

Article

Study on Selecting the Optimal Algorithm and the Effective Methodology to ANN-Based Short-Term Load Forecasting Model for the Southern Power Company in Vietnam [†]

Manh-Hai Pham ^{1,*}, T-A-Tho Vu ¹, Duc-Quang Nguyen ¹, Viet-Hung Dang ¹,
Ngoc-Trung Nguyen ¹, Thu-Huyen Dang ¹ and The Vinh Nguyen ²

¹ Faculty of Electrical Engineering, Electrical Power University, Hanoi 11355, Vietnam; thovta@epu.edu.vn (T.-A.-T.V.); quangndhtd@epu.edu.vn (D.-Q.N.); hungdv79@epu.edu.vn (V.-H.D.); trungnn@epu.edu.vn (N.-T.N.); huyendt@epu.edu.vn (T.-H.D.)

² Quang Ninh University of Industry, Quang Ninh 02451 Vietnam; vinhnt@qui.edu.vn

* Correspondence: haipm@epu.edu.vn; Tel.: +84-9637-175-83

[†] The present work is an extension of the paper “An Effective Approach to ANN-Based Short-Term Load Forecasting Model Using Hybrid Algorithm GA-PSO” presented to 2018 IEEE Industrial and Commercial Power Systems Europe (IEEEIC/I & CPS Europe), 12–15 June 2018, Palermo, Italy.

Received: 11 March 2019; Accepted: 3 June 2019; Published: 14 June 2019



Abstract: Recently, power companies apply optimal algorithms for short-term load forecasting, especially the daily load. However, in Vietnam, the load forecasting of the power system has not focused on this solution. Optimal algorithms can help experts improve forecasting results including accuracy and the time required for forecasting. To achieve both goals, the combinations of different algorithms are still being studied. This article describes research using a new combination of two optimal algorithms: Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). This combination limits the weakness of the convergence speed of GA as well as the weakness of PSO that it easily falls into local optima (thereby reducing accuracy). This new hybrid algorithm was applied to the Southern Power Corporation’s (SPC—a large Power company in Vietnam) daily load forecasting. The results show the algorithm’s potential to provide a solution. The most accurate result was for the forecasting of a normal working day with an average error of 1.15% while the largest error was 3.74% and the smallest was 0.02%. For holidays and weekends, the average error always approximated the allowable limit of 3%. On the other hand, some poor results also provide an opportunity to re-check the real data provided by SPC.

Keywords: short-term load forecasting; GA; PSO; 24-h daily load

1. Introduction

Regulations on the accuracy of forecasts are almost always relative. In Vietnam, following Decision No. 07/QĐ-ĐTĐL-2013 of Electricity Regulatory Authority of Vietnam (ERAV) [1], the errors of short-term load forecasting (STLF) have to be within $\pm 2\%$. However, most load forecasting research does not reach this error value. The concept of STLF does not specify the forecast object (peak load or energy consumption) but the forecast time. However, most published research considers implicitly STLF as the forecasting of instantaneous power. Therefore, conflicts still exist between the regulations and the forecast job.

The following are the main difficulties of STLF research in Vietnam:

- Selecting and filtering valuable data to prepare materials for forecasting
- Selecting suitable models of each load characteristics corresponding local site
- Legal regulations

Recently, power companies apply optimal algorithms for short-term load forecasting, especially the daily load. The Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) are applied separately to train artificial neural networks (ANNs). The integration of GA into STLF is published in many papers (e.g., [2–8]). In these studies, the authors carefully used the most suitable parameters of GA to improve the precision. While GA shows improved accuracy, PSO accelerates the training process [9]. Recently, selecting among GA, PSO, back-propagation or other algorithms based on ANN is becoming more and more enticing [10–13]. On the other hand, the precision is affected by many criteria, one of which is the selected method or algorithm. Hybrid methods are described in [14,15]. H. Garg [14] used GA-PSO to solve a nonlinear optimal problem. The author used GA to improve each particle. Correspondingly, GA needs to frequently carry out the dual operation: crossover and mutation. This will extremely enlarge the total number of loops, thus affecting the run-time. Considering this drawback, Sahoo et al. [16] also exploited a hybrid GA-PSO algorithm, in which GA is executed repeatedly.

This paper describes an effective hybrid method where GA is executed only once. This paper presents an outline of the STLF approach (Section 2), the input-target data and materials (Section 3) and our effective hybrid GA-PSO algorithm for ANN training (Section 4). The conclusion is drawn unambiguously in Section 5. This paper is an extended version of our paper published in 2018 IEEE Industrial and Commercial Power Systems Europe (IEEEIC/I & CPS Europe), 12–15 June 2018, Palermo, Italy [17].

2. Methodology of Short-Term Load Forecasting

2.1. Selecting the Model of STLF

Many models are currently used for STLF in Vietnam. However, the three main models are:

- Model 1: Combining peak and valley load and daily load pattern.
- Model 2: Directly forecasting 24 h daily load at the same time.
- Model 3: Forecasting instantaneous power load step by step of 1 h.

This work selects the Model 1 illustrated in [18]. This model consists of three separate parts: the maximum real load power (peak load) forecasting, the minimum real load power (valley load) forecasting and the identified day types (load pattern). The overview of this model is demonstrated in Figure 1.

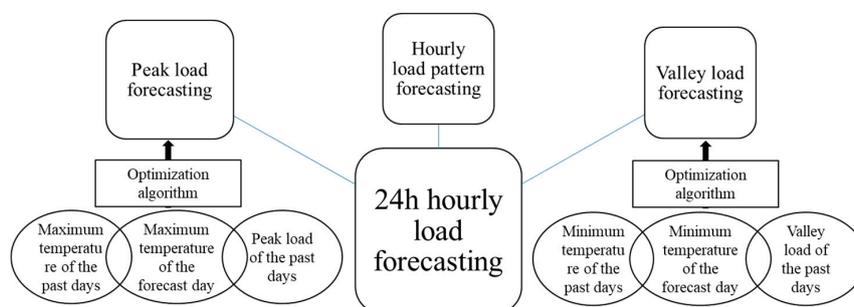


Figure 1. The selected model of STLF for Southern Power Corporation in Vietnam.

In this model, the forecast error depends on all three parts: peak and valley load and load pattern forecasting. The peak and valley load errors are usually smaller than the load pattern. The partial errors in this model are differentiated and optimized separately to gain suitable adjustments during the grid operation for minimizing the errors in future forecasting.

2.2. Determining the Daily Load Forecasting from Peak Load, Valley Load and Load Pattern

The peak load and the valley load are forecasted firstly as explained in Sections 3 and 4. We use the following two equations to determine daily load:

$$P_{ni} = \frac{P_i - P_{pmin}}{P_{pmax} - P_{pmin}} \quad (1)$$

$$P_i = P_{fmin} + (P_{fmax} - P_{fmin}) \times P_{ni} \quad (2)$$

where

- P_{ni} is the normalized load for hour i (this normalization increases the execution speed of algorithm);
- P_{pmax} and P_{pmin} are the real average peak load and valley load of past days, which are in the same group with the forecast day (the number of similar days used is five by experience when we tried to change it in on forecasting);
- P_i is the forecasting load of i th hour of the forecasting day; and
- P_{fmax} and P_{fmin} are the peak load and valley load of the forecasting day.

2.3. Determining the Load Pattern

Calculating the load pattern greatly affects the accuracy of the forecasting. There are several published forecast methods but the most effective method is still based on the experience of predictors to choose previous days with similar load patterns as the forecast day. In our method, all days may be classified in terms of the daily load pattern into eight groups: five working day groups (from Monday to Friday), two weekend groups (Saturday and Sunday), and one holiday (or special-event day) group. The demonstration of load's characteristics can be observed in the case of one week in Figure 2 (from 29 August to 3 September 2016). This week consists of a holiday (Friday, 2 September, is Vietnam Nation day). The differences among the three groups is striking: working days, weekend and holidays. However, caused by characteristics of SPC's load, each day of the week has its own load pattern, which is further distinguished if it coincides with a holiday.

