Application of hybrid neural networks combined with comprehensive learning particle swarm optimization to short-term load forecasting

Mohammadreza Emarati[1], Farshid Keynia[2], Alireza Askarzadeh[3]

[1] PhD Student, Department of Electrical Engineering,
Graduate University of Advanced Technology, Kerman, Iran
Mr.emarati@yahoo.com
[2] Assistant Professor, Department of Energy Management and Optimization, Institute of Science and High
Technology and Environmental Sciences, Graduate University of Advanced Technology, Kerman, Iran
f.keynia@kgut.ac.ir
[3] Assistant Professor, Department of Energy Management and Optimization, Institute of Science and High
Technology and Environmental Sciences, Graduate University of Advanced Technology, Kerman, Iran
a.askarzadeh@kgut.ac.ir

Abstract:

Short term load forecasting is one of the key components for economical and safe operation of power systems. In competitive environment of electricity market, electricity utilities require more accurate load forecasting strategies to make better decisions on purchasing or generating electricity. This article offers a new method based on machine learning short-term load forecasting which is made up of a two-level feature selection technique and a new forecast engine. The feature selection part uses irrelevancy and redundancy filters to select best sets of input features. The proposed forecast engine is composed of a support vector regression machine, hybrid neural network and comprehensive learning particle swarm optimization. By applying comprehensive learning particle swarm optimization along with hybrid neural networks, the accuracy of forecasting is improved and its error decreases effectively. The proposed strategy is tested on PJM and AEMO electricity markets. The numerical results show the effectiveness and robustness of this method in comparison with recent short-term load forecasting methods.

Keywords: Feature selection, Forecasting engine, Hybrid neural network, Particle swarm optimization, Short-term load forecast.

1- Introduction

Load forecasting helps electrical power systems to make important decisions on purchasing and generating electric power, load switching and substructure improvement. Load forecasts can be divided into three categories: short-term forecasts which are usually from one hour to one week, medium forecasts which are usually from a week to a year, and long-term forecasts which are longer than a year [1]. Short-term load forecasting (STLF) has become a serious issue for electricity supply. It has a significant role in security and reliability which are two essential necessities for proper planning and operation of power systems. A reliable STLF can be practically used in power systems for meeting power consumed continuously. In addition, improving economy of operation and control of power system can be attained by increasing the accuracy of STLF [2-4].

Various methods have been used for load forecasting up to the present time. Majority of these approaches can be broadly divided into two classes: the traditional approaches depicted by time series and the modern intelligent approaches depicted by artificial neural networks (ANN) [5]. Traditional methods include classical multiple linear regression [6], ARMA (automatic regressive moving average) [7], data mining models [8], time-series models [9] and exponential smoothing models [10]. However, the modern intelligent approaches have presented higher performance for non-linear time series than the traditional approaches [4]. Nowadays, artificial intelligence (AI)-based methods such as pattern recognition [11], fuzzy feature selection [12], fuzzy time series [13], neural networks(NN) [14, 15], and fuzzy NNs [16] are highly regarded as powerful computational tools for solving the problem of load forecasting [17]. Although available approaches have provided significant enhancement throughout the years, more precise

and robust load forecasting methods are still needed.

In this study, a new strategy based on machine learning short-term load forecasting (ML-STLF) is proposed. This method employs a coalition of machine learning (ML) for an efficient two-level feature selection and Support Vector Regression (SVR) for initial training of the nonlinear mapping function.

The introduced forecast engine employs three-stage hybrid neural network (HNN) and comprehensive learning particle swarm optimization (CLPSO) simultaneously. This composition helps the forecast engine to create a more precise prediction. Using CLPSO because of high global search ability and its high capability in combination with local search methods can have an important role in enhancing the precision of the forecasting engines. The efficiency of proposed strategy is proved by using some numerical experiments. The main parts of this article can be summarized in two sections:

(1) Most of previous studies emphasizes on the forecast strategy, whereas they don't pay attention to design of the input vector. Here, an efficient data preparation (normalization and shuffling) and a novel ML approach containing two-level feature selection technique is employed to select the privileged candidate inputs for the proposed forecast engine. The first level filters irrelevant inputs while the second one removes redundant candidate features. Inputs that have been able to pass through this two-level feature selection are applied to the forecast engine.

