f_t: decides the information to forget in the block.
- Input gate i_t: decides which input values to update the memory state.
- Output gate o_t: decides the output dependent on the input and memory state.

Each block represents a state-based machine where each gate has its own weights that were learned during the training process [35]. This allows a large web of RNN to be created for addressing complicated problems and achieving optimum results.

A variation type of LSTM is the GRU which was introduced by Cho and co-authors [36]. This system has a single update gate that combines the input and output gate. It also merges the hidden and cell state which makes a simplified model than a standard LSTM model. Fig. 8 shows the details of the GRU system [33].

Fig. 8 shows each line representing an entire vector, from one output of a node to another input of a node. The grey circles are mathematic operations, while the orange boxes are the neural network layers. The lines denote the direction for the contents. The GRU layer was derived from the LSTM layer which results in similar equations:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \tag{2}$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \tag{3}$$

$$\underset{\sim}{h}_t = \tanh(W_h \cdot [r_t \cdot h_{t-1},\ x_t] + b_h) \tag{4}$$

$$h_t = (1 - z_t) \cdot h_{t-1} + z_t \cdot [+ \cdot 1ex]\tilde{h}_t \tag{5}$$

where t: time step, x_t: input value, h_t: output value, r_t: reset gate, z_t: update gate, $[+ .1ex]\tilde{h}_t$: candidate value, W_r: reset gate weights, W_z: update gate weights, W_h: candidate gate weights, b_z: update gate bias value, b_r: reset gate bias value, b_h: candidate bias value, and σ: gate state.

The reset gate determines the new input and the previous memory combination and the update gate determines the amount of previous memory to be kept. The idea of using a gating mechanism is similar to LSTM with an objective to learn long-term dependencies. The key differences are

- GRU has two gates while LSTM has three.
- GRU does not have an output gate and internal memory.
- GRU trains faster due to lesser parameters.

The backward components, simplified from LSTM, for GRU are defined as follows [37, 38]:

Note, the following representations are defined for simplicity:

$$Gates_t = \begin{bmatrix} r_t \\ z_t \\ \underset{\sim}{h}_t \end{bmatrix}, \quad W_t = \begin{bmatrix} W_r \\ W_z \\ W_h \end{bmatrix}, \quad b_t = \begin{bmatrix} b_r \\ b_z \\ b_h \end{bmatrix} \tag{6}$$

$$E_t = \frac{(T_t - h_t)^2}{2}$$

$$\partial_x E_t = T_t - h_t \tag{7}$$

$$\delta h_t = \partial_x E_t + \Delta h_t \tag{8}$$

$$\delta z_t = \delta h_t \cdot (\underset{\sim}{h}_t - h_{t-1}) \cdot z_t \cdot (1 - z_t) \tag{9}$$

$$\delta \underset{\sim}{h}_t = \delta h_t \cdot z_t \cdot (1 - \underset{\sim}{h}_t^2) \tag{10}$$

$$\delta r_t = \delta h_t \cdot z_t \cdot h_{t-1} \cdot (1 - \tanh^2(r_t)) \cdot r_t \cdot (1 - r_t) \tag{11}$$

$$\Delta h_{t-1} = \delta x_t = W_t^{\mathrm{T}} \cdot \delta Gates_t \tag{12}$$

$$\delta W_t = \sum_{i=0}^{t} \delta Gates_i \cdot x_i \tag{13}$$

$$\delta b_t = \sum_{i=0}^{t} \delta Gates_{i+1} \tag{14}$$

$$[W_{t+1}, b_{t+1}] = [W_t, b_t] - \eta \cdot \delta[W_t, b_t] \tag{15}$$

where E_t: square error, T_t: targeted value, and η: learning rate.

Equations (6)–(8) and (12)–(15) are implemented for both LSTM and GRU backward components algorithm. The difference is (9)–(11) that are derivations from the initial GRU algorithms. GRU and LSTM models both solve the long-term dependencies issues but GRU proves to be advantageous for faster computation speed due to a simpler process [35].