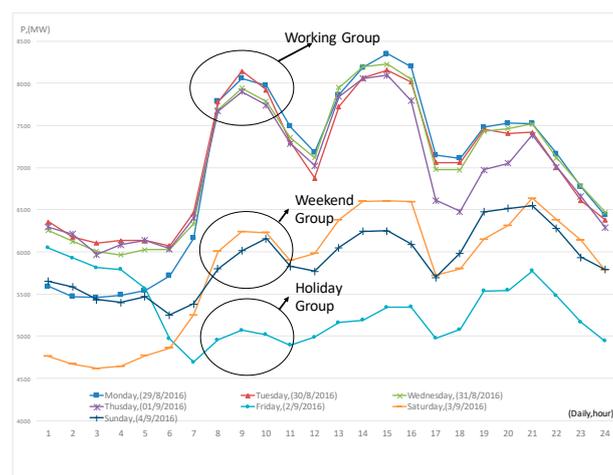


Figure 2. The daily load of one week in August 2016.

In this article, five similar days (holiday, day of week, and season of year) are used to reform Equation (1). Then, the normalized load pattern is calculated by the average formula:

$$P_i = \frac{1}{5} \sum_{j:1}^5 P_{ij} \quad (3)$$

where P_{ij} is the normalized hourly load of the selected similar days. During the normalization, we eliminate some days if there are sudden differences on the load pattern graph by checking the correlative coefficients with two steps:

- Calculate the correlative coefficients between the similar days by the CORREL function.
- Evaluate the value of correlative coefficients and eliminate the days corresponding to the value out of range [0.9, 1].

2.4. Selecting the GA Algorithm

Currently, Genetic Algorithm (GA) is one of the most popular algorithms on research using ANN. The basic knowledge of GA is described clearly in Vietnamese and international publications. Therefore, we do not focus on describing GA in this paper but only using GA to apply the load forecasting for a Vietnamese power company.

GA has shown to be a strong and fairly accurate algorithm in research about optimization problems for a large power system. There are many GA-based studies in the field of STLF. In 1994, Maifeld et al. [19] published STLF research based on ANN and GA. In this article, the authors called the basic GA by the name of RGA. RGA also includes three operators: reproduction, crossover and mutation. The diagram of RGA is shown in Figure 3; however, the order of steps is different from other studies.

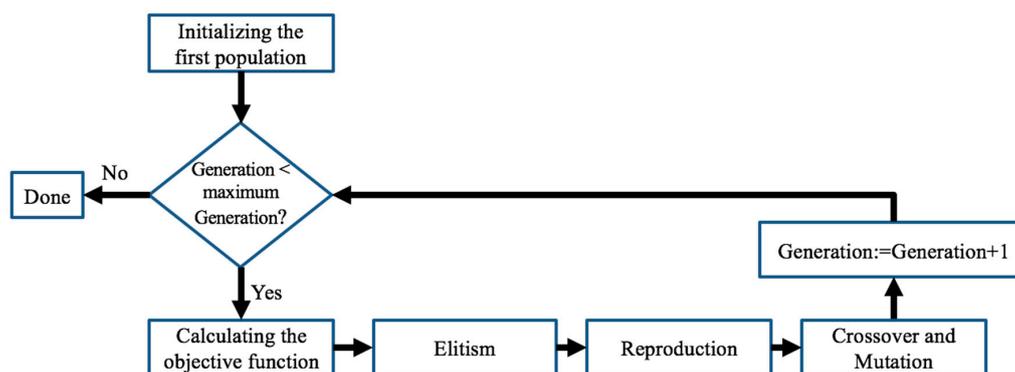


Figure 3. The model of RGA.

Maifeld et al. [19] provided comparisons between back propagation (BP) and RGA on forecasting of 12 h of a day. The errors are shown in Table 1.

Table 1. The comparison between BP and RGA.

Algorithm	Average Error (%)	Maximum Error (%)
RGA	1.8	4.0
BP	2.6	6.6

Although the error of RGA is smaller than the one of BP algorithm, the authors stated that the execution time of RGA is much greater.

Several studies focus on optimizing the ANN operations instead of improving GA. One successful approach was presented by Ling et al. [20]. They compared results using the traditional ANN and an improved ANN integrating the same GA. The object of this publication is the household daily load. On that, the GA affects only some hidden class neurons to create a well-trained neural network with respect to the fitness value. Recently, studies on ANN using GA are mostly improved using hybrid algorithms, where GA acts as one of the main partners. The hybrid algorithm usually exploits basic GA with three key operators: reproduction, crossover and mutation. In each operator, it is

absolutely necessary to select key parameters. In [21,22], the encryption is of two kinds, namely encryption by binary string and encryption by real value; the crossover consists of three types, namely the crossover-weights, the crossover-nodes and the crossover-features; and the mutation can be either unbiased or biased. Finally, the key parameters were determined as below:

- Crossover method: Crossover-weights
- Mutation method: Biased (with fixed probability 0.1)

2.5. Selecting the PSO Algorithm

2.5.1. Overview of PSO Algorithms

The construction of PSO was first articulated by J. Kennedy and R.C. Eberhart (1995) [23] and improved step-by-step in their later publications. The basic PSO is defined as:

$$\begin{cases} V_i^{k+1} = V_i^k + C_1 \times R_1 \times (P_i^k - x_i^k) + C_2 \times R_2 \times (G^k - x_i^k) \\ x_i^{k+1} = x_i^k + V_i^{k+1} \end{cases} \quad (4)$$

where

V_i^k is the velocity of particle i at the loop k ;

x_i^k is the position of particle i at the loop k ;

C_1 and C_2 are the learning fixed factors;

R_1 and R_2 are random values within (0,1);

P_i^k is the best position of particle i at the loop k ; and

G^k is the best position of swarm at the loop k .

Note that choosing $C_1 > C_2$ shows the bias direction of the swarm's movement according to individual optimization (P_{best}) or global optimization (G_{best}) and thus affects the convergence rate of PSO.

To avoid the separation of the swarm caused by the speed of movement, a proposed speed limit is set at each iteration [24]. However, the speed limit also adversely affects to the swarm's searching space. By this conflict, a parameter w (a constant or even a function) called the "inertia weight" was given by Y. Shi and R.C. Eberhart [25]. The inertia weight is brought into the basic PSO as shown in Equation (5) (an advanced PSO):

$$\begin{cases} V_i^{k+1} = w \times V_i^k + C_1 \times R_1 \times (P_i^k - x_i^k) + C_2 \times R_2 \times (G^k - x_i^k) \\ x_i^{k+1} = x_i^k + V_i^{k+1} \end{cases} \quad (5)$$

The authors of [26,27] utilized Equation (6) to give the inertia weight w^k on each iteration of swarm:

$$w^k = w_{max} - \frac{k}{k_{max}}(w_{max} - w_{min}) \quad (6)$$

where k stands for the actual number of epochs and k_{max} is the maximum number of epochs.

The authors showed that the recommended range for $[w_{min}, w_{max}]$ is $[0.4, 0.9]$.

The participation of inertia weights w aims to avoid the swarm's separation; however, the algorithm may still fall into the local convergence in the case of the multidimensional searching space. This is caused by information exchange mechanism among particles in the swarm. Thus, another modified PSO was proposed for this mechanism. Instead of exchanging information with all particles in the swarm, each one exchanges information of position and velocity in a group of similar particles.

Therefore, G^k in the basic PSO is replaced by L^k , meaning the optimal local value. The general equations of this advanced PSO is as follows:

$$\begin{cases} V_i^{k+1} = w \times V_i^k + C_1 \times R_1 \times (P_i^k - x_i^k) + C_2 \times R_2 \times (L^k - x_i^k) \\ x_i^{k+1} = x_i^k + V_i^{k+1} \end{cases} \quad (7)$$

One of the new PSO trends is Stand PSO (SPSO), which is currently presented in many studies on PSO applications. The first was conducted by Ozcan and Mohan [28]. Their results show the movement of particles in the searching space. A few years later, their research was updated with more details by Clerc and Kennedy [29], who analyzed the convergence of the algorithm. The SPSO is defined as:

$$\begin{cases} V_i^{k+1} = X \times [V_i^k + C_1 \times R_1 \times (P_i^k - x_i^k) + C_2 \times R_2 \times (G^k - x_i^k)] \\ x_i^{k+1} = x_i^k + V_i^{k+1} \end{cases} \quad (8)$$

This model may also be applied to the advanced PSO mentioned in Equation (7):

$$\begin{cases} V_i^{k+1} = X \times [V_i^k + C_1 \times R_1 \times (P_i^k - x_i^k) + C_2 \times R_2 \times (L^k - x_i^k)] \\ x_i^{k+1} = x_i^k + V_i^{k+1} \end{cases} \quad (9)$$

where X is called by the constriction factor. This research experimentally demonstrated the optimal value of X by the following formula:

$$X = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|} \quad (10)$$

$$\varphi = C_1 + C_2, \varphi > 4$$

As mentioned above, SPSO focuses primarily on the convergence of the algorithm. The selected coefficients also aim to accelerate the convergence speed while ensuring the accuracy of the optimization. However, as we know, rapid convergence also means the possibility of falling into local convergence traps. Therefore, we need to choose either PSO or SPSO to combine with GA in our hybrid algorithm before applying to a specific subject. To select which PSO to combine with the hybrid algorithm, we separately implemented both modified PSO algorithms (basic PSO and SPSO). After that, the load forecasting was compared using the predictive errors. In addition, the rate of convergence was also considered by comparing the graphs of the value of objective function (MSE-mean square error) according to each iteration. Due to the data source collection, we utilized the available dataset of Southern Power Corporation (SPC: <https://evnspc.vn/>) in 2014.