(2) A new powerful and efficient STLF engine is proposed. This engine is made up of a combination of three interconnected core units. The first part is an auxiliary predictor which employs a SVR machine to produce initial forecast of target variables. The second and the main part is constructed by an HNN which uses different training functions in each stage. And the last part, which is a cooperative tool for HNN, applies CLPSO method to improve the learning capability of HNN.

Following this introduction, the remaining parts of this paper is organized as follows: In Section 2, the proposed ML-STLF approach is introduced. The obtained numerical results of the proposed approach are indicated and compared with other new STLF methods In Section 3. And Section 4 presents the conclusion.

2- Description of the proposed ML-STLF strategy

Fig(1) indicates the general structure of proposed ML-STLF approach. As can be seen, the intended method is made by combining a two-level feature selection part with a STLF engine, as explained in Sections 2.1 and 2.2, respectively. Also, in sections 2.3 and 2.4 detailed explanations about CLPSO and its integration with HNN are given.
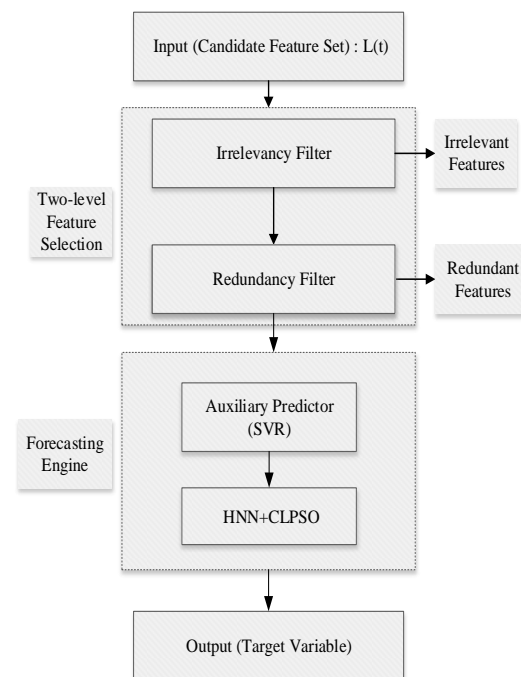
Fig(1): Structure of proposed ML-STLF strategy.

2-1- Two-level feature selection

A key issue for the achievement of any forecast strategy is a suitable choice of effective input variables. Feature selection can simplify the learning process of the forecasting engine and improve its generalization capability for unseen data. At the stage of feature selection at first, irrelevant features are removed and then redundant features are executed to create a subset covering the best input features. The best subset contains the least number of key features, which are vital to forecast more precisely.

Using feature selection in prepossessing phase can reduce the dimension of input variables in an effective way. In most of the previous studies like [7, 14] the authors paid attention to forecasting models and different heuristic methods were applied for selecting input variables. Details of correlation approach has been explained in [18] and it has been

employed in the feature selection part of STLF in [4] and [19]. In correlation approach, the relevancy among each candidate input and the target value (in this article, load of the next hour) is calculated and a depending factor determines the relevancy between them. More correlation between the target variable and a candidate input results in more chance of that variable to be selected as an input feature. The depending factor among two variables like X and Y, denoted by $\rho_{x,y}$, with standard deviations σ_x and σ_y, is computed from following relation:

$$\rho_{X,Y} = \frac{\text{cov}(X,Y)}{\sigma_X \sigma_Y} \qquad (1)$$

where $cov(X,Y)$ is the covariance of X and Y. The absolute value of depending factor (which is a number between 0 and 1) indicates the amount of linear reliance among the variables. The two-level feature selection removes the irrelevant and redundant input features respectively. By decreasing the number of inputs to the forecast engine, the number of optimization variables reduces. This action not only improves the training accuracy of forecasting engine but also increases its training speed.