In GRU, there are various models for different applications and the following models such as:

- RM
- RTSM
- MBBM
- SM

These models were further elaborated in the subsections.

3.3.1 Regression model: Fig. 9 represents the sequence averages all the time steps output of the GRU to the average pooling resulting in representation h. These outputs are fed to a logistic regression layer whose target is the input sequence associated with the class label.

The logistic regression formula was represented as such

$$\sigma = 1/(1 + e^{-h}) \tag{16}$$

3.3.2 Regression with time steps model: It relabelled the data with time steps sequence which uses it to add another input feature. This method improves the accuracy of the prediction model. An example is shown in Fig. 10.

3.3.3 Memory between batches model: MBBM reuses the states for the samples of each batch as initial states for the samples in the next batch as shown in Fig. 11.

3.3.4 Stacked model: In this model, we stack numerous GRU layers on top of each other, making the model capable of learning

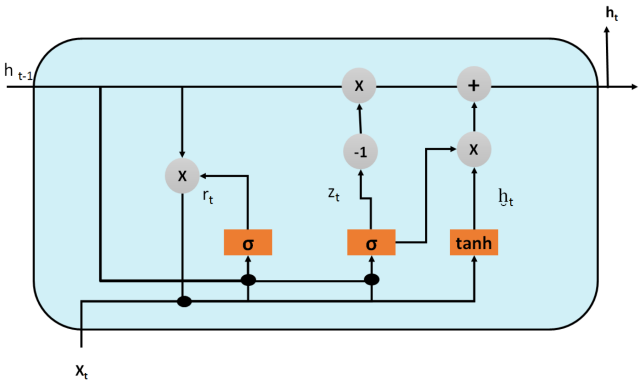

Fig. 8 *GRU block diagram*

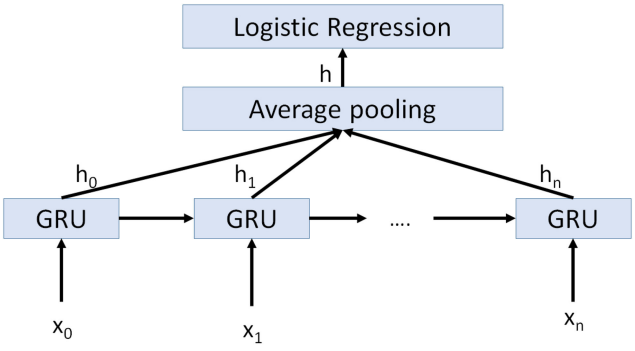

Fig. 9 *RM diagram*

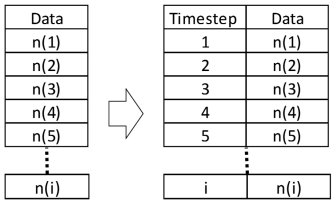

Fig. 10 *RTSM diagram*

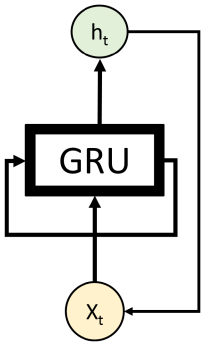

Fig. 11 *MBBM diagram*

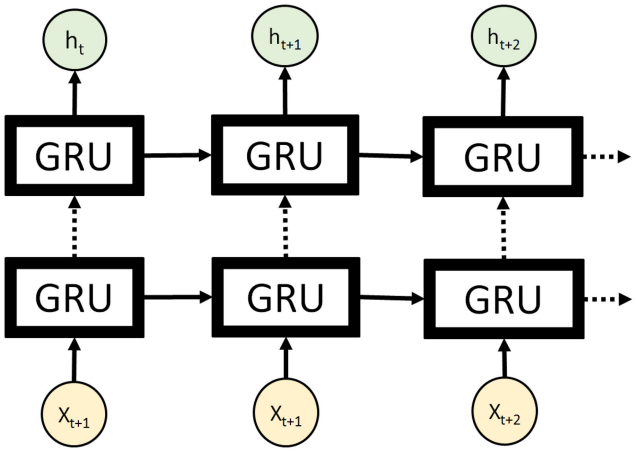

Fig. 12 *SM diagram*

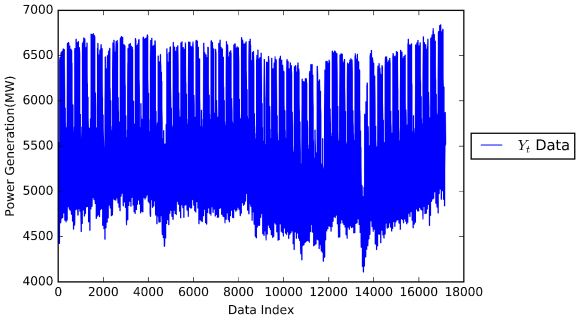

Fig. 13 *Plot of the actual total generation data graph*

higher-level temporal representations. The last layer of the GRU layers returns the last step in its output sequence which results in dropping the temporal dimension. An example such as Fig. 12 is shown for illustration.

3.4 Mean absolute percentage error

MAPE determines the accuracy result of MPS. A lower MAPE value shows a better system. The accuracy of the predicted data is then measured by MAPE