2.5.2. Comparison between PSO and SPSO

Both PSO algorithms were tested with the same algorithm shown in Figure 4.

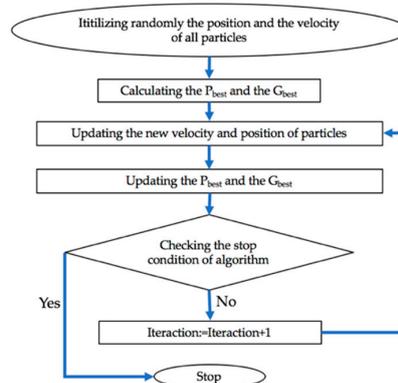


Figure 4. The general schema of both PSO algorithms.

The PSO algorithms were configured with the following coefficients:

The Basic PSO: $C_1 = 2$ and $C_2 = 1$. The values of C_1 and C_2 represent the movement priority of particles according to P_{best} or G_{best} . The value of C_1 being more than C_2 denotes the reduction of rate of signal propagation to G_{best} of the swarm. This also helps overcome the disadvantage relating to the local convergence traps. R_1 and R_2 are random coefficients in the range of $[0,1]$. According to the above-mentioned studies, the value of w decreases from $w_{max} = 0.9$ to $w_{min} = 0.4$. However, during the simulating process, the decrease of w on the next iterations would reduce the movement speed of the swarm, leading to fall into the local convergence traps. Thus, we propose a solution to overcome this disadvantage. In the first 100 iterations, the value of w decreases from 0.9 to 0.4 according to Equation (6). From Iteration 101, the value of w remains equal to 0.9. This solution will help the swarm searching space be big enough to increase the likelihood of finding the optimal solution.

The SPSO: According to Clerc and Kennedy [29], the value of coefficients significantly affects the error of an algorithm. By empirical simulations, the optimal values of coefficients were relatively determined as follows: $X = 0.729$; $C_1 = 2.05$; $C_2 = 2.05$; and R_1 and R_2 randomly fixed in the range of $[0,1]$.

Table 2 shows the comparison between the basic PSO and the SPSO.

Table 2. The comparison between the basic PSO and SPSO.

Algorithm	Average Error (%)	Maximum Error (%)	Minimum Error (%)
Basic PSO	0.85	4.225	0.002
SPSO	1.021	4.803	0.056

The basic PSO is slightly better than SPSO but the difference is still not clear. To be able to see more clearly the process of both algorithm, the MSE value of each training process was analyzed and represented as a graph. Figure 5 shows the MSE value through each iteration.

Observing Figure 5, it is clear that:

- MSE value with SPSO reduces very quickly during the first 100–200 iterations. Then, it almost does not change. This is the main disadvantage of SPSO algorithm because SPSO is researched and developed to increase the convergence speed of the swarm. We can see that, if the number of iterations is bigger, the value of MSE almost insignificantly decreases. Thus, the accuracy of the algorithm is not improved with more iterations.
- MSE value with basic PSO has a slower reduction than SPSO but, after 200 iterations, the MSE value continues to decrease steadily and is smaller than SPSO's from about 400 iterations. This shows the positive effect of adjusting the value of inertia weight w after 100 iterations (remaining equal to 0.9).

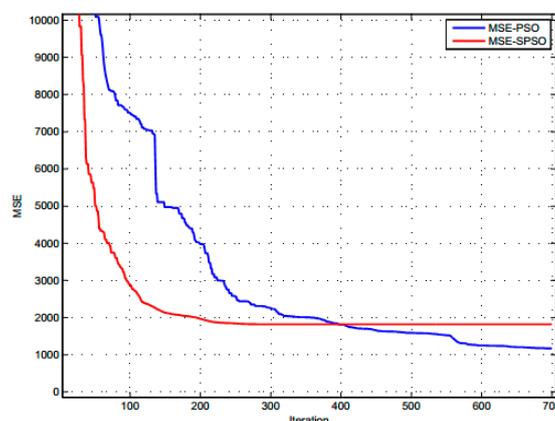


Figure 5. The value of MSE in all iterations with basic PSO and SPSO.

By comparing the two PSO algorithms, we conclude that applying SPSO algorithm to load forecasting only increases the speed of implementation but does not improve the result and it may fall into the local convergence traps. Meanwhile, we can still improve the error of result by increasing the number of iterations with the basic PSO. Therefore, we chose the basic PSO to combine with GA in the GA-PSO hybrid algorithm applied in the load forecasting of SPC.

2.6. Selecting the Hybrid GA-PSO Algorithm

As we known, an optimal algorithm also may apply to all studies of optimization. H. Garg [14] conducted research using the GA-PSO hybrid algorithm to solve a nonlinear optimization problem. The roles of GA and PSO are clearly described in Figure 6.

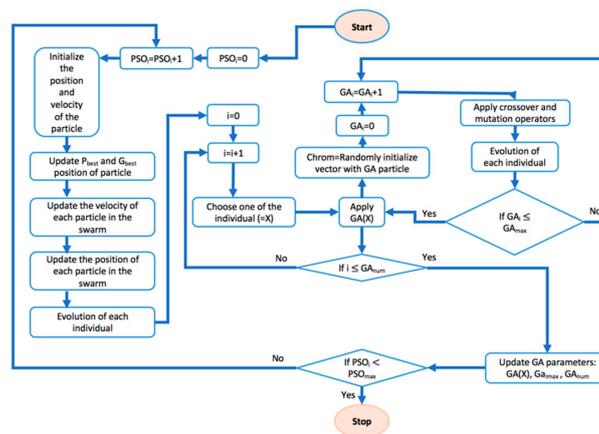


Figure 6. The hybrid method of Harish Garg.

In this combination method, the GA is used to optimize each individual in the swarm. Accordingly, the GA algorithm must be executed continuously with both crossover and mutation operators. This greatly increases the runtime of the simulation.

Using the same idea as Harish Garg (PSO algorithm is used to select better individuals of the initial iteration before performing steps of evolution), Q. Zhang et al. [15] provided a simpler algorithm to optimize the parameters of direct-injection diesel engine running with soy biodiesel. The PSO algorithm is done on the n best individuals, thus producing n offspring for use in the next iteration (generation). The remaining $N-n$ individuals are eliminated to make room for the new better individuals generated by GA step. This step aims to create a new generation with the same quantity of particles of swarm. The flowchart of this combination method is presented in Figure 7.

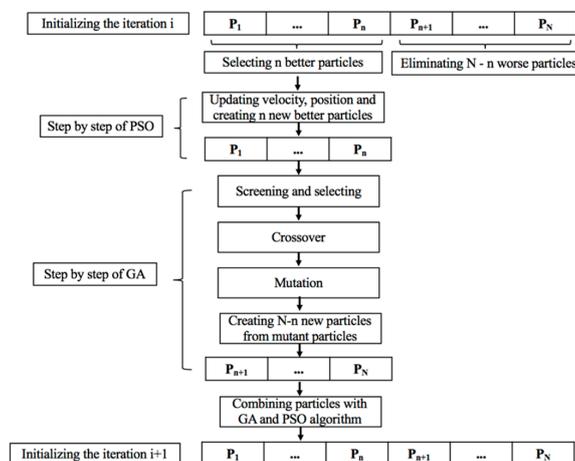


Figure 7. The combination method of Q. Zhang reproduced from [15], Appl. Energy: 2016.

Following this research, the authors mentioned the optimal runtime of hybrid PSO-GA algorithm in comparison to basic GA and basic PSO. However, demonstrating the comparison between PSO-GA and single PSO is not shown. In another study using a parallel combination of GA-PSO, Sahoo et al. [16] also presented similar comparisons and mentioned the inherent weakness of runtime if GA algorithm executes iterations in more than one program loop. Therefore, in our hybrid algorithm, GA algorithm is implemented first and does not repeat in any other loop to avoid this drawback. After that, PSO loops perform its optimal work. This is a helpful, simple solution to apply to load forecasting in Vietnam. The GA step is implemented independently of PSO.

