energies

MDPI

Article

Integrating Long Short-Term Memory and Genetic Algorithm for Short-Term Load Forecasting

Arpita Samanta Santra [1] and Jun-Lin Lin [1,2,*]

[1] Department of Information Management, Yuan Ze University, Taoyuan 32003, Taiwan; santraarpita83@gmail.com

[2] Innovation Center for Big Data and Digital Convergence, Yuan Ze University, Taoyuan 32003, Taiwan

* Correspondence: jun@saturn.yzu.edu.tw; Tel.: +886-3-463-8800 (ext. 2611)

Received: 18 April 2019; Accepted: 25 May 2019; Published: 28 May 2019

Abstract: Electricity load forecasting is an important task for enhancing energy efficiency and operation reliability of the power system. Forecasting the hourly electricity load of the next day assists in optimizing the resources and minimizing the energy wastage. The main motivation of this study was to improve the robustness of short-term load forecasting (STLF) by utilizing long short-term memory (LSTM) and genetic algorithm (GA). The proposed method is novel: LSTM networks are designed to avoid the problem of long-term dependencies, and GA is used to obtain the optimal LSTM's parameters, which are then applied to predict the hourly electricity load for the next day. The proposed method was trained using actual load and weather data, and the performance results showed that it yielded small mean absolute percentage error on the test data.

Keywords: long short term memory (LSTM); genetic algorithm (GA); short term load forecasting (STLF); electricity load forecasting; multivariate time series

1. Introduction

Electricity load forecasting is a mandatory procedure in the capacity planning process of the power industry. Three types of load forecasting are categorized with different time scales. Medium-term and long-term load forecastings predict the weekly, monthly, or yearly electricity load, and both are necessary for system planning, resource investment, and budget allocation. Short-term load forecasting (STLF), the focus of this study, predicts hourly or daily electricity load one hour to one week ahead, which is crucial for resource planning and load balancing in power system management [1–3].

Imprecise load forecasting increases operating cost. For example, an increase of only 1% in forecast error caused an increase of 10 million pounds in operating cost per year for an electric utility company in the United Kingdom [4–6]. Due to the economic and the environmental concerns, electricity load forecasting has drawn considerable attention from both the academic and industrial field. Many approaches have been proposed to solve the STLF problem. For example, many conventional approaches for time series prediction, such as statistical analysis, regression methods [7], smoothing techniques, stochastic process and autoregressive moving average (ARMA) models [8], have been applied to the STLF problem. In last few decades, many data mining approaches have achieved good performance on tackling the uncertainties in STLF, including artificial neural networks (ANN) [9,10], fuzzy inference systems, neuro-fuzzy systems [11], support vector machines, and artificial immune systems [12], etc. Among these data mining approaches, ANN is one of the most popular methods. ANN is suitable for nonlinear problems, has excellent learning ability, and is robust to noise in data.

For time series prediction problems, as in the case of STLF, a special type of ANN called recurrent neural network (RNN) is often used to handle the time-dependency property in time series data. RNN greatly reduces the number of nodes (and consequently, the number of parameters to be learned) in

the neural network, but it suffers from the vanishing gradient and long-term dependency problems. To overcome the drawbacks of the traditional RNN, Hochreiter et al. [13] proposed long short-term memory (LSTM) RNN, which added more controls to the traditional RNN to retain both short-term and long-term memory in the network. Since then, LSTM-based networks have become one of the most promising technologies for deep learning, and have been successfully applied in many research areas such as natural language translation [14], image captioning [15–17], speech recognition [18], and handwriting recognition [19].

With LSTM, the initial weighting of each link in the neural network must be determined before the training iterations start. However, poorly chosen initial weightings could sometimes lead to a bad performance. The motivation of this study was to propose a method such that LSTM could start with a set of properly chosen initial weightings to reduce the forecasting error of the STLF problem. Specifically, the proposed method integrated the searching capability of genetic algorithm (GA) and the learning capability of LSTM. GA is responsible for searching the optimal initial weightings through its population-based bio-inspired evolution operations [20] and LSTM is responsible for learning and memorization intelligence from power usage data and weather data. Overall, the proposed method and the underlying STLF problem can be summarized as follows:

- Input: Weather data such as temperature and humidity are closely related to electricity consumption. Thus, this study used hourly weather data and electricity load for the past 24 h as input in the training process.
- Output: The objective of this study was to produce the hourly electricity load for the next day.
- Methodology framework: The proposed method integrated GA and LSTM. GA was implemented to optimize the parameters of LSTM; LSTM was employed to learn from the past data to yield a better prediction.

The rest of the paper is organized as follows: Section 2 surveys the literature on STLF, and Section 3 describes LSTM for STLF. Section 4 presents our proposed approach. Sections 5 and 6 describe the settings and the results of the performance study, respectively. Section 7 concludes this paper, and gives direction for future research.

2. Related Works on Short-Term Load Forecasting

The problem of load forecasting has attracted much attention since the mid-1960s [21,22]. This section focuses the techniques for STLF. STLF is essentially a time series problem, and thus many traditional time series prediction techniques have been used to solve this problem, e.g., ARMA [23], ARIMA [24], and a hybrid of ARIMA and SVM [25]. Under normal conditions, these statistical techniques deliver good prediction results. However, sudden changes of the weather conditions can dramatically affect the short-term power usage patterns, and consequently, render these statistical techniques failing to provide accurate prediction.