$$\text{MAPE} = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{A_i - F_i}{A_i}\right|, \quad \text{where } A_i \neq 0 \tag{17}$$

Table 1 Percentage results of MPS's RM

MPS's RM system	SIFM	DIFM	MFM	SIFM	DIFM	MFM
epoch		50			100	
runtime, s	750	800	800	1500	1600	1600
MAPE, %	0.4649	0.4953	0.1181	0.3538	0.3120	0.1083
max, %	3.5873	3.6151	1.4843	3.1487	3.5068	1.9485
min, %	0.00021	0.00036	0.00007	0.00016	0.00004	0.00008

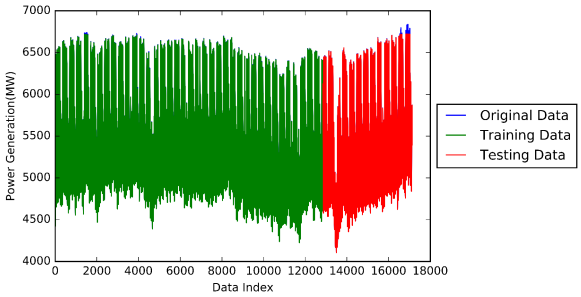

Fig. 14 *MPS's RM: 100 epochs MFM*

where n: number of data, A_i: actual output data, and F_i: forecast output data.

4 Simulation studies and results

The simulation studies include standard RNN, LSTM, GRU, and the proposed MPS for benchmarking. The system consists of the following standard parameters throughout to ensure consistency of the different type of neural networks:

- Hidden layer of four blocks/neurons.
- Batch size of 1.
- Epoch (system iteration) of 50 and 100.
- Training data of 75% (9 months).
- Testing data of 25% (3 months).

For the MPS, the window size was 48. These systems were simulated using Keras library in Python language [39]. The features of the data include the time stamp, periods, and power generated.

4.1 Original data plots

Fig. 13 shows the plots of original data taken from EMC [31] between 3-5-2015 and 24-4-2016. The data was used as simulated real-time data for the simulation studies.

4.2 Multi-GRU prediction system

4.2.1 MPS: RM: Table 1 shows the percentage results from the testing data for MPS's RM.

Table 1 shows the lowest MAPE result for 50 epochs was 0.1181 and 0.1083% for 100 epochs. MPS's RM system with MFM method in 100 epochs was shown to be the optimised combination. Fig. 14 shows the plot of the MPS's RM prediction results.