2.6.1. Step by Step of the Selected Hybrid Algorithm

The steps of combining GA and PSO algorithms in the progress of optimizing the ANN's parameters for application to load forecasting are briefly described according to Figure 8. All detailed steps are listed in order below.

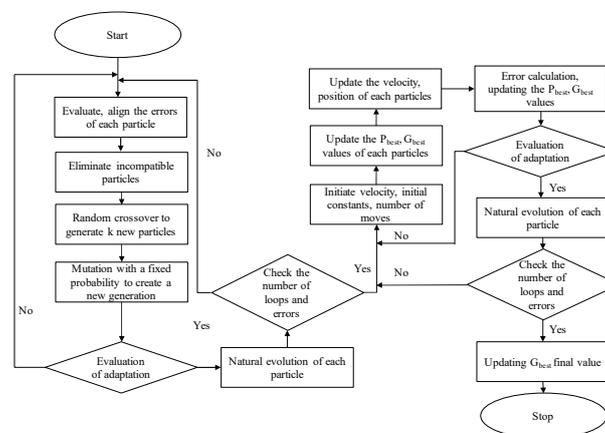


Figure 8. The schema of the hybrid GA-PSO algorithm.

Genetic Algorithm

Step 1: Prepare the data for ANN.

Step 2: Configure the ANN.

Step 3: Initialize the first weights (particle in swarm) and the number of generations.

Step 4: Train ANN with each generated particle using the same load data. Calculate the respective errors (MSE values).

Step 5: After obtaining the MSE values, sort them in ascending order and then remove the large value errors from the swarm (natural selection).

Step 6: With the remaining particles of the swarm, randomly implement the crossover together to generate the new particles, ensuring the initial population size remains unchanged.

Step 7: Implement the mutation step with selected probability.

Step 8: Assess the adaptability of new generation with MSE error and stop the loop.

Step 9: Implement natural evolution until one of the stop requirements is attained.

Particle Swarm Optimization

Step 10: Initialize the velocity, the position, the constants and the number of iterations. Take the first generation of swarm from the GA loop.

Step 11: Integrate particles on ANN and simulate to pick of the MSE errors. Calculate the P_{best} of each particle and G_{best} of the swarm.

Step 12: Calculate the velocity of each particle and update for all.

Step 13: Train the ANN with the new generation of the swarm. Calculate and save the MSE errors. End the iteration.

Step 14: Continue the swarm's natural evolution until the maximum iteration.

Step 15: Take the final G_{best} to train the ANN for load forecasting.

This hybrid algorithm is implanted in the ANN structure as demonstrated:

- Structure type: feedforward network
- Neuron quantity of input layer: 13
- Neuron quantity of output layer: 1
- Quantity of hidden layers: 1
- Neuron quantity of hidden layer: 5
- Fitness function in GA-PSO: MSE

2.6.2. GA-PSO vs. Basic PSO

Usually the accuracy of load forecasting using the GA-PSO hybrid algorithm would be demonstrated by comparison with basic GA and basic PSO. However, in discussing the results from simulation, we decided not to compare with simulation using basic GA due to its big errors. Therefore, in this section, we only show simulation results using GA-PSO hybrid algorithm and the basic PSO algorithm. To ensure the total of iterations corresponding to the simulation using basic PSO (700 iterations), we performed 100 iterations with GA and 600 iterations with PSO algorithm. This is an imperfect comparison because one iteration of GA is not the same as one of PSO. However, we could compare the two simulation results. The comparison of MSE errors according to each iteration between these two algorithms is demonstrated by Figure 9.

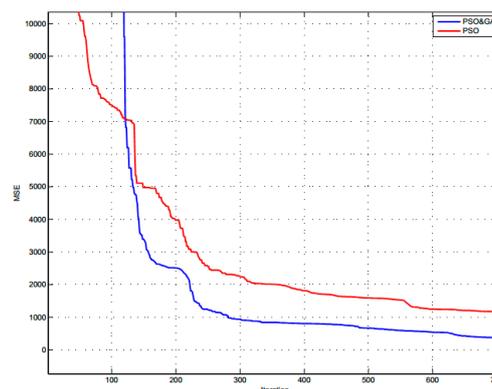


Figure 9. The value of MSE in all iterations with basic PSO and GA-PSO.

It is easy to see that, in the first iterations, the MSE value with GA decreased more slowly than the PSO algorithm, but, after that, the MSE value dropped very quickly and gradually surpassed the one using the basic PSO algorithm. Meanwhile, after 100 iterations, the MSE value using the basic PSO algorithm began to decrease more slowly. This shows that GA algorithm make the MSE value to achieve relatively good accuracy to continue the evolution with PSO algorithm, ensuring the large search space of the swarm.

There are two issues to keep in mind when using hybrid GA-PSO algorithm:

- Accuracy of GA algorithm decreased more slowly than PSO algorithm. Therefore, if the total of iterations with GA were too large, it would reduce the overall performance of hybrid algorithm.
- One of the operators in GA algorithm is crossover, meaning that more iterations being implemented will lead to more individuals of future generations having more characteristics of their parent individuals, thus they will gradually become more similar. Thus, if the total iterations with GA were too large, the initial weights would be nearly the same or, in other words, the search space of the swarm would be smaller. This would lead to the reduction of performance of the hybrid algorithm. Accordingly, we infer that the selected total iterations with GA needs to be balanced

(depending much on the experience of the forecaster) to bring the most efficiency to the whole ANN training process.

3. Prepare Input Data and Materials

This research exploited the huge input data supplied by one of Vietnam's big local power companies (SPC). In this experiment, the datasheet stopped on 18 November 2016 and it was required to forecast daily power load of 19 November and 20 November. Further, the forecast was implemented for one week (14–20 November) to improve the accuracy of forecasting algorithm. All obtained results were assessed and discussed. As the same implementation, we comprehensively present the input data for 19 November 2016 (only for comparison purposes). The daily highest temperature were used as the weather criterion for peak load and the daily lowest temperature for valley load. The temperature variables of all past days and forecast day were collected from the data source on "weather.com". Note that the SPC's load consists of 21 provinces with large area, so the temperature also varies. Thus, we only used the average temperature of the whole region in selected hybrid algorithm.

3.1. The Datasheet of Input Data

The input data of 19 November 2016 was prepared in an Excel datasheet (see Table 3 (for peak load) and Table 4 (for valley load)). These tables show 28 days from 28 October 2016 to 18 November 2016 on 28 rows and 15 columns that describe 13 parameters (or 13 neurons on output layout):

- Neuron 1 (Column 3): Day's temperature of one day from current day (the current day is noted in the Columns 1 and 2).
- Neuron 2 (Column 4): Temperature of this day last week.
- Neuron 3 (Column 5): Temperature of current day.
- Neuron 4 (Column 6): Day's peak load of one day from current day.
- Neuron 5 (Column 7): Peak load of this day last week.
- Neurons 6–13: Encoding all days of week (considering holidays).

Table 3. The input data for peak load training.