Due to the impact of weather on short-term power usage, many studies have factored into the weather conditions for STLF. Reference [26] applied fuzzy logic [27] to find rules to handle similar weather conditions. Reference [28] used wavelet transform to decompose the electrical load data into components of various frequencies, and then built a SVM model based on the temperature data and the low-frequency components of the electricity load data. Reference [29] built two regression models to respectively predict daily and hourly loads based on the weather data and the electrical load data. Reference [30] applied ANN for long-term load forecasting, and their results showed that ANN is superior to linear regression and SVM.

ANN is a widely used technique for learning nonlinear patterns. Because STLF is a nonlinear problem, many previous works have utilized ANN for STLF, e.g., Reference [31] applied a feed-forward backpropagation ANN for STLF. Hybrids of ANN and other techniques are also common for STLF, e.g., regression tree model [32], time series analysis [33], genetic algorithm [34], chaos genetic algorithm and simulated annealing [35].

With the advance of computing power, deep neural network (DNN) has gained much popularity and been applied to STLF in recent years [36]. LSTM is a special type of DNN that is suitable for time series prediction due to its capability of remembering both the short-term and the long-term behavior in time series data. In Reference [37], two types of LSTM were compared against other deep learning techniques for predicting electricity load of every hour or every minute. Their results showed that LSTM outperformed the other techniques. However, all of the neural network approaches above require setting up the initial weightings of the links in the network, but poorly chosen weightings could lead the searching process trapped in the local optimum. A hybrid of LSTM and GA is proposed in Section 4 to resolve this problem.

3. Long Short-Term Memory for Short-term Load Forecasting

Prior to presenting our approach in Section 4, this section gives a detailed description of LSTM. LSTM is an augmented recurrent neural network model. It learns sequential information with long term dependencies, and preserves information for a long period of time. Traditional recurrent neural network suffers from the vanishing gradient problem. That is, as the number of layers using the same activation function increases, the gradients of the loss function approaches zero, making it difficult to train the network through backpropagation of errors. To prevent the vanishing gradient problem, LSTM utilizes memory cells, where each cell maintains a cell state and a hidden cell state, and uses three gates (namely, input gate, output gate, and forget gate) to control the flow of information into or out of the cell. A formal explanation of the LSTM model is given below.

LSTM is for time series modeling, which maps an input sequence $x = \{x_1, x_2, \ldots, x_n\}$ to an output sequence $y = \{y_1, y_2, \ldots, y_n\}$. For the STLF problem under study, each x_i represents the hourly electricity load and the weather data of day i, and each y_i represents the hourly electricity load of day $i + 1$, indicating a look-ahead parameter of value one. The LSTM contains layers of memory cells, where the interaction between the LSTM layers is shown in Figure 1, and the architecture of a LSTM cell is shown in Figure 2.

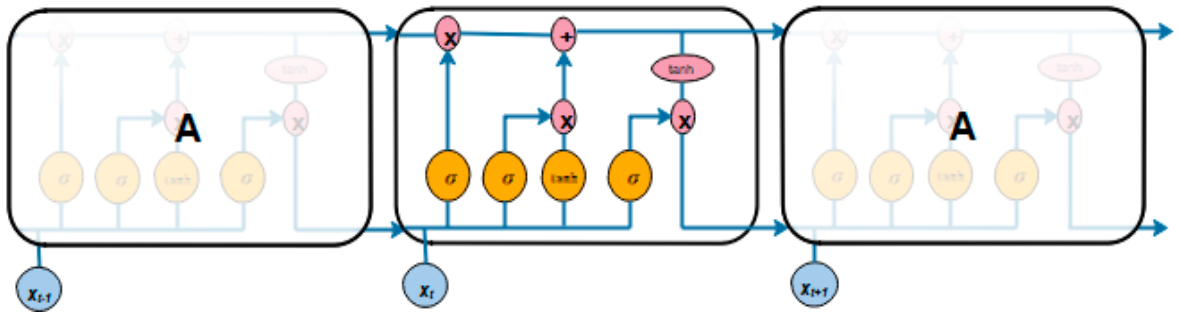

Figure 1. The interaction between Long Short-Term Memory layers.

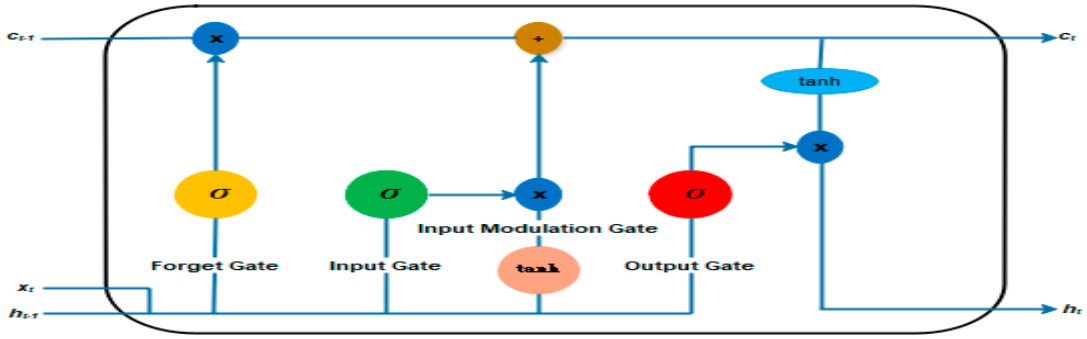

Figure 2. The architecture of the Long Short-Term Memory cell.