4.2.2 MPS: RTSM: Table 2 shows the percentage results from the testing data for MPS's RTSM.

Table 2 shows the lowest MAPE result for 50 epochs was 0.0477 and 0.1077% for 100 epochs. MPS's RTSM system with MFM method in 50 epochs was shown to be the optimised combination. Fig. 15 shows the plot of the MPS's RTSM prediction results.

4.2.3 MPS: MBBM: Table 3 shows the percentage results from the testing data for MPS's MBBM.

Table 3 shows the lowest MAPE result for 50 epochs was 0.3036 and 0.2109% for 100 epochs. MPS's MBBM system with

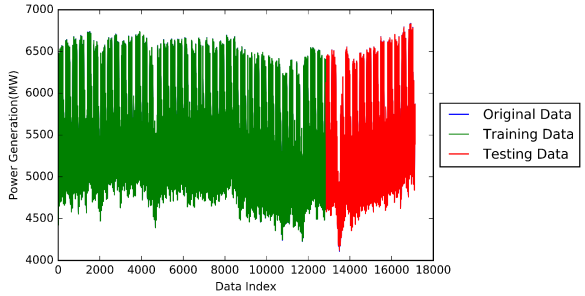

Fig. 15 *MPS's RTSM: 50 epochs MFM*

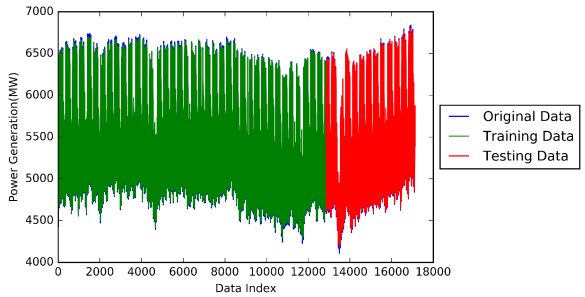

Fig. 16 *MPS's MBBM: 100 epochs MFM*

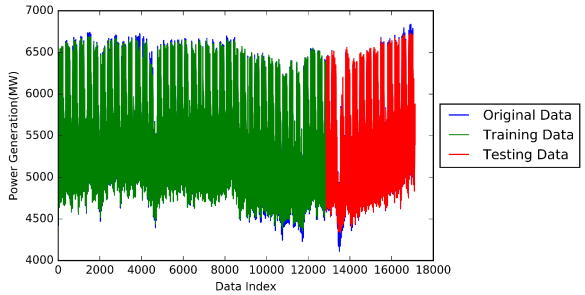

Fig. 17 *MPS's SM: 100 epochs MFM*

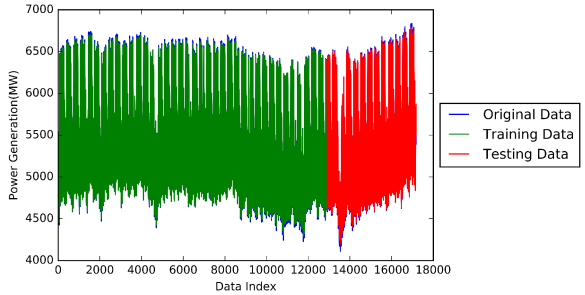

Fig. 18 *RNN: 50 epochs MFM*

MFM method in 100 epochs was shown to be the optimised combination. Fig. 16 shows the plot of the MPS's MBBM prediction results.

4.2.4 MPS: SM: Table 4 shows the percentage results from the testing data for MPS's SM.

Table 4 shows the lowest MAPE result for 50 epochs was 1.5345 and 0.947% for 100 epochs. MPS's SM system with MFM method in 100 epochs was shown to be the optimised combination. Fig. 17 shows the plot of the MPS's SM prediction results.

5 Discussion

This section discusses the performance of MPS. The performance of the MPS was benchmarked with the standard RNN, LSTM, and GRU.