If correlation index between the output feature and a candidate variable is more than a relevancy threshold TH1, then this candidate is regarded as the relevant feature of the forecast process. Other candidate inputs with correlation index less than TH1 are regarded as irrelevant features as shown in Fig. 1. Next, for the remaining candidates, a cross-covariance test is executed. Greater value of correlation between two selected features indicates more common information between. In other words, these features have a remarkable level of redundancy. If the correlation index across any two candidate variables is less than a predetermined value TH2, then both variables are selected; otherwise, only the variable with the greatest correlation according to the target value is remained while the other is not considered any further. Selected candidate features are regarded as the final entries of the ML-STLF engine as indicated in Fig. 1. Next, the proposed forecast engine must carry out the prediction procedure. Its efficient performance in short-term load forecasting utilization is one of the key ideas of this paper.

2-2- Proposed HNN

The main propose of this section is improvement of forecasting model through learning from the selected candidate features which are obtained by two-level feature selection strategy. SVR is a supervised machine learning method which has a high learning capability. SVR models are able to deal with different kinds of data patterns. For SVR, the tendency of the data which may present fluctuation or sustained increasing or decreasing types does not make much difference. Generally, SVR is applied for solving non-linear regression and time series problems. A computational regression function in a high dimensional feature space forms the main structure of SVR. This function plots the input data to the higher dimensional space. In other words, the basic notion of the SVR is to map the original data into a higher dimensional feature space using a nonlinear process. In this strategy, the structural risk minimization inductive principle is implemented to generate limited numbers of learning patterns. In this article, as shown in Fig (2), SVR is applied as an auxiliary predictor. Additional information about SVR and its operation can be found in [20]. Furthermore, SVR has been used for short term load forecasting in [7].

In this way, the SVR machine receives the input features selected by two-level feature selection and forwards the output predicted values along with input features to HNN.

A properly designed composition of different NNs can strongly improve their capability of learning in modelling a complicated operation. For instance, several various (cascaded and parallel) structures for combination of different NNs with enhanced capability are presented in [21-23]. An efficient HNN, has been proposed in [24] for electricity price forecasting. All three NNs applied in HNN of [24] have similar structure of multi-layer perceptron with a hidden layer. MLP is a high yield structure of forecasting neural networks. Furthermore, in accordance with Kolmogorov's theory, by selecting appropriate number of neurons, only one hidden layer is enough for MLP to deal with a problem [25]. Therefore, one hidden layer considered to apply in the

structure of each NN. As the forecast engine of electricity load prediction, a new HNN with the architecture shown in Fig (2) is proposed in this paper.
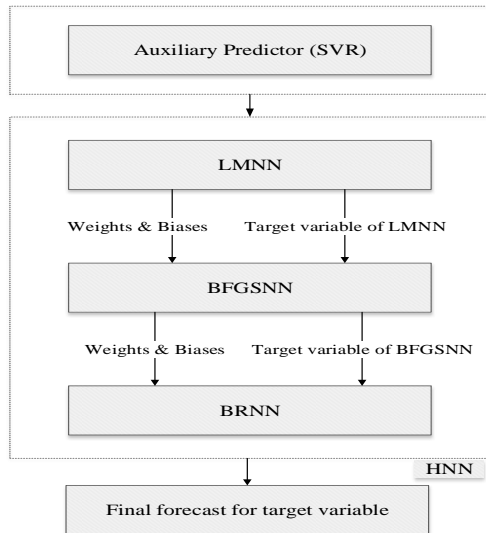

Fig (2): Architecture of Hybrid Forecasting Engine.