For ease of exploration, cell t refers to the LSTM cell at the t-th layer in the network. As shown in Figure 2, the input for cell t included x_t, h_{t-1} and c_{t-1}, where x_t was from the input sequence x, and h_{t-1} and c_{t-1} were from the output of cell $t-1$. Directed weighted links connected various components within a LSTM cell or between two adjacent LSTM cells.

The input gate (i_t) of cell t used the Logistic sigmoid function $\sigma(\cdot)$ to decide whether to store the current input x_t and the new cell state in the memory, as shown in Equation (1), where W_{ix}, W_{ih} and W_{ic} were the weights of the incoming links associated with the input gate, and b_i was the bias input of the input gate.

$$i_t = \sigma(W_{ix}x_t + W_{ih}h_{t-1} + W_{ic}c_{t-1} + b_i) \tag{1}$$

Similarly, the forget gate (f_t) of cell t used the Logistic sigmoid function to decide whether to remove the previous cell state (c_{t-1}) from the memory, as shown in Equation (2), where W_{fx}, W_{fh} and W_{fc} were the weights of the incoming links associated with the forget gate, and b_f was the bias input of the forget gate.

$$f_t = \sigma\left(W_{fx}x_t + W_{fh}h_{t-1} + W_{fc}c_{t-1} + b_f\right) \tag{2}$$

The new cell state (c_t) was determined by the amount of old cell state to forget (i.e., $f_t \circ c_{t-1}$) and the amount of new information to include (i.e., $i_t \circ \tanh(W_{cx}x_t + W_{ch}h_{t-1} + b_c)$), as shown in Equation (3), where W_{cx} and W_{ch} were the weights of the incoming links associated with the cell state, b_c was the bias input of the cell state, and $\circ$ represented the hadamard product.

$$c_t = f_t \circ c_{t-1} + i_t \circ \tanh(W_{cx}x_t + W_{ch}h_{t-1} + b_c) \tag{3}$$

The output gate (o_t) used the Logistic sigmoid function to filter information from the current input x_t, previous cell state c_{t-1}, and previous hidden state h_{t-1}, and as shown in Equation (4), where W_{ox}, W_{oh} and W_{oc} were the weights of the incoming links associated with the output gate, and b_o was the bias input of the output gate.

$$o_t = \sigma(W_{ox}x_t + W_{oh}h_{t-1} + W_{oc}c_{t-1} + b_o) \tag{4}$$

The new hidden state (h_t) was calculated as the hadamard product between the output gate values o_t and the tan h function value of the current state c_t, as shown in Equation (5).

$$h_t = o_t \circ \tanh(c_t) \tag{5}$$

Finally, the output of the memory cell (i.e., the predicted value of y_t) was calculated from the hidden state cell state h_t, as shown in Equation (6), where W_{yh} were the weights of the incoming links associated with the hidden state and b_y was the bias input of the hidden state.

$$\hat{y}_t = \left(W_{yh}h_t + b_y\right) \tag{6}$$

For the STLF problem under study, y_t and $\hat{y}_t$ represented the actual and the predicted hourly electricity load of a day, respectively. Thus, both y_t and $\hat{y}_t$ contained 24 values, one for each hour. The predicted error (i.e., $y_t - \hat{y}_t$) was calculated as the mean absolute percentage error (MAPE) of the 24 pairs of corresponding values in y_t and $\hat{y}_t$. The predicted error at time step t was back propagated to refine the weighting matrices in the LSTM. The objective of training the LSTM was to refine the weighting matrices to minimize the predicted error.

Notably, the LSTM contained only one LSTM cell layer, and Figure 1 actually depicts how the cell layer unfolds over time. That is, cell t refers to the status of the LSTM cell at time step t (or day t for the STLF problem under study). Thus, LSTM only maintained a set of weighting matrices and a bias vector for all gates and memory states.

4. Integration of Long Short-Term Memory and Genetic Algorithm: The Proposed Method

As described earlier, the initial values of the weighting matrices could affect the performance of the LSTM. Our proposed method used GA to assist searching the proper initial values for the weighting matrices of the LSTM. GA is a population-based searching technique that employs a population of chromosomes in the searching process. Each chromosome represents a feasible solution. For LSTM, a feasible solution consists of the values of all the weighting matrices described in Section 3. Figure 3 shows the flow diagram of the proposed method. The main steps of the proposed method are described in detail below.