Table 2 Percentage results of MPS's RTSM

MPS's RTSM system	SIFM	DIFM	MFM	SIFM	DIFM	MFM
epoch		50			100	
runtime, s	7900	7900	13,900	15,800	15,800	27,800
MAPE, %	0.4139	0.4155	0.0477	0.4058	0.4057	0.1077
max, %	3.331	3.4241	0.235	3.6335	3.6626	0.5161
min, %	0.0004	0.00007	0.00001	0.0005	0.0003	0.00132

Table 3 Percentage results of MPS's MBBM

MBBM system	SIFM	DIFM	MFM	SIFM	DIFM	MFM
epoch		50			100	
runtime, s	7700	7900	14,450	15,400	15,800	28,900
MAPE, %	0.6624	0.766	0.3036	1.0809	1.467	0.2109
max, %	4.6371	5.6524	1.7427	5.8835	6.5384	0.7765
min, %	0.00006	0.00045	0.00006	0.00002	0.0006	0.00007

Table 4 Percentage results of MPS's SM

MPS's SM system	SIFM	DIFM	MFM	SIFM	DIFM	MFM
epoch		50			100	
runtime, s	14,550	14,350	26,900	29,100	28,700	53,800
MAPE, %	1.5978	2.9055	1.5345	1.3758	1.1164	0.947
max, %	5.6262	9.5328	4.8746	6.4143	6.3706	6.384
min, %	0.00272	0.06286	0.00093	0.00098	0.00087	0.00021

Table 5 Percentage results of RNN

RNN system	SIFM	DIFM	MFM	SIFM	DIFM	MFM
epoch		50			100	
runtime, s	450	450	500	900	900	1000
MAPE, %	1.6059	1.3982	0.1677	1.3704	1.3713	0.3136
max, %	5.7412	5.6232	2.0779	5.8223	5.8245	3.41
min, %	0.0016	0.001	0.0001	0	0.0002	0.00003

Table 6 Percentage results of LSTM

LSTM system	SIFM	DIFM	MFM	SIFM	DIFM	MFM
epoch		50			100	
runtime, s	1000	1050	1000	2000	2050	2000
MAPE, %	1.3171	1.3181	0.2697	1.2929	1.2933	0.115
max, %	6.4324	6.4422	1.4077	6.0699	6.0534	0.3818
min, %	0.0006	0.0007	0.00004	0.000008	0.0002	0.00009

5.1 Standard RNN

Table 5 shows the percentage results from the testing data for standard RNN.

Table 5 shows the lowest MAPE result for 50 epochs was 0.1677 and 0.3136% for 100 epochs. RNN system with MFM method in 50 epochs was shown to be the optimised combination. Fig. 18 shows the plot of the RNN prediction results.

5.2 Standard LSTM

Table 6 shows the percentage results from testing data for standard LSTM.

Table 6 shows the lowest MAPE result for 50 epochs was 0.2697 and 0.115% for 100 epochs. LSTM system with MFM method in 100 epochs was shown to be the optimised combination. Fig. 19 shows the plot of the LSTM prediction results.

5.3 Standard GRU

Table 7 shows the percentage results from the testing data for standard GRU.

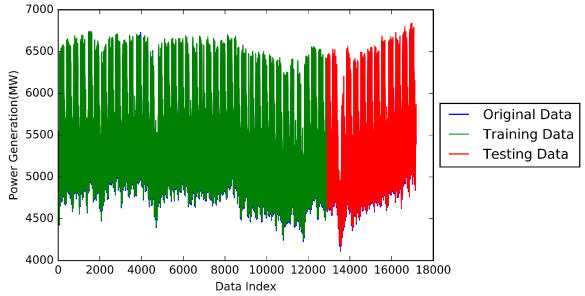

Fig. 19 *LSTM: 100 epochs MFM*

Table 7 Percentage results of GRU

GRU system	SIFM	DIFM	MFM	SIFM	DIFM	MFM
epoch		50			100	
runtime, s	800	800	800	1600	1600	1600
MAPE, %	1.3102	1.3102	0.2913	1.2908	1.291	0.1606
max, %	5.946	5.9523	0.8549	6.1358	6.1409	2.069
min, %	0.0003	0.0002	0.0001	0.00006	0.0004	0.00008