As shown in Fig (2), after the SVR machine performs the preliminary forecast, the results of initial prediction along with the selected features are applied to the first stage of HNN (LMNN). Also, in each stage of HNN a specific NN has been used. Moreover, in this structure each NN transmits two vectors of results to the subsequent NN. Only the first NN should begin with an initial set of random values for the adjustable parameters. In other expression, each trained NN hands its obtained proficiency to the following NN. Thus, instead of beginning from a random point, the training process of each NN can be started from the place, that its former NN has been reached. Hence, the next NN can directly employ the weight and bias values of former one, because all NNs of the HNN have identical number of inputs, hidden and output neurons, the obtained knowledge of the previous one can be improved. Furthermore, the second sets of consequences transferred among NNs are the prediction of target variables. By this manner, all NNs also have a preliminary forecast of target as an input value, which gets great benefit to improve accuracy of the prediction.

In addition, by suitable selection of different MLP training algorithms, the HNN can be learned much more than a single NN. Further discussions for the most efficient NN training mechanisms and their mathematical details can be found in [25]. In view of the above

discussion, an improved version of HNN forecast engine is proposed for ML-STLF in this article. Three kinds of training algorithms have been considered for the HNN, owning Levenberg–Marquardt Neural Network (LMNN), Broyden Fletcher Goldfar Shanno neural network (BFGSNN), and Bayesian regularization neural network (BRNN). In [3], it is explained that the best results are related to cascade MLPs which benefit from LMNN In the beginning of the training stage. In this way, MLP can quickly learn about the problem and its training error rapidly decreases. LMNN is a fast learning algorithm. Therefore As seen in Fig (2) LMNN has been selected as the first NN of HNN. BFGSNN is known as the most powerful quasi-Nowton method for training NNs. If this algorithm starts the learning process from a suitable initial point, it will show greater ability to find better solutions in the search space. Thus, in the second stage BFGSNN has been used for detecting superior weights and biases in the solution space. In addition, BR learning algorithm minimizes a combination of squared errors and weights and then determines the correct combination so as to produce a network that generalizes well [26]. Therefore, BR training mechanism is considered as the last NN for final tuning of the adjustable parameters and getting the maximum training efficiency.

2-3- CLPSO Algorithm

Particle swarm optimization (PSO) is a global minimization algorithm, which is a powerful tool for solving high-dimensional problems. Each potential solution is considered as a particle, which tries to make its current position better than its former position. In other words, the position of each particle depends on its current position and a velocity vector, which is defined for the same particle. The position and velocity of particle i in a physical d dimensional search space are expressed as the vectors of $X_i = [X_{i1}, X_{i2},..., X_{id}]$ and $V_i = [V_{i1}, V_{i2},..., V_{id}]$, its individual best position is $P_{besti} = [X_{i1pbest}, X_{i2pbest},..., X_{idpbest}]$, and Its global best position is $G_{best} = [X_{1gbest}, X_{2gbest},..., X_{ngbest}]$. In each iteration, the velocity and position of particle i is updated as follows :

$$V_i(k+1) = \omega V_i(k) + c_1 r_1 (P_{besti} - X_i(k)) + c_2 r_2 (G_{best} - X_i(k)) \tag{2}$$

$$X_i(k+1) = X_i(k) + V_i(k+1) \tag{3}$$

where $V_i(k)$ is the velocity of particle i at iteration k; ω denotes inertia weight factor; C_1, C_2 are acceleration coefficients; R_1, R_2 are uniformly distributed random number among 0 and 1; $X_i(k)$ is the position of particle i at k iteration; P_{besti} is the best position of particle i until iteration k; and G_{best} is the global best position of all particles until iteration k.

In this paper, an advanced version of PSO called comprehensive learning particle swarm optimization (CLPSO) [27] has been used. CLPSO has demonstrated good performance in high dimensional problems. In this algorithm for updating the velocity of each particle, P_{best} of all particles will get involved instead of using P_{best} of each particle for the same particle. Thus, the equation (2) is changed as follows:

$$V_i(k+1) = \omega V_i(k) + c_1 r_1 (P_{best}^{f_i(d)} - X_i(k)) + c_2 r_2 (G_{best} - X_i(k)) \quad (4)$$

where $f_i(d) = [f_i(1), f_i(2), ..., f_i(d)]$ is a vector, which determines the particle i should use P_{best} of which particle in each dimension. CLPSO is completely explained in [27].