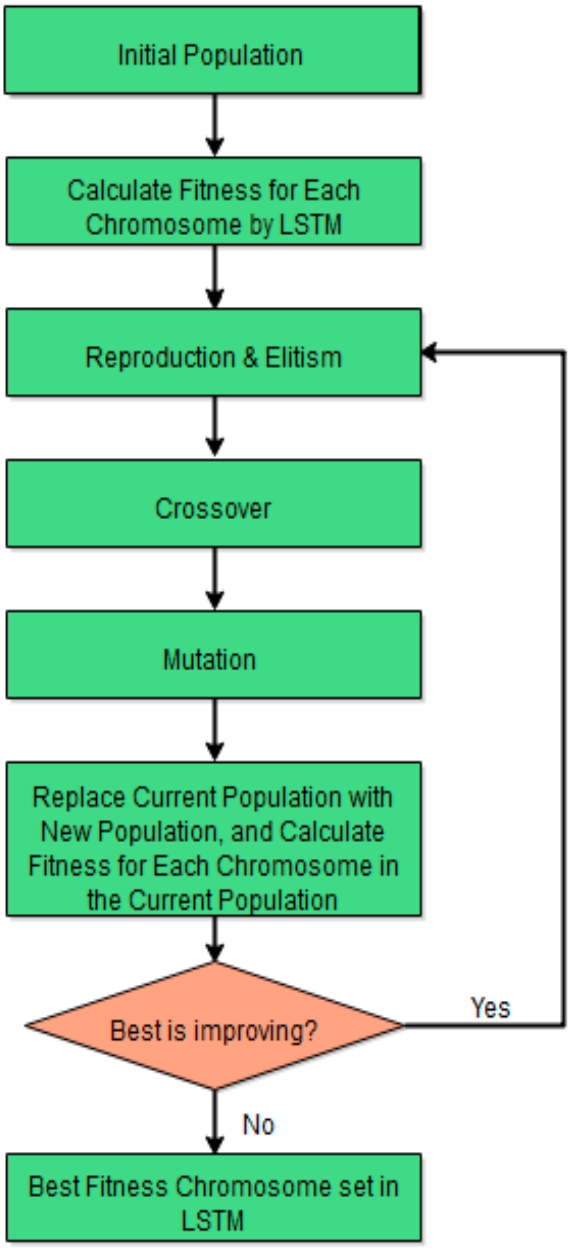

Figure 3. The flow diagram of the proposed method.

Step 1: Generate initial population of chromosomes.

A set of n chromosomes were randomly generated. Each chromosome W contained the values for all the weighting matrices in LSTM, i.e., $W = [W_{ix}\ W_{ih}\ W_{fx}\ W_{fh}\ W_{cx}\ W_{ch}\ W_{ox}\ W_{oh}\ W_{yh}]$.

Step 2: Calculate fitness for each chromosome via LSTM.

For each chromosome in the current population, its value was used to initialize the weighting matrices in the LSTM. Then, training data was fed into the LSTM so that the LSTM could learn from the data to adjust the value of W to minimize the mean absolute percentage error (MAPE) of the training data, as is done in the traditional LSTM technique. The fitness of this chromosome was the final MAPE value of the training data.

Step 3: Generate the new population via genetic operations.

This step generated a new population that contained the same number of chromosomes as the current population. The chromosomes of the new population were generated by applying genetic operations (i.e., reproduction, crossover, or mutation) on the chromosomes selected from the current population. This study used the roulette wheel selection so that a chromosome with a higher fitness value had a higher probability of being selected for genetic operations. In the new population, the proportions of the chromosomes generated via the reproduction, crossover, and mutation operations were referred to as the reproduction, crossover, and mutation ratios, respectively. GA usually adopts a large crossover ratio, and a low mutation ratio.

The reproduction operation repeatedly selected a chromosome from the current population, and added it to the new population, until the reproduction ratio was reached. This study applied an elitism policy so that the best chromosome in the current chromosome was always added to the new population.

The crossover operation repeatedly selected two chromosomes from the current population acting as the parent chromosomes to generate and add two offspring chromosomes to the new population, until the crossover ratio was reached. Uniform crossover was adopted in this study. With uniform crossover, each value in the offspring chromosomes is independently chosen from the two values at the same corresponding position in the two parent chromosomes, as shown in Figure 4.

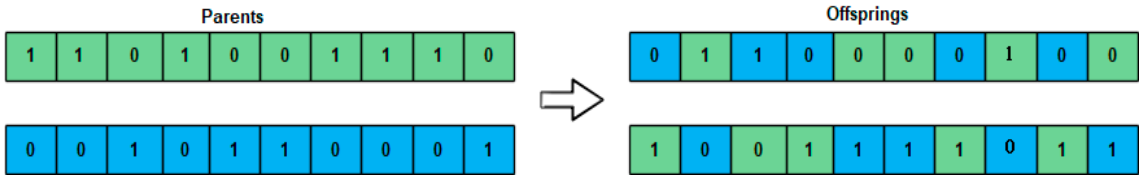

Figure 4. Uniform crossover.

The mutation operation repeatedly selected a chromosome from the current population, modified the selected chromosome to generate a new chromosome (referred to as a mutant), and added the mutant to the new population, until the mutation ratio had reached. One-point mutation was adopted in this study. With one-point mutation, a small random change is injected into the value of a randomly selected position in the selected chromosome to generate the mutant, as shown in Figure 5. The mutation operation introduced diversity to the population of chromosomes

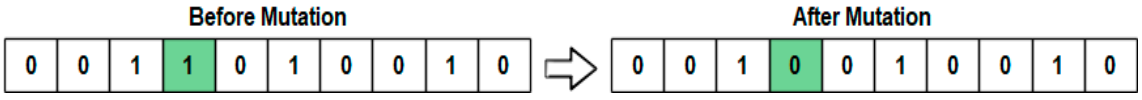

Figure 5. One-point mutation.

Step 4: Replace population and calculate fitness.

At this step, the current population could be abandoned, and the new population became the current population. Similar to Step 2, each chromosome in the current population was used to initialize the weighting matrices in a LSTM, and the training data was fed into the LSTM to yield the fitness value of this chromosome.

Step 5: Stop criterion

If the fitness value of the best chromosome did not improve for s continuous generations, then the LSTM of the best chromosome was adopted, and the proposed method was terminated. Otherwise, go to Step 3 to generate a new population. In this study, s was set to 50.