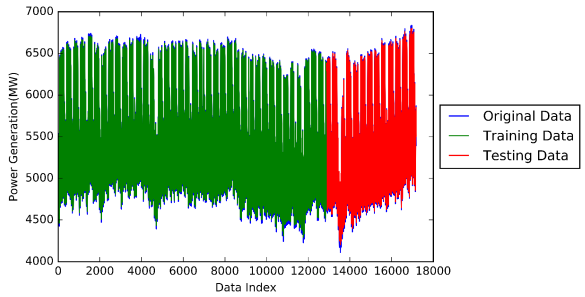

Fig. 20 *GRU: 100 epochs MFM*

Table 8 Best MAPE results

System	Best results, %	
	50 epochs	100 epochs
RNN	0.1677	0.3136
LSTM	0.2697	0.115
GRU	0.2913	0.1606
MPS's RM	0.1181	0.1083
MPS's RTSM	0.0477	0.1077
MPS's MBBM	0.3036	0.2109
MPS's SM	1.5345	0.947

Table 7 shows the lowest MAPE result for 50 epochs was 0.2913 and 0.1606% for 100 epochs. GRU system with MFM method in 100 epochs was shown to be the optimised combination. Fig. 20 shows the plot of the GRU prediction results.

Table 8 shows the best MAPE results for the different systems.

By comparing the results based on lower MAPE, RTSM has the lowest MAPE value of 0.0477% using 50 epochs and 0.1077% using 100 epochs. Although RNN has better results than most MPS models in the 50 epoch simulations, its result for 100 epochs was the worst among the forecasting systems. This validates the advantage of MPS in large designed forecast systems. By comparing both results at the end, MPS's RTSM using MFM proves to be the most optimal for the time series data of the electricity market's generation.

6 Conclusion

In this paper, an innovative MPS is proposed for electricity generation planning. A multi-features input model consists of three different features input method for the forecasting systems such as SIFM, DIFM, and multiple feature method as part of MPS. MPS includes the multi-GRU forecast model which integrates the RM, RTSM, MBBM, and SM to predict the future electricity generation. MPS uses the MAPE to determine the prediction accuracy. It presents a finding of the most optimised forecasting system such as MPS's RTSM using multiple feature method narrows down the scope on further improvement in this system for future electricity market generation forecast. With proper use of this novel prediction system, better forecasting accuracy for electricity generation planning and operation could be achieved for electricity markets.

7 References

[1] Motamedi, A., Zareipour, H., Rosehart, W.D.: 'Electricity price and demand forecasting in smart grids', *IEEE Trans. Smart Grid*, 2012, **3**, (2), pp. 664–674

[2] Li, W., Logenthiran, T., Woo, W.: 'Intelligent multi-agent system for smart home energy management'. 2015 IEEE Innovative Smart Grid Technologies-Asia (ISGT ASIA), Bangkok, Thailand, 2015, pp. 1–6

[3] Zhang, X., Shahidehpour, M., Alabdulwahab, A., *et al.*: 'Hourly electricity demand response in the stochastic day-ahead scheduling of coordinated electricity and natural gas networks', *IEEE Trans. Power Syst.*, 2016, **31**, (1), pp. 592–601

[4] Muratori, M., Rizzoni, G.: 'Residential demand response: dynamic energy management and time-varying electricity pricing', *IEEE Trans. Power Syst.*, 2016, **31**, (2), pp. 1108–1117

[5] Ferdavani, A.K., Hooshmand, R.A., Gooi, H.B.: 'Analytical solution for demand contracting with forecasting-error analysis on maximum demands and prices', *IET Gener. Transm. Distrib.*, 2018, **12**, (12), pp. 3097–3105