In the process of velocity updating, the value of some parameters such as ω, C_1, C_2 should be decided beforehand. Experimental results show that more convergence can be achieved by reducing the inertia weight ω in each iteration. Therefore, the value of ω is reduced linearly as the iteration k proceeds and figured as follows:

$$\omega = \omega_{max} - \frac{(\omega_{max} - \omega_{min})iter}{iter_{max}} \quad (5)$$

where ω_{max} is final inertia weight; ω_{min} is initial inertia weight; $iter$ is current iteration number; and $iter_{max}$ is maximum iteration number. In this study, all parameters of CLPSO are fine-tuned based on the proposed method of [27].

2-4- Combination of HNN and CLPSO for designing the Proposed ML-STLF Engine

The hybridization approach of HNN and CLPSO to create the suggested ML-STLF engine is illustrated in Fig (3). Although LMNN, BFGSNN and BRNN benefit from high efficient training algorithms, they explore the solution space in a particular direction. In this manner, these training algorithms may be got stuck in a local minima without finding the global minima. However, exploration capability of CLPSO algorithm can broadly investigate the solution space in different directions. Therefore, the proposed training method is more possible to escape from the local minima.

At first LMNN is trained by the LM learning algorithm. To avoid the over fitting problem, early stopping condition is used in the training procedure of all NNs, as shown in Fig (3) the obtained weight and bias values are transferred to CLPSO. Then, CLPSO continues the process of training by modelling this process as an optimization problem.

Weights and biases can easily be transferred because NNs of HNN and CLPSO component have equal training and validation samples. The objective function of the optimization problem is the error function of LMNN, which should be minimized. In other words, CLPSO tries to further minimize the validation error of LMNN after its learning algorithm is completed.

The decision variables of the optimization problem (particles of CLPSO) are potential solutions for weight and bias vectors of LMNN. Generally, the position of the particles in CLPSO are initialized randomly:

$$\text{The initial swarm of CLPSO} = X_{1,0}, X_{2,0}, ..., X_{NP,0} \quad (6)$$

The structure of each particle can be shown as follows:

$$X_i = W \& B_{LM} \quad (7)$$

where $W \& B_{LM}$ are weight and bias vectors, which are initialized randomly.

where NP is the number of particles in solution space.

In (6), $X_{1,0}, X_{2,0}, ..., X_{NP,0}$ demonstrate the initialized positions of the NP particles of CLPSO.

In (7), $W \& B_{LM}$ are obtained results of the weight and bias vectors by LMNN.

Also, the initial velocity vector of NP particles are initialized randomly within allowed specific ranges. Then, iterative particles of CLPSO change their positions and explore the solution space thoroughly. Here, if the value of the validation error does not reduce after four consecutive iterations, the search process will be terminated. Next, the best particle of CLPSO (which is covering a weight and bias vector) is given back to LMNN; this vector is regarded as the final weights and biases of LMNN. Here, the learning process of LMNN is completed.

Then, as seen in Fig. 3 BFGSNN accepts the final weights and biases of LMNN as the initial values to start the learning process. For BFGSNN, training proses is the same as LMNN. Similarly, after finishing the learning process of BFGSNN, its final weights and biases are transferred to BRNN. Starting from this point, the learning process of BRNN is executed similar to training process of the last two NNs. After completion of the training process of BRNN, all NNs of HNN are trained. At this point, the proposed ML-STLF engine is learned and prepared for the forecast. In this way, LMNN using its obtained weight and bias values, produces a forecast for the input series (Here, the electricity load of the previous hours) $L(t)$, which is applied to BFGSNN. Also, BFGSNN and BRNN generate their forecast values until the last prediction of $L(t)$ is obtained from BRNN.
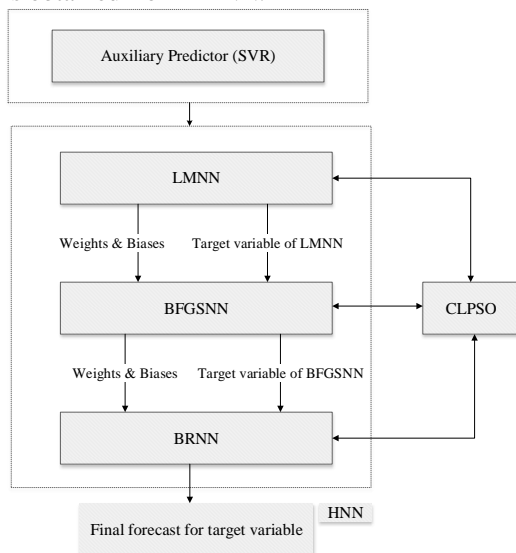
Fig(3): Architecture of the proposed ML-STLF engine.