5. Experiment Settings

To evaluate the performance of the proposed method, a performance study was conducted on a desktop computer with Intel Core2 Duo E7500@2.94 GHz CPU, 64-bit Windows 10 operating system and 4 GB memory (RAM) using Python 3.6.6. The performance study used hourly electricity load data obtained from the Australian Energy Market Operator, and hourly weather data of the Sydney Observatory obtained from the Bureau of Meteorology (Australia). The timespan of the dataset was from 1 January 2006 to 1 January 2011. The first three years of the dataset was used for training, and the final year was used for testing purposes. The dataset was normalized before being used for training and testing. Because the hourly electricity load patterns were quite different between the weekend and weekday, this study focused only on weekday data.

Each record in the dataset contained the electricity load, dry bulb temperature, dew point temperature, wet bulb temperature, and humidity for every hour within one day. Figure 6 shows that the input of the LSTM network was a record (for day i), and the output was the hourly electricity load part of the next record (for day $i + 1$). Thus, the LSTM network had 120 inputs and 24 outputs. The parameter settings of the proposed method were as follows: The number of hidden layers in the LSTM network = 50, the number of epochs for LSTM = 400, population size = 10, crossover ratio = 0.8, and mutation ratio = 0.003.

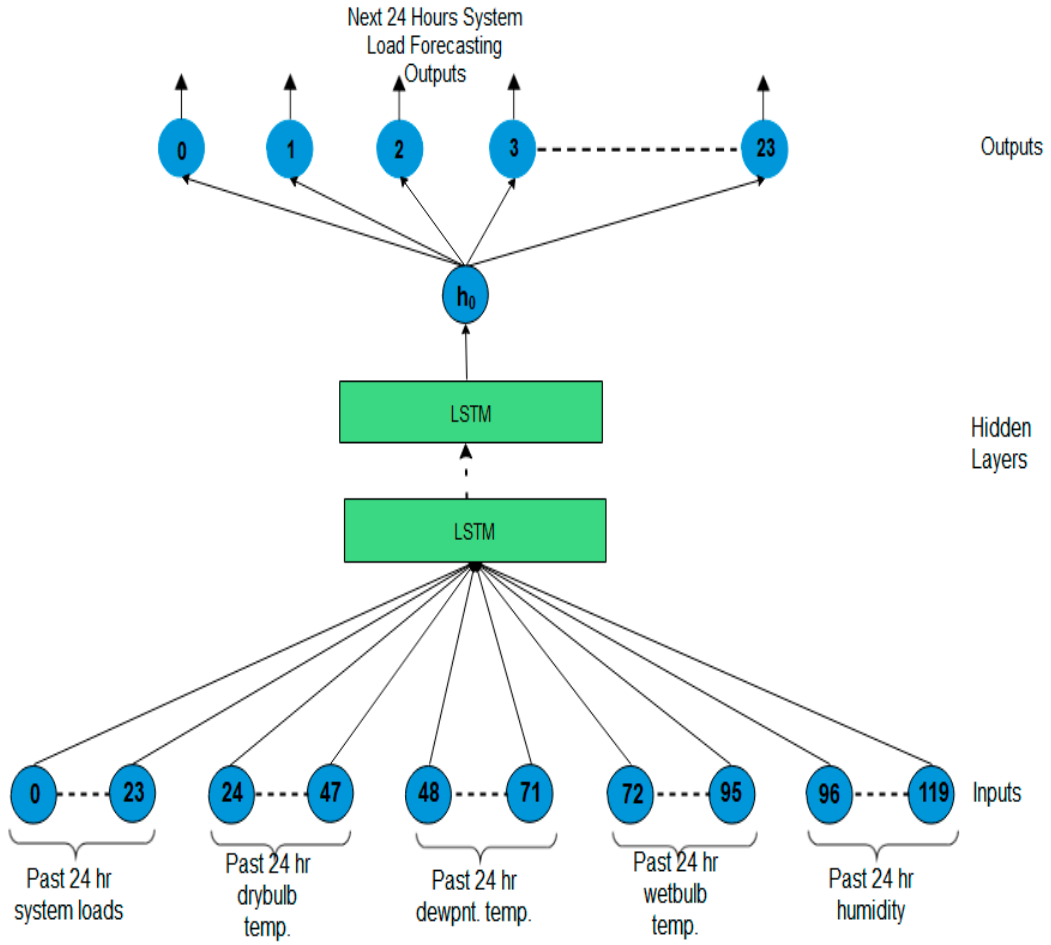

Figure 6. Input and output of the Long Short-Term Memory network for Short-Term Load Forecasting.

6. Experimental Results

In the experimental results, we compared the performance of the proposed method against the LSTM. The forecast results for five randomly selected working days and one randomly selected week in the testing data is plotted in Figure 7. The results showed that the prediction of the proposed methods followed very close with the actual electricity load. LSTM also yielded good results, but it was less stable and could sometimes incur large errors.