Day of Week	Date	T_{pmax-1} °C	T_{pmax-7} °C	T_{pmax} °C	P_{pmax-1} MW	P_{pmax-7} MW	Encryption								
Saturday	22 October 2016	32	33	32	8020	7874	0	0	0	0	0	0	0	1	0
Sunday	23 October 2016	32	33	33	7569	6248	0	0	0	0	0	0	0	0	1
Monday	24 October 2016	33	32	31	6113	8007	0	1	0	0	0	0	0	0	0
Tuesday	25 October 2016	31	31	30	7961	8115	0	0	1	0	0	0	0	0	0
Wednesday	26 October 2016	30	32	33	7873	7873	0	0	0	1	0	0	0	0	0
Thursday	27 October 2016	33	30	34	8014	7935	0	0	0	0	1	0	0	0	0
Friday	28 October 2016	34	32	33	8184	8020	0	0	0	0	0	1	0	0	0
Saturday	29 October 2016	33	32	33	8031	7569	0	0	0	0	0	0	0	1	0
Sunday	30 October 2016	33	33	31	7505	6113	0	0	0	0	0	0	0	0	1
Monday	31 October 2016	31	31	32	6164	7961	0	1	0	0	0	0	0	0	0
Tuesday	1 November 2016	32	30	32	7775	7873	0	0	1	0	0	0	0	0	0
Wednesday	2 November 2016	32	33	32	7926	8014	0	0	0	1	0	0	0	0	0
Thursday	3 November 2016	32	34	30	7950	8184	0	0	0	0	1	0	0	0	0
Friday	4 November 2016	30	33	29	7320	8031	0	0	0	0	0	0	1	0	0
Saturday	5 November 2016	29	33	27	7725	7505	0	0	0	0	0	0	0	1	0
Sunday	6 November 2016	27	31	28	7311	6164	0	0	0	0	0	0	0	0	1
Monday	7 November 2016	28	32	34	5953	7775	0	1	0	0	0	0	0	0	0
Tuesday	8 November 2016	34	32	32	7725	7926	0	0	1	0	0	0	0	0	0
Wednesday	9 November 2016	32	32	33	7774	7950	0	0	0	1	0	0	0	0	0
Thursday	10 November 2016	33	30	34	8264	7320	0	0	0	0	1	0	0	0	0
Friday	11 November 2016	34	29	34	8109	7725	0	0	0	0	0	0	1	0	0
Saturday	12 November 2016	34	27	34	7976	7311	0	0	0	0	0	0	0	1	0
Sunday	13 November 2016	34	28	34	7828	5953	0	0	0	0	0	0	0	0	1
Monday	14 November 2016	34	34	35	7725	7725	0	1	0	0	0	0	0	0	0
Tuesday	15 November 2016	35	32	34	8052	7774	0	0	1	0	0	0	0	0	0
Wednesday	16 November 2016	34	33	35	8382	8264	0	0	0	1	0	0	0	0	0
Thursday	17 November 2016	35	34	33	8290	8109	0	0	0	0	1	0	0	0	0
Friday	18 November 2016	33	34	33	8197	7976	0	0	0	0	0	0	1	0	0

Table 4. The input data for valley load training.

Day of Week	Date	T_{pmin-1} °C	T_{pmin-7} °C	T_{pmin} °C	P_{pmin-1} MW	P_{pmin-7} MW	Encryption								
Saturday	22 October 2016	24	25	25	6052	6043	0	0	0	0	0	0	0	1	0
Sunday	23 October 2016	25	24	24	5775	5268	0	0	0	0	0	0	0	0	1
Monday	24 October 2016	24	24	24	5135	5205	0	1	0	0	0	0	0	0	0
Tuesday	25 October 2016	24	25	24	5299	6006	0	0	1	0	0	0	0	0	0
Wednesday	26 October 2016	24	25	25	5961	6079	0	0	0	1	0	0	0	0	0
Thursday	27 October 2016	25	24	25	5989	5914	0	0	0	0	1	0	0	0	0
Friday	28 October 2016	25	24	24	6088	6052	0	0	0	0	0	0	1	0	0
Saturday	29 October 2016	24	25	24	6031	5775	0	0	0	0	0	0	0	1	0
Sunday	30 October 2016	24	24	24	5256	5135	0	0	0	0	0	0	0	0	1
Monday	31 October 2016	24	24	24	4966	5299	0	1	0	0	0	0	0	0	0
Tuesday	1 November 2016	24	24	23	5210	5961	0	0	1	0	0	0	0	0	0
Wednesday	2 November 2016	23	25	24	5919	5989	0	0	0	1	0	0	0	0	0
Thursday	3 November 2016	24	25	25	5980	6088	0	0	0	0	1	0	0	0	0
Friday	4 November 2016	25	24	24	5830	6031	0	0	0	0	0	0	1	0	0
Saturday	5 November 2016	24	24	24	5984	5256	0	0	0	0	0	0	0	1	0
Sunday	6 November 2016	24	24	24	5734	4966	0	0	0	0	0	0	0	0	1
Monday	7 November 2016	24	24	25	5005	5210	0	1	0	0	0	0	0	0	0
Tuesday	8 November 2016	25	23	25	5984	5919	0	0	1	0	0	0	0	0	0
Wednesday	9 November 2016	25	24	24	5812	5980	0	0	0	1	0	0	0	0	0
Thursday	10 November 2016	24	25	25	6006	5830	0	0	0	0	1	0	0	0	0
Friday	11 November 2016	25	24	25	6104	5984	0	0	0	0	0	0	1	0	0
Saturday	12 November 2016	25	24	25	6125	5734	0	0	0	0	0	0	0	1	0
Sunday	13 November 2016	25	24	25	6109	5005	0	0	0	0	0	0	0	0	1
Monday	14 November 2016	25	25	25	5984	5984	0	1	0	0	0	0	0	0	0
Tuesday	15 November 2016	25	25	26	5270	5812	0	0	1	0	0	0	0	0	0
Wednesday	16 November 2016	26	24	25	6358	6006	0	0	0	1	0	0	0	0	0
Thursday	17 November 2016	25	25	26	6387	6104	0	0	0	0	1	0	0	0	0
Friday	18 November 2016	26	25	26	6283	6125	0	0	0	0	0	0	1	0	0

The mean is shown in Table 5.

Table 5. The encryption of days.

Date	Holiday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
Code	11000000	01000000	00100000	00010000	00001000	00000100	00000010	00000001

Notes that 11000000 is a Monday’s encryption on Holiday.

3.2. The Datasheet of Target Data

The target data were used in the training process, as shown in Tables 6 and 7.

Table 6. The target data for peak load training.

Day of Week	Date	$P_{maxcurrent\ day}$ (MW)
Saturday	22 October 2016	7569
Sunday	23 October 2016	6113
Monday	24 October 2016	7961
Tuesday	25 October 2016	7873
Wednesday	26 October 2016	8014
Thursday	27 October 2016	8184
Friday	28 October 2016	8031
Saturday	29 October 2016	7505
Sunday	30 October 2016	6164
Monday	31 October 2016	7775
Tuesday	1 November 2016	7926
Wednesday	2 November 2016	7950
Thursday	3 November 2016	7320
Friday	4 November 2016	7725
Saturday	5 November 2016	7311
Sunday	6 November 2016	5953
Monday	7 November 2016	7725
Tuesday	8 November 2016	7774
Wednesday	9 November 2016	8264
Thursday	10 November 2016	8109
Friday	11 November 2016	7976
Saturday	12 November 2016	7828
Sunday	13 November 2016	7725
Monday	14 November 2016	8052
Tuesday	15 November 2016	8382
Wednesday	16 November 2016	8290
Thursday	17 November 2016	8197
Friday	18 November 2016	8173

Table 7. The target data for valley load training.

Day of Week	Date	$P_{\text{mincurrent day}}$ (MW)
Saturday	22 October 2016	5775
Sunday	23 October 2016	5135
Monday	24 October 2016	5299
Tuesday	25 October 2016	5961
Wednesday	26 October 2016	5989
Thursday	27 October 2016	6088
Friday	28 October 2016	6031
Saturday	29 October 2016	5256
Sunday	30 October 2016	4966
Monday	31 October 2016	5198
Tuesday	1 November 2016	5919
Wednesday	2 November 2016	5980
Thursday	3 November 2016	5830
Friday	4 November 2016	5984
Saturday	5 November 2016	5734
Sunday	6 November 2016	5005
Monday	7 November 2016	5984
Tuesday	8 November 2016	5812
Wednesday	9 November 2016	6006
Thursday	10 November 2016	6104
Friday	11 November 2016	6125
Saturday	12 November 2016	6109
Sunday	13 November 2016	5984
Monday	14 November 2016	5270
Tuesday	15 November 2016	6358
Wednesday	16 November 2016	6387
Thursday	17 November 2016	6283
Friday	18 November 2016	5830

3.3. The Datasheet of Predicted Data

After ANN training process, the optimal weights of ANN were used to predict the peak and valley load. Tables 8 and 9 show the datasheets prepared for the forecasting of 19 November 2016.

Table 8. The predicted data for valley load of 19 November 2016.

Day of Week	Date	T_{pmin-1} °C	T_{pmin-7} °C	T_{pmin} °C	P_{pmin-1} MW	P_{pmin-7} MW	Encryption								
Saturday	19 November 2016	26	25	26	5830	6109	0	0	0	0	0	0	0	1	0

Table 9. The predicted data for peak load of 19 November 2016.

Day of Week	Date	T_{pmax-1} °C	T_{pmax-7} °C	T_{pmax} °C	P_{pmax-1} MW	P_{pmax-7} MW	Encryption								
Saturday	19 November 2016	33	34	34	8173	7828	0	0	0	0	0	0	0	1	0

Note that the past temperatures are always real values but the current day is a predicted value.