It should be mentioned that the numerical fine-tuning of the proposed ML-STLF method's adjustable parameters, including TH1 and TH2 which are used in the two-level feature selection process and the number of applied neurons in each NN's hidden layer, has been carried out by a computationally efficient cross-validation method described In [28].

3- Numerical Results

There are two prevalent types of STLF strategies in electric power systems: hourly (next hour) and daily (next day) load forecasting [29]. Real-time and future electricity markets benefit from hourly and daily load forecasting,

respectively. For prediction of the next hour's load, the data updates at the end of each time interval. Also, prediction of the next day's load is attained by substitution of forecasted values for input variables called recursion method [4]. This procedure is repeated until the next day's load forecast value is achieved. In this research, hourly load forecast and one-day ahead (as the prediction horizon) have been considered.

The proposed ML-STLF strategy has been examined on load forecast of day-ahead electricity market of PJM. PJM is a well-known site in electricity market that coordinates the movement of wholesale electricity in different states of US. Our test case includes day-ahead demand historical data over the period 2011-2012 which can be found at [30]. The numerical experimentations, which are presented in the following, are designed to show the high performance of the proposed ML-STLF engine and evaluate its effectiveness in a comparative manner. Fig (4) shows the correlation of candidate features of 500 hours ago according to the output for December 12, 2012.

In Fig (4), the horizontal axis indicates (candidate according to 1 hour ago) up to (candidate according to 500 hours ago) and the vertical axis demonstrates the absolute amount of correlation coefficients. The greater amount of correlation means more relation between corresponding candidate and the target vector's value.
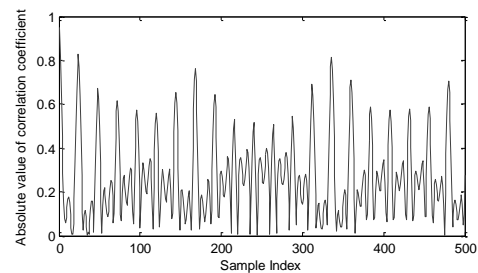
Fig(4):Correlation of candidate features of 500 hours ago according to the output for December 12, 2012.

Results of two-level feature selection process with different values of TH1 and TH2 are presented in Table (1). It can be seen by increasing TH1, the initial level of feature selection selects candidates that are more relevant and by decreasing TH2 the secondary level of feature selection filters more redundant candidates. However, if these two terms are satisfied simultaneously, number of selected variables will decrease. So, determination of TH1 and TH2 is a trade-off between quality and number of features. In this test case, the best values of TH1 and TH2 by the cross-validation method are determined 0.6 and 0.9,

respectively. In addition, in this study, the proper number of neurons for hidden layer of all NNs for the minimum of the validation error by cross validation technique, occurs at NH=10.

Table (1): Number of Selected Features

TH1	TH2	Number of selected inputs
0.7	1	13
0.7	0.9	6
0.7	0.8	4
0.6	1	33
0.6	0.9	14
0.6	0.8	7
0.5	1	69
0.5	0.9	28
0.5	0.8	13

In this study, training samples (subsets of data) from electricity load related to 39 days before the forecast day (results in 39*24=936) have been considered. The validation set has been randomly selected from training samples. Moreover, the 10% of total samples are selected as validation set and the rest are considered as train set.