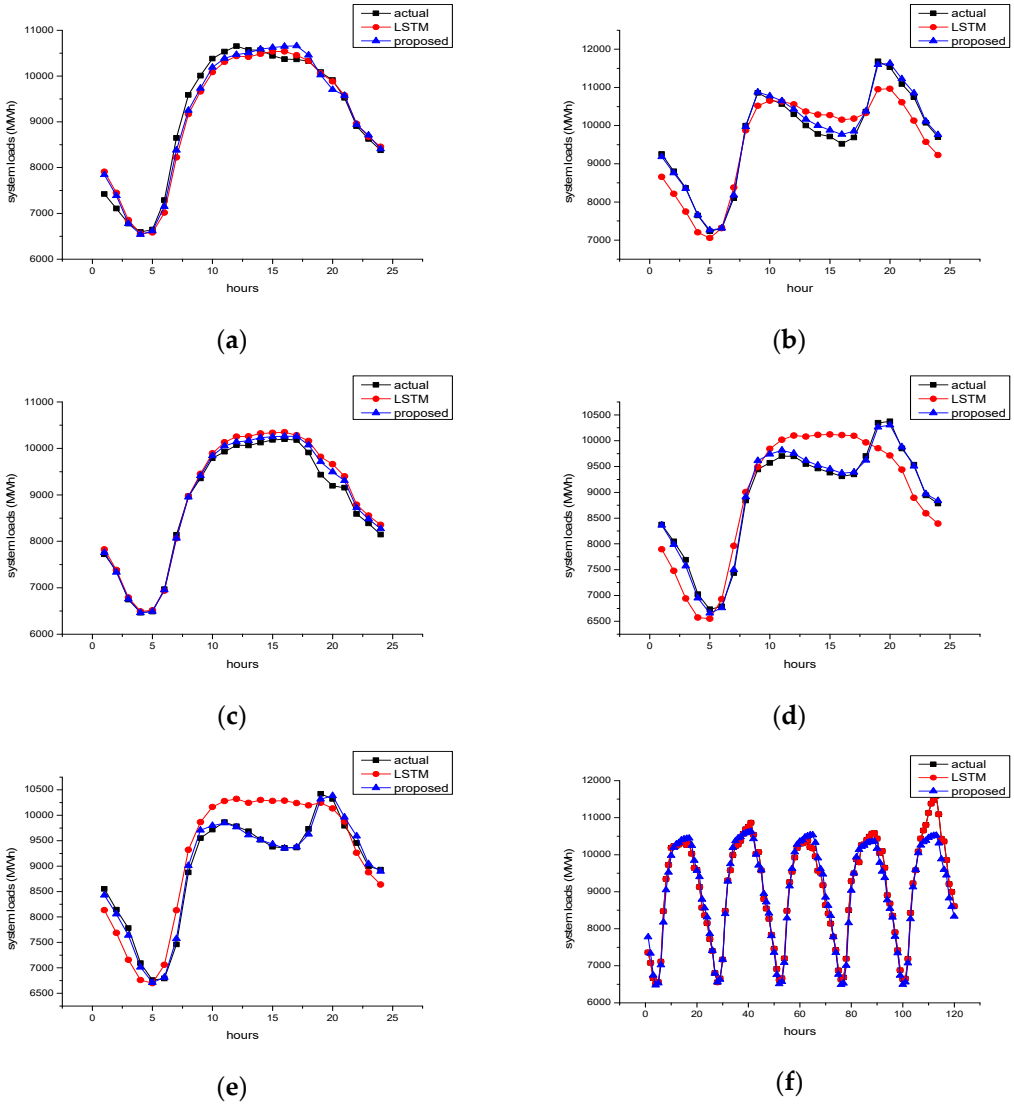

Figure 7. Forecast results for (**a**) 29 March 2010, Monday; (**b**) 8 June 2010, Tuesday; (**c**) 24 November 2010, Wednesday; (**d**) 2 September 2010, Thursday; (**e**) 3 September 2010, Friday; (**f**) the week from 22 March (Monday) to 26 March (Friday) 2010.

Table 1 compares the MAPE of the LSTM and our proposed method for the week of 15 to 19 March 2010. The proposed method consistently performed better than the LSTM and reduced the MAPE of the LSTM by 5.38% to 53.33%. Table 2 compares the MAPE of the LSTM and our proposed method for five randomly chosen days. Similar to Table 1, our proposed method consistently outperformed the LSTM.

The one-year testing data contained 250 records (or weekdays). Figure 8 shows the daily MAPE for the testing data, where the horizontal axis is time, and the vertical axis is MAPE. The results showed that the proposed method consistently improved the LSTM. Descriptive statistics of the 250 daily

MAPEs is shown in Table 3. The proposed method yielded a smaller mean and standard deviation than the LSTM, indicating that the proposed method was more accurate and stable than the LSTM.

Table 1. MAPE of LSTM and proposed method for the week of 15 to 19 March 2010.

Day of the Week	Method		
	LSTM	Proposed Method	Improvement (%)
Mon.	2.94	1.96	33.33
Tue.	1.79	1.19	33.51
Wed.	1.80	0.84	53.33
Thurs.	1.62	1.19	26.54
Fri.	1.30	1.23	5.38

Table 2. MAPE of LSTM and proposed method for five randomly chosen days in the testing data.