[6] Hayes, B.P., Gruber, J.K., Prodanovic, M.: 'Multi-nodal short-term energy forecasting using smart meter data', *IET Gener. Transm. Distrib.*, 2018, **12**, (12), pp. 2988–2994, doi: 10.1049/iet-gtd.2017.1599

[7] Yadav, A., Peesapati, R., Kumar, N.: 'Electricity price forecasting and classification through wavelet–dynamic weighted PSO–FFNN approach', *IEEE Syst. J.*, 2017, **12**, (4), pp. 3075–3084, doi: 10.1109/JSYST.2017.2717446

[8] Amjady, N., Hemmati, M.: 'Energy price forecasting-problems and proposals for such predictions', *IEEE Power Energy Mag.*, 2006, **4**, (2), pp. 20–29

[9] Li, W., Logenthiran, T., Woo, W., *et al.*: 'Implementation of demand side management of a smart home using multi-agent system'. IEEE World Congress on Computational Intelligence, Vancouver, Canada, 2016, pp. 1–8

[10] Maciejowska, K., Weron, R.: 'Short-and mid-term forecasting of baseload electricity prices in the UK: The impact of intra-day price relationships and market fundamentals', *IEEE Trans. Power Syst.*, 2016, **31**, (2), pp. 994–1005

[11] Alamaniotis, M., Bargiotas, D., Bourbakis, N.G., *et al.*: 'Genetic optimal regression of relevance vector machines for electricity pricing signal forecasting in smart grids', *IEEE Trans. Smart Grid*, 2015, **6**, (6), pp. 2997–3005

[12] Voronin, S., Partanen, J.: 'Price forecasting in the day-ahead energy market by an iterative method with separate normal price and price spike frameworks', *Energies*, 2013, **6**, (11), pp. 5897–5920

[13] Taylor, J.W., McSharry, P.E.: 'Short-term load forecasting methods: an evaluation based on European data', *IEEE Trans. Power Syst.*, 2007, **22**, (4), pp. 2213–2219

[14] Elattar, E.E., Goulermas, J.Y., Wu, Q.H.: 'Electric load forecasting based on locally weighted support vector regression', *IEEE Trans. Syst. Man Cybern. C*, 2010, **40**, (4), pp. 438–447

[15] Thouvenot, V., Pichavant, A., Goude, Y., *et al.*: 'Electricity forecasting using multi-stage estimators of nonlinear additive models', *IEEE Trans. Power Syst.*, 2016, **31**, (5), pp. 3665–3673

[16] Rafiei, M., Niknam, T., Khooban, M.H.: 'Probabilistic forecasting of hourly electricity price by generalization of ELM for usage in improved wavelet neural network', *IEEE Trans. Ind. Inf.*, 2017, **13**, (1), pp. 71–79

[17] Han, L., Romero, C.E., Wang, X., *et al.*: 'Economic dispatch considering the wind power forecast error', *IET Gener. Transm. Distrib.*, 2018, **12**, (12), pp. 2861–2870, doi: 10.1049/iet-gtd.2017.1638

[18] Zeng, P., Jin, M.: 'Peak load forecasting based on multi-source data and day-to-day topological network', *IET Gener. Transm. Distrib.*, 2017, **12**, (6), pp. 1374–1381

[19] Grant, J., Eltoukhy, M., Asfour, S.: 'Short-term electrical peak demand forecasting in a large government building using artificial neural networks', *Energies*, 2014, **7**, (4), pp. 1935–1953

[20] Zhao, R., Wang, D., Yan, R., *et al.*: 'Machine health monitoring using local feature-based gated recurrent unit networks', *IEEE Trans. Ind. Electron.*, 2018, **65**, (2), pp. 1539–1548

[21] Hoiles, W., Krishnamurthy, V.: 'Nonparametric demand forecasting and detection of energy aware consumers', *IEEE Trans. Smart Grid*, 2015, **6**, (2), pp. 695–704