In Table(2), the first and third rank selected features (L_{t-1} and L_{t-2}) consist of information about earlier hours. The next more effective features contain information about the daily seasonality (one day ago) such as L_{t-24} and L_{t-23}, while the later terms are related to weekly seasonality (one to three weeks ago). Farther days and weeks have less correlation with the target hour and so are not considered here. Note that by increasing input candidates the selected candidates provided in Table (2) will not be changed, indicating that more earlier candidates have less information value than others. In other words, the latest selected candidate features (as validation set) have more resemblance to the prediction horizon.

Table (2): Selected features for PJM on December 12, 2012.

Rank	Selected Feature	Rank	Selected Feature
1	L_{t-1}	8	L_{t-168}
2	L_{t-24}	9	L_{t-337}
3	L_{t-2}	10	L_{t-167}
4	L_{t-336}	11	L_{t-360}
5	L_{t-23}	12	L_{t-169}
6	L_{t-25}	13	L_{t-480}
7	L_{t-335}	14	L_{t-312}

The measurement of forecasting accuracy is accomplished by Mean Absolute Percentage Error (MAPE), which can be computed as follows:

$$MAPE(\%) = \frac{1}{N} \sum_{i=1}^{N} \frac{\left| L_A^i - L_F^i \right|}{L_A^i} \times 100 \quad (8)$$

where: L_A is the actual load, L_F is the forecasted load, N is the prediction horizon (the amount of hours in the prediction period) and i is the hour index. Another criterion for evaluating forecast accuracy is mean absolute error (MAE), which is defined by the following equation:

$$MAE = \frac{1}{N} \sum_{i=1}^{N} \left| L_A^i - L_F^i \right| \quad (9)$$

The proposed ML-STLF approach covers the prediction period by one-hour steps up to reaching out to the end of forecast period. Here, the prediction horizon is considered 24 hours. For this case study, obtained results (MAPE and MAE) from each NN of the proposed HNN are represented in Table 3. As shown in Table 3, the values of MAPE and MAE have been reduced from 2.03% and 0.74 GW to 1.48% and 0.54 GW, respectively. This table specifies the role of each stage of the forecast engine in reducing the prediction error.

Table (3): MAPE (%) and MAE (MW) results of proposed forecast engine on December 12, 2012.

NN output	MAPE (%)	MAE (MW)
LMNN+CLPSO	2.03	748
BFGSNN+CLPSO	1.85	717
BRNN+CLPSO	1.48	548

To compare proposed method with the other hybrid approaches, results of different hybrid methods are demonstrated in Table 4. In all of mentioned methods in Table (4) data pre-processor and feature selection have been used. NN is a one hidden layer perceptron neural network, HNN2 is a hybrid neural network with two stages and HNN3 is a hybrid neural network with three stages. In Table (4) trial period is December 12, 2012. In this table, it is observed that the proposed ML-STLF engine benefiting from CLPSO has been able to decrease the values of MAPE and MAE by 54% and 37% (respectively) in comparison with HNN3. Also, In Fig (5), actual load, forecasted load and forecast error on the same day have

been shown. It can be seen from Fig (5) that the proposed strategy can give an accurate forecast. Also, the convergence plot of CLPSO algorithm to improve the results obtained by LMNN is depicted in Fig.6. This figure shows the value of Mean Squared Error (MSE) for normalized data. The optimized results in this stage are given to the next neural network (BFGSNN) for continuing the training process of the forecast engine.

Table (4): MAPE (%) and MAE (MW) results of suggested ML-STLF and three other methods for December 12, 2012.