Day	Method		
	LSTM	Proposed Method	Improvement (%)
13 Dec. Mon.	2.88	1.84	36.11
8 July. Tue.	4.08	0.87	78.67
17 Feb. Wed.	1.52	1.20	21.05
13 May. Thurs.	3.59	0.86	76.04
27 Aug. Fri.	4.17	1.07	74.34

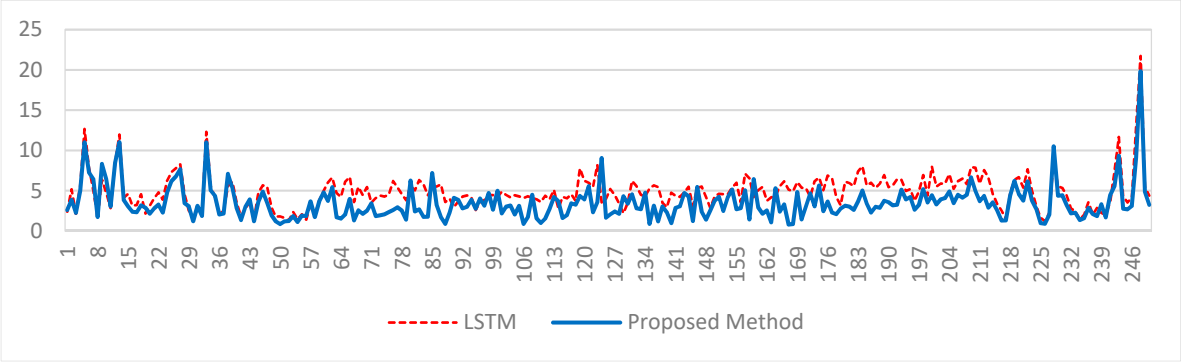

Figure 8. Daily MAPE of the LSTM and the proposed methods for the 250 weekdays in the testing data.

Table 3. Statistics of the daily MAPE for the 250 weekdays in the testing data.

	Method	
	LSTM	Proposed Method
mean	4.8469	3.4889
stdev.	2.2116	2.1506
min	1.2258	0.7879
max	21.7188	19.8419

7. Conclusions

This study proposed a new method that integrated GA and the LSTM for STLF. Although the traditional LSTM has a great learning capability for time series data, it sometimes suffers from the poorly chosen values for its initialization parameters. The proposed method mitigated this problem by applying GA to search suitable values for the initialization parameters of the LSTM. Our performance study showed that the proposed method could effectively improve the prediction accuracy of the LSTM.

This study uses MAPE to measure the error between the predictions and the actual electricity loads. This is based on the assumption that the damage or cost incurred by the error is linearly

proportional to the magnitude of the error. However, there are situations that a large prediction error causes a much larger cost or damage to the power companies. For those situations, mean squared error or other more suitable measurements should be adopted, instead of using MAPE.

The electricity demand is easily influenced by many factors. In this study, the weather was used to improve the prediction of the short-term electricity demand. Adding more related factors, e.g., economic conditions, could be of interest for further study. Other meta-heuristic searching techniques and their impact on the execution time are also worthy of investigation for STLF.

Author Contributions: Conceptualization, A.S.S. and J.-L.L.; methodology, J.-L.L.; software, A.S.S.; validation, J.-L.L.; data curation, A.S.S.; visualization, A.S.S.; writing—original draft preparation, A.S.S.; writing—review and editing, J.-L.L.; supervision, J.-L.L.; funding acquisition, J.-L.L. overall contribution: A.S.S. (50%) and J.-L.L. (50%).

Funding: This research was supported by the Ministry of Science and Technology (MOST), Taiwan, under Grant MOST 106-2221-E-155-038.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Suganthi, L.; Samuel, A.A. Energy models for demand forecasting—A review. *Renew. Sustain. Energy Rev.* **2012**, *16*, 1223–1240. [CrossRef]
2. Al-Hamadi, H.; Soliman, S. Long-term/mid-term electric load forecasting based on short-term correlation and annual growth. *Electr. Power Syst. Res.* **2005**, *74*, 353–361. [CrossRef]
3. Al-Shobaki, S.; Mohsen, M. Modeling and forecasting of electrical power demands for capacity planning. *Energy Convers. Manag.* **2008**, *49*, 3367–3375. [CrossRef]
4. Dudek, G. Artificial immune system for short-term electric load forecasting. In Proceedings of the International Conference on Artificial Intelligence and Soft Computing, Zakopane, Poland, 22–26 June 2008; pp. 1007–1017.
5. Lo, K.; Wu, Y. Risk assessment due to local demand forecast uncertainty in the competitive supply industry. *IEE Proc. Gener. Transm. Distrib.* **2003**, *150*, 573–581. [CrossRef]
6. Soares, L.J.; Medeiros, M.C. Modeling and forecasting short-term electricity load: A comparison of methods with an application to Brazilian data. *Int. J. Forecast.* **2008**, *24*, 630–644. [CrossRef]
7. Yildiz, B.; Bilbao, J.I.; Sproul, A.B. A review and analysis of regression and machine learning models on commercial building electricity load forecasting. *Renew. Sustain. Energy Rev.* **2017**, *73*, 1104–1122. [CrossRef]
8. Pappas, S.S.; Ekonomou, L.; Karampelas, P.; Karamousantas, D.; Katsikas, S.; Chatzarakis, G.; Skafidas, P. Electricity demand load forecasting of the Hellenic power system using an ARMA model. *Electr. Power Syst. Res.* **2010**, *80*, 256–264. [CrossRef]
9. Ekonomou, L.; Oikonomou, D. Application and comparison of several artificial neural networks for forecasting the Hellenic daily electricity demand load. In Proceedings of the 7th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases, Cairo, Egypt, 20–22 February 2008; pp. 67–71.
10. Ekonomou, L.; Christodoulou, C.; Mladenov, V. A short-term load forecasting method using artificial neural networks and wavelet analysis. *Int. J. Power Syst* **2016**, *1*, 64–68.
11. Dudek, G. Neuro-fuzzy approach to the next day load curve forecasting. *Przegląd Elektrotechniczny* **2011**, *87*, 61–64.
12. Dudek, G. Artificial immune system with local feature selection for short-term load forecasting. *IEEE Trans. Evol. Comput.* **2017**, *21*, 116–130. [CrossRef]
13. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
14. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 3104–3112.
15. Vinyals, O.; Toshev, A.; Bengio, S.; Erhan, D. Show and tell: A neural image caption generator. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3156–3164.