3.4. Materials for Forecasting

We used MATLAB version R2015b, The MathWorks, Inc. with ANN toolbox to implement the STLF. One MacBook Pro version 2015 with processor Intel Core i7 2 GHz, 8 GB memory 1600 MHz DDR3 was exploited.

4. Using Effective Hybrid Algorithm GA-PSO for STLF

4.1. Forecasting for 19 November

After training ANN and executing the hybrid algorithm, values of peak load and valley load forecasting on 19 November 2016 were, respectively, 7769.8 MW and 6134.1 MW. Using Equations (1) and (2), the results are shown in Figure 10.

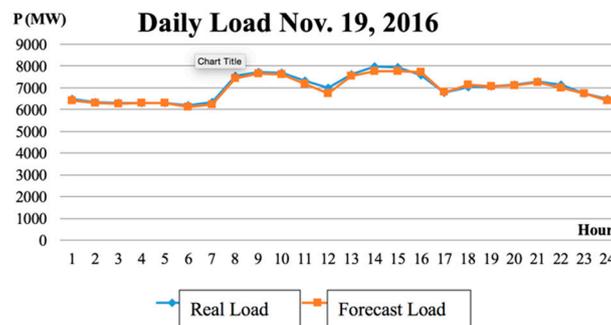


Figure 10. The result of 19 November 2016.

As shown in Figure 11, the largest error was 3.736% at 12:00, the smallest was 0.018% at 04:00 and the average error was 1.153%. With this distribution, we found that no large errors (more than 2%) were during the rush-hour load. Considering the provisions in Decision No. 7 of the Electricity Regulatory Authority of Vietnam allow the error of 2% for the daily load forecasting, the results exceed the allowable error at three times (12:00, 14:00 and 22:00). However, the average error was still within the allowable range. Thus, we suggest these results for the grid operation.

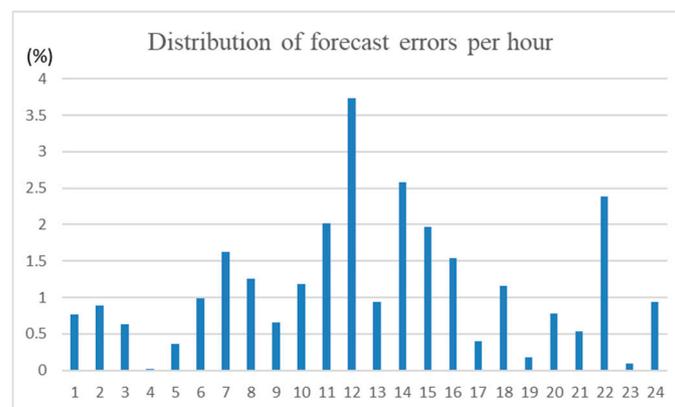


Figure 11. Distribution of forecast errors of 19 November 2016 forecasting.

4.2. Forecasting for 14–18 November

Turning now to the experimental evidence on 16 November 2016, where the result was not good. This day returned the worst result (the worst average error). Figures 12 and 13 show that weak points of the hybrid algorithm exist.

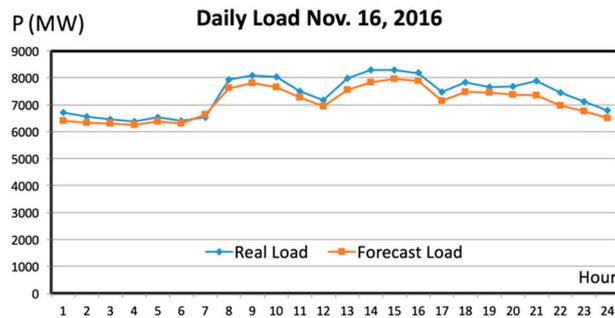


Figure 12. The result of 16 November 2016.

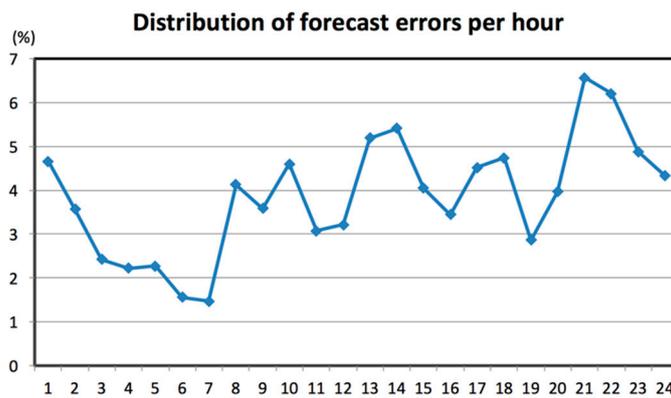


Figure 13. Distribution of forecast errors of 16 November 2016 forecasting.

The biggest error was 6.57% at 21:00, the smallest was 1.47% at 07:00 and the average was 3.87%. This result is not allowable for load forecasting. This excess error can be explained by two reasons. The first is due to the load’s variation of similar days in processing of hourly load pattern forecasting. The second is due to the difference between local temperatures of 21 provinces on this day.

However, the results were not too poor on 14, 15, 17 and 18 November. The obtained results for peak load, valley load and forecasting are summarized in Table 10

Table 10. The Summary of the forecasts for 14, 15, 17 and 18 November.

Date	Peak load			Valley Load			Daily Load Forecasting		
	Forecast	Real	Error (%)	Forecast	Real	Error (%)	Max Error (%)	Min Error (%)	Average Error (%)
14 November	7840	8052	2.63	5157.9	5270	2.13	5.99	0.15	2.53
15 November	7960.2	8382	5.03	6008.5	6358	5.50	6.24	0.11	3.00
17 November	8016	8197	2.21	6008.1	6283	4.37	5.72	1.24	3.36
18 November	8137.3	8173	0.44	5996.5	5830	2.85	25.97	0.07	2.44

It is easy to see the average errors of these forecasts were always approximately 3% (satisfying the requirement of load forecasting regulation in Vietnam). The forecast errors in Table 9 include peak, valley and daily load error. As mentioned above, these error types were released by three forecast parts: peak, valley and hourly load forecasting respectively. We realize that the peaks and valleys are not synonymous with poor daily errors. Specifically, the 14, 15 and 17 November results have peak and valley load errors of approximately 5% corresponding to the maximum of daily load error of 6% and the minimum of 1%. Meanwhile, on 18 November, although the peak and valley load errors were good (0.44% and 2.85%), the maximum of daily load error reached 25.97%. Although the minimum of error remained as stable as the other days, even better (0.07%), such a maximum daily load error is unacceptable. This evokes doubts about the real load data provided by SPC. Indeed, we tried to compare similar days (all Fridays in the last two months, i.e., October and November) on the hourly load pattern. Figure 14 shows this comparison.

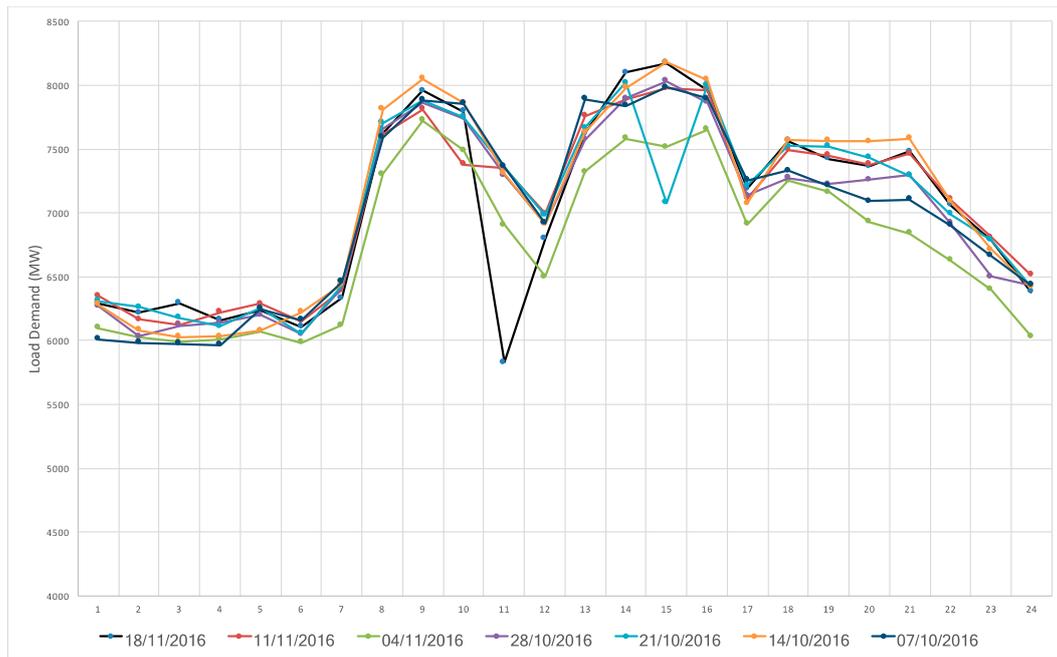


Figure 14. The comparison between Fridays of October and November on Hourly Load Pattern.