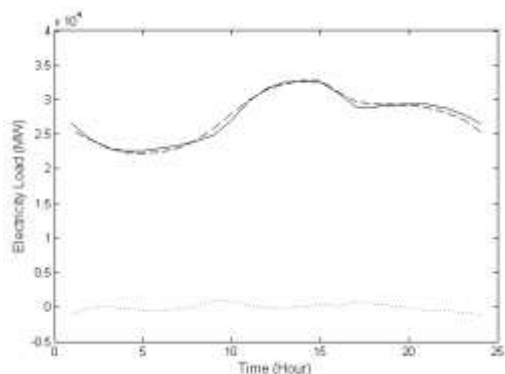| Forecast method	MAPE (%)	MAE (MW)
NN	4.00	1043
HNN2	3.64	939
HNN3	3.27	872
Proposed (ML-STLF)	1.48	548

Fig (5): Actual load (solid line), forecasted load of proposed method (dashed line), and its error (dotted line) for PJM in December 12, 2012.
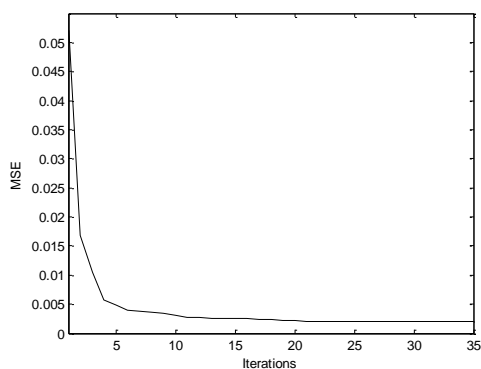

Fig (6): The convergence plot of CLPSO algorithm used to improve the results obtained by LMNN.

To demonstrate the capability of the suggested STLF, in Table (5) four test weeks of the year 2012 from the PJM electric power market have been examined. The four test weeks are February 4 to February 11, May 5 to May 12, August 4 to August 11, and November 10 to November 17. In Table 5, the proposed forecast engine (ML-STLF) is compared with three MLP neural networks which are trained by LM, BFG, and BR learning algorithms, respectively. These nonlinear forecast methods have been used frequently in many articles to predict electricity load demand, electricity price and wind power. As shown in Table 5, the proposed forecast engine has better forecast accuracy than the other forecast methods. As presented in the last row of Table 5 the average MAPE and MAE for the proposed ML-STLF method are 1.34% and 0.526 GW respectively. However, the other mentioned methods with the same conditions have shown the average value of MAPE and MAE 3% and 0.777 GW respectively. The proposed method using CLPSO in its training mechanism can reach better solutions because of its ability to escape from local minima. In addition, the proposed forecast engine is composed of three consecutive NNs to improve the obtained knowledge during the forecast process while the three other methods use one NN for prediction.

In Table 6, the obtained results are compared with results of reference [31]. These results demonstrate good performance of proposed method in comparison with the other mentioned methods. The structure of Additive model presented in [31] is based on a three layer feed-forward NN which employs Levenberg–Marquardt algorithm for training. For speeding up the training process, the hyperbolic tangent function has been applied for hidden neurons and output neurons.

The results presented in Table (6) were obtained using historical data from Australian Energy Market Operator (AEMO) website [32] since October 2008 to March 2009. It can be seen from Table 6 that the proposed strategy has reached more precise results than the other mentioned methods. In the last row of Table (6), it is found out that the obtained average values of MAPE and MAE from the proposed ML-STLF method are 8% and 19% (respectively) less than the Additive model. These results can show the satisfactory performance of the proposed method compared with the recent strategies.

The average computation time taken for training the proposed ML-STLF model is about 15 minutes. The hardware configuration of the computer used is Intel Core i5 processor with

2.30 GHz CPU, 4 GB RAM and the operating system used is Windows 7 ultimate.

4- Conclusion

Load forecasting is very important for secure operation of power systems. This paper presents a new strategy for STLF. In this strategy, a two-level feature selection technique and a new hybrid forecasting engine are employed. The two-level feature selection technique is designed for removing both irrelevant and redundant candidate inputs. Thus, the most instructive features are applied to forecast engine. The

proposed forecasting engine is a hybrid neural network, which benefits from good global search capability of CLPSO. In addition, by using CLPSO alongside NN high convergence-