16. Karpathy, A.; Fei-Fei, L. Deep visual-semantic alignments for generating image descriptions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3128–3137.
17. Mao, J.; Xu, W.; Yang, Y.; Wang, J.; Huang, Z.; Yuille, A. Deep captioning with multimodal recurrent neural networks (m-rnn). *arXiv* **2014**, arXiv:1412.6632.
18. Graves, A.; Jaitly, N. Towards end-to-end speech recognition with recurrent neural networks. In Proceedings of the International Conference on Machine Learning, Beijing, China, 21–26 June 2013; pp. 1764–1772.
19. Graves, A.; Schmidhuber, J. Offline handwriting recognition with multidimensional recurrent neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–10 December 2009; pp. 545–552.
20. Mitchell, M. *An Introduction to Genetic Algorithms*; MIT Press: Cambridge, MA, USA, 1998.
21. Khuntia, S.R.; Rueda, J.L.; van der Meijden, M.A. Forecasting the load of electrical power systems in mid-and long-term horizons: A review. *IET Gener. Transm. Distrib.* **2016**, *10*, 3971–3977. [CrossRef]
22. Hahn, H.; Meyer-Nieberg, S.; Pickl, S. Electric load forecasting methods: Tools for decision making. *Eur. J. Oper. Res.* **2009**, *199*, 902–907. [CrossRef]
23. Huang, S.-J.; Shih, K.-R. Short-term load forecasting via ARMA model identification including non-Gaussian process considerations. *IEEE Trans. Power Syst.* **2003**, *18*, 673–679. [CrossRef]
24. Pappas, S.S.; Ekonomou, L.; Moussas, V.; Karampelas, P.; Katsikas, S. Adaptive load forecasting of the Hellenic electric grid. *J. Zhejiang Univ. Sci. A* **2008**, *9*, 1724–1730. [CrossRef]
25. Karthika, S.; Margaret, V.; Balaraman, K. Hybrid short term load forecasting using ARIMA-SVM. In Proceedings of the 2017 Innovations in Power and Advanced Computing Technologies (i-PACT), Vellore, India, 21–22 April 2017; pp. 1–7.
26. Mukhopadhyay, P.; Mitra, G.; Banerjee, S.; Mukherjee, G. Electricity load forecasting using fuzzy logic: Short term load forecasting factoring weather parameter. In Proceedings of the 2017 7th International Conference on Power Systems (ICPS), Pune, India, 21–23 December 2017; pp. 812–819.
27. Roman, R.-C.; Precup, R.-E.; David, R.-C. Second order intelligent proportional-integral fuzzy control of twin rotor aerodynamic systems. *Procedia Comput. Sci.* **2018**, *139*, 372–380. [CrossRef]
28. Li, Y.; Fang, T. Wavelet and support vector machines for short-term electrical load forecasting. In *Wavelet Analysis and Its Applications: (In 2 Volumes)*; World Scientific: Singapore, 2003; pp. 399–404.
29. Ruzic, S.; Vuckovic, A.; Nikolic, N. Weather sensitive method for short term load forecasting in electric power utility of Serbia. *IEEE Trans. Power Syst.* **2003**, *18*, 1581–1586. [CrossRef]
30. Ekonomou, L. Greek long-term energy consumption prediction using artificial neural networks. *Energy* **2010**, *35*, 512–517. [CrossRef]
31. McClelland, J.L.; Rumelhart, D.E.; Group, P.R. Parallel distributed processing. *Explor. Microstruct. Cogn.* **1986**, *2*, 216–271.
32. Yang, J.; Stenzel, J. Short-term load forecasting with increment regression tree. *Electr. Power Syst. Res.* **2006**, *76*, 880–888. [CrossRef]
33. Deng, J.; Jirutitijaroen, P. Short-term load forecasting using time series analysis: A case study for Singapore. In Proceedings of the 2010 IEEE Conference on Cybernetics and Intelligent Systems, Singapore, 28–30 June 2010; pp. 231–236.
34. Ling, S.-H.; Leung, F.H.-F.; Lam, H.-K.; Lee, Y.-S.; Tam, P.K.-S. A novel genetic-algorithm-based neural network for short-term load forecasting. *IEEE Trans. Ind. Electron.* **2003**, *50*, 793–799. [CrossRef]
35. Liao, G.-C.; Tsao, T.-P. Application of a fuzzy neural network combined with a chaos genetic algorithm and simulated annealing to short-term load forecasting. *IEEE Trans. Evol. Comput.* **2006**, *10*, 330–340. [CrossRef]
36. Ryu, S.; Noh, J.; Kim, H. Deep neural network based demand side short term load forecasting. *Energies* **2016**, *10*, 3. [CrossRef]
37. Marino, D.L.; Amarasinghe, K.; Manic, M. Building energy load forecasting using deep neural networks. In Proceedings of the IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society, Florence, Italy, 23–26 October 2016; pp. 7046–7051.