Figure 14 shows that a valley load appears at 11:00 on 18 November. This is very unusual when looking at the load pattern on Fridays of the SPC’s electrical load. We suggest there is a certain error from the SPC data collection equipment, thus warning the system to be checked.

4.3. Forecasting for 20 November

After executing the valley and peak load forecasting, we encountered the difficulty of the load pattern forecasting. For two days (16 and 19 November), load patterns of the selected similar days have no big differences. However, the difference is very clear in the case of 20 November (when comparing the five previous Sundays). Thus, we eliminated the date which has the least correlated pattern with others to continue to the next step. Figure 15 shows the pattern’s correlation.

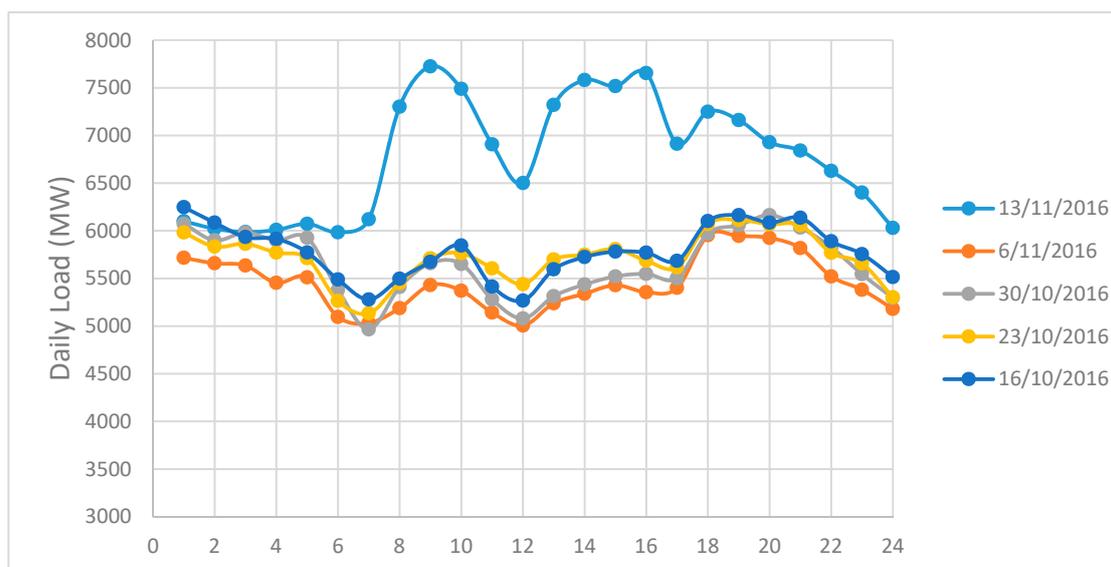


Figure 15. Load pattern of similar days to 20 November 2016.

Following Figure 15, we excluded 13 November from the similar days to avoid errors due to the standardization of the load pattern. After that, we obtained the result shown in Figure 16 and the distribution of forecast errors in Figure 17.

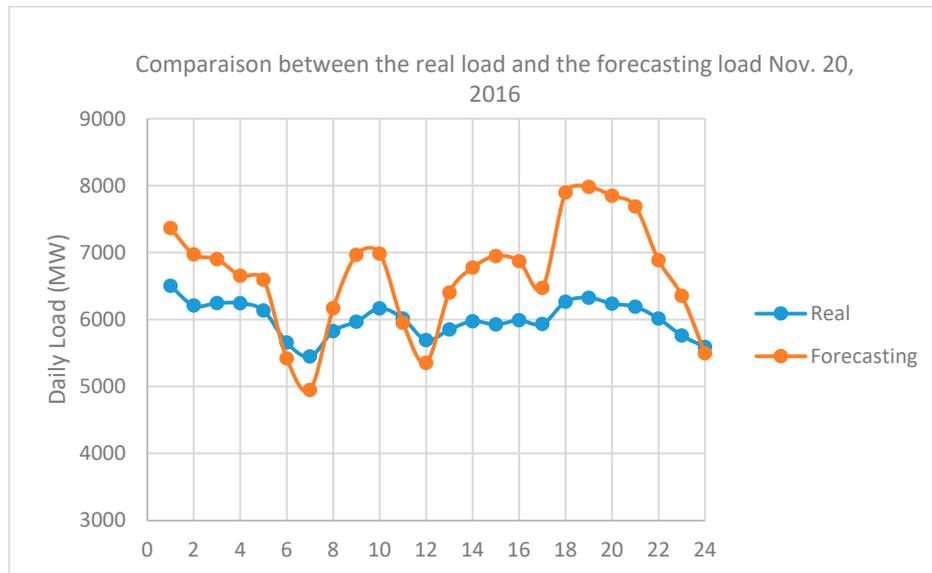


Figure 16. The pattern correlation between the previous similar day of 20 November 2016.

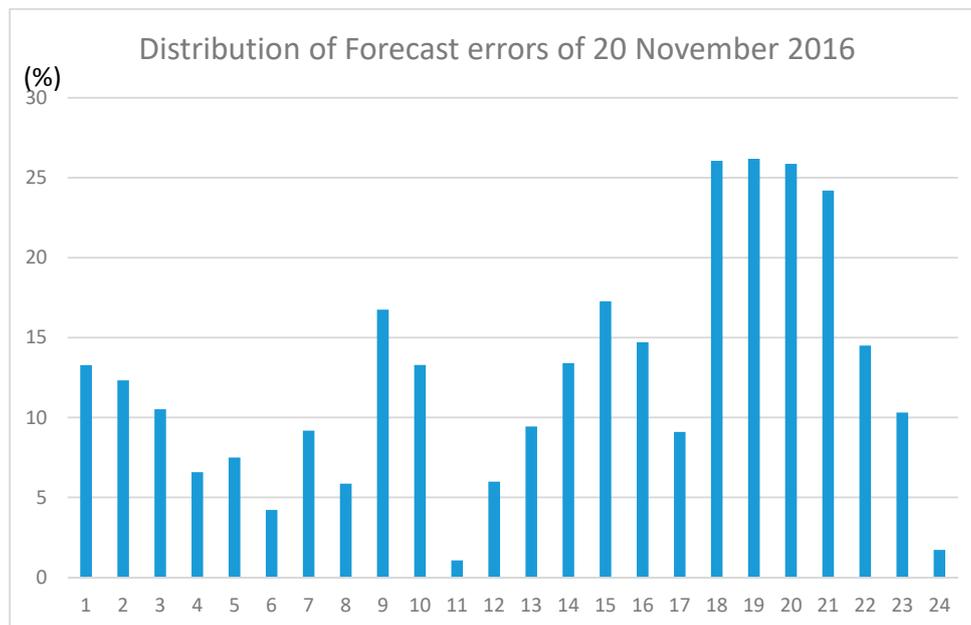


Figure 17. Distribution of forecast errors of 20 November 2016 forecasting.

Unlike other normal tested days, 20 November 2016 gave unfavorable results. With the largest error of 26.18% and the average error of 12.46%, this result is unacceptable to apply to STLF practice. The cause of this unfavorable point is not only due to the temperature or the load pattern of the similar days but also due to unusual events during the holidays. It may be coincidental that some households or load groups consumed electricity on 20 November 2016 due to events taking place differently from the ANN training (this point is due to not enough data for training). We also tried 20 November of the previous years (from 2011 to 2015) to standardize load pattern of this date but these results are also not

positive. Finally, we decided that the hybrid algorithm needs to be improved or the ANN structure needs to be re-selected in the case of holidays overlapping with the weekend.

5. Conclusions

The key goal of this research was to create an effective STLF tool and confirm the GA-PSO combination's prospect. The results of this investigation show that our hybrid algorithm is effective, with small error for 19 November 2016. However, the generalizability of these results is subject to certain limitations (for instance, 16 November 2016) or even unacceptable on 20 November 2016.