[22] Shrivastava, N.A., Panigrahi, B.K.: 'Prediction interval estimations for electricity demands and prices: a multi-objective approach', *IET Gener. Transm. Distrib.*, 2015, **9**, (5), pp. 494–502

[23] Xu, X., Niu, D., Wang, Q., *et al.*: 'Intelligent forecasting model for regional power grid with distributed generation', *IEEE Syst. J.*, 2017, **11**, (3), pp. 1836–1845

[24] Anbazhagan, S., Kumarappan, N.: 'Day-ahead deregulated electricity market price forecasting using recurrent neural network', *IEEE Syst. J.*, 2013, **7**, (4), pp. 866–872

[25] Al-Qahtani, F.H., Crone, S.F.: 'Multivariate k-nearest neighbour regression for time series data – a novel algorithm for forecasting UK electricity demand'. The 2013 Int. Joint Conf. on Neural Networks (IJCNN), Dallas, TX, USA, 2013, pp. 1–8

[26] Erdener, B.C., Pambour, K.A., Lavin, R.B., *et al.*: 'An integrated simulation model for analysing electricity and gas systems', *Int. J. Electr. Power Energy Syst.*, 2014, **61**, pp. 410–420
[27] Roh, J.H., Shahidehpour, M., Wu, L.: 'Market-based generation and transmission planning with uncertainties', *IEEE Trans. Power Syst.*, 2009, **24**, (3), pp. 1587–1598
[28] El-Khattam, W., Hegazy, Y., Salama, M.: 'An integrated distributed generation optimization model for distribution system planning', *IEEE Trans. Power Syst.*, 2005, **20**, (2), pp. 1158–1165
[29] El-Khattam, W., Bhattacharya, K., Hegazy, Y., *et al.*: 'Optimal investment planning for distributed generation in a competitive electricity market', *IEEE Trans. Power Syst.*, 2004, **19**, (3), pp. 1674–1684
[30] Mill, S.: 'Electric load forecasting: advantages and challenges', 2016. Available at http://engineering.electrical-equipment.org/electrical-distribution/electric-load-forecasting-advantages-challenges.html
[31] Company, E.M.: 'Market trading reports', 2016. Available at https://www.emcsg.com/marketdata/markettradingreports
[32] Elman, J.L.: 'Distributed representations, simple recurrent networks, and grammatical structure', *Mach. Learn.*, 1991, **7**, (2–3), pp. 195–225
[33] Olah, C.: 'Understanding LSTM networks', 2015. Available at http://colah.github.io/posts/2015-08-Understanding-LSTMs/
[34] Hochreiter, S., Schmidhuber, J.: 'Long short-term memory', *Neural Comput.*, 1997, **9**, (8), pp. 1735–1780
[35] Britz, D.: 'Recurrent neural network tutorial, part 4 – implementing a GRU/LSTM RNN with python and theano', 2015. Available at http://www.wildml.com/2015/10/recurrent-neural-network-tutorial-part-4-implementing-a-grulstm-rnn-with-python-and-theano/
[36] Chung, J., Gulcehre, C., Cho, K., *et al.*: 'Empirical evaluation of gated recurrent neural networks on sequence modeling', arXiv preprint arXiv:14123555, 2014
[37] Gomez, A.: 'Backpropogating an LSTM: a numerical example', 2016. Available at https://medium.com/@aidangomez/let-s-do-this-f9b699de31d9
[38] Seo, J.D.: 'Only numpy: deriving forward feed and back propagation in gated recurrent neural networks (GRU)–empirical evaluation of gated recurrent neural networks on sequence modeling – part 1', 2018. Available at https://medium.com/swlh/only-numpy-deriving-forward-feed-and-back-propagation-in-gated-recurrent-neural-networks-gru-8b6810f91bad
[39] Chollet, F.: 'Keras'. GitHub, 2015. Available at https://github.com/fchollet/keras