This hybrid algorithm can also suggest a collection equipment check to correct the real obtained data from SPC throughout the unusual error periods.

The SPSO operators will need improve to use the basic PSO in the hybrid algorithm. The GA-PSO hybrid algorithm is always better than the basic PSO when comparing errors. Holidays are harder to forecast than the normal days, especially if they coincide with the weekend (20 November 2016 was a Sunday).

There are still many unanswered questions about these high errors and the effectiveness of our hybrid algorithm. This is an important issue for future research. In future investigations, it might be possible to use more neurons in the input layer to represent more local temperatures. Moreover, a further study with more focus on determining hourly load pattern is suggested.

Author Contributions: Conceptualization, T.V.N. and T.-H.D.; Methodology, T.-A.-T.V.; Software, M.-H.P.; Validation, M.-H.P. and T.V.N.; Investigation, D.-Q.N.; Resources, M.-H.P.; Data Curation, V.-H.D.; Writing—Original Draft Preparation, M.-H.P.; Writing—Review & Editing, M.-H.P.; Visualization, N.-T.N.; Supervision, M.-H.P.; Project Administration, M.-H.P.; Funding Acquisition, T.V.N., T.-A.-T.V., T.-H.D., V.-H.D., N.-T.N., D.-Q.N.

Funding: This research received no external funding.

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

References

1. ERAV. *Decision N07/QD-DTDL-2013 on Procedure for Load Forecasting of National Power System*; ERAV: Ha Noi, Vietnam, 2013.
2. Mustapha, M.; Mustafa, M.W.; Khalid, S.N.; Abubakar, I.; Shareef, H. Classification of electricity load forecasting based on the factors influencing the load consumption and methods used: An-overview. In *Proceedings of the 2015 IEEE Conference on Energy Conversion (CENCON)*, Johor Bahru, Malaysia, 19–20 October 2015; pp. 442–447.
3. Baliyan, A.; Gaurav, K.; Mishra, S.K. A Review of Short Term Load Forecasting using Artificial Neural Network Models. *Procedia Comput. Sci.* **2015**, *48*, 121–125. [[CrossRef](#)]
4. Fan, J.; McDonald, J. A real-time implementation of short-term load forecasting for distribution power systems. *IEEE Trans. Power Syst.* **1994**, *9*, 988–994. [[CrossRef](#)]
5. Raza, M.Q.; Baharudin, Z. A review on short term load forecasting using hybrid neural network techniques. In *Proceedings of the IEEE International Conference on Power and Energy (PECon 2012)*, Kota Kinabalu, Malaysia, 2–5 December 2012; pp. 846–851.
6. Ray, P.; Mishra, D.P.; Lenka, R.K. Short term load forecasting by artificial neural network. In *Proceedings of the International Conference on Next Generation Intelligent Systems (ICNGIS)*, Kottayam, India, 1–3 September 2016; pp. 1–6.
7. Hippert, H.S.; Pedreira, C.E.; Souza, R.C. Neural networks for short-term load forecasting: A review and evaluation. *IEEE Trans. Power Syst.* **2001**, *16*, 44–55. [[CrossRef](#)]
8. Singh, A.; Ibraheem, I.; Khatoun, S.; Muazzam, M.; Chaturvedi, D. Load forecasting techniques and methodologies: A review. In *Proceedings of the 2nd International Conference on Power, Control and Embedded Systems (ICPCES)*, Allahabad, India, 17–19 December 2012; pp. 1–10.
9. Mishra, S.; Patra, S.K. Short term load forecasting using neural network trained with genetic algorithm & particle swarm optimization. In *Proceedings of the 1st International Conference on Emerging Trends in Engineering and Technology, ICETET*, Nagpur, Maharashtra, India, 16–18 July 2008; pp. 606–611.

10. Suresh, A.; Harish, K.V.; Radhika, N. Particle Swarm Optimization over back propagation neural network for length of stay prediction. *Procedia Comput. Sci.* **2015**, *46*, 268–275. [[CrossRef](#)]
11. Alobaidi, M.H.; Chebana, F.; Meguid, M.A. Robust ensemble learning framework for day-ahead forecasting of household based energy consumption. *Appl. Energy* **2018**, *212*, 997–1012. [[CrossRef](#)]
12. Muralitharan, K.; Sakthivel, R.; Vishnuvarthan, R. Neural network based optimization approach for energy demand prediction in smart grid. *Neurocomputing* **2018**, *273*, 199–208. [[CrossRef](#)]
13. Capizzi, G.; Sciuto, G.L.; Napoli, C.; Tramontana, E. Advanced and adaptive dispatch for smart grids by means of predictive models. *IEEE Trans. Smart Grid* **2018**, *9*, 6684–6691. [[CrossRef](#)]
14. Garg, H. A hybrid PSO-GA algorithm for constrained optimization problems. *Appl. Math. Comput.* **2016**, *274*, 292–305. [[CrossRef](#)]
15. Zhang, Q.; Ogren, R.M.; Kong, S. A comparative study of biodiesel engine performance optimization using enhanced hybrid PSO—GA and basic GA. *Appl. Energy* **2016**, *165*, 676–684. [[CrossRef](#)]
16. Sahoo, L.; Banerjee, A.; Kumar, A. An efficient GA—PSO approach for solving mixed-integer nonlinear programming problem in reliability optimization. *Swarm Evol. Comput.* **2014**, *19*, 43–51. [[CrossRef](#)]
17. Pham, M.-H.; Vu, T.A.-T.; Dang, T.-H.; Nguyen, D.-Q. An Effective Approach to ANN-Based Short-Term Load Forecasting Model Using Hybrid Algorithm GA-PSO. In Proceedings of the 2018 IEEE International Conference on Environment and Electrical Engineering and 2018 IEEE Industrial and Commercial Power Systems Europe, IEEEIC/I and CPS Europe 2018, Palermo, Italy, 12–15 June 2018. [[CrossRef](#)]
18. Hsu, Y.-Y.; Yang, C.-C. Design of artificial neural networks for short-term load forecasting. Part I: Self-organising feature maps for day type identification. *IEE Proc. C Gener. Transm. Distrib.* **1991**, *138*, 407–413. [[CrossRef](#)]
19. Maifeld, T.; Sheble, G. Short-term load forecasting by a neural network and a refined genetic algorithm. *Electr. Power Syst. Res.* **1994**, *31*, 147–152. [[CrossRef](#)]
20. Ling, S.H.; Leung, F.H.F.; Lam, H.-K.; Lee, Y.-S.; Tam, P. A novel genetic-algorithm-based neural network for short-term load forecasting. *IEEE Trans. Ind. Electron.* **2003**, *50*, 793–799. [[CrossRef](#)]
21. Whitley, D.; Starkweather, T.; Bogart, C. Genetic algorithms and neural networks: Optimizing connections and connectivity. *Parallel Comput.* **1990**, *14*, 347–361. [[CrossRef](#)]
22. Montana, D.J.; Davis, L. Training Feedforward Neural Networks Using Genetic Algorithms. In Proceedings of the 11th International Joint Conference on Artificial Intelligence—Volume 1, Detroit, Michigan, 20–25 August 1989; pp. 762–767.
23. Kennedy, J.; Eberhart, R.C. Particle swarm optimization. In Proceedings of the International Conference on Neural Networks, Perth, Western Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
24. Parsopoulos, K.E.; Vrahatis, M.N. *Particle Swarm Optimization and Intelligence: Advances and Applications*; IGI Global: Hershey, PA, USA, 2010; ISBN 9781615206667.
25. Eberhart, R.C.; Shi, Y. Comparison between Genetic Algorithms and Particle Swarm Optimization. In Proceedings of the 7th International Conference on Evolutionary Programming, San Diego, CA, USA, 25–27 March 1998; pp. 611–616.
26. AlRashidi, M.R.; L-Naggar, K.M.E. Long term electric load forecasting based on particle swarm optimization. *Appl. Energy* **2010**, *87*, 320–326. [[CrossRef](#)]
27. Meissner, M.; Schmuker, M.; Schneider, G. Optimized Particle Swarm Optimization (OPSO) and its application to artificial neural network training. *BMC Bioinform.* **2006**, *7*, 125. [[CrossRef](#)]
28. Ozcan, E.; Mohan, C.K. Particle swarm optimization: Surfing the waves. In Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), Washington, DC, USA, 6–9 July 1999; pp. 1939–1944.
29. Clerc, M.; Kennedy, J. The particle swarm—explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans. Evol. Comput.* **2002**, *6*, 58–73. [[CrossRef](#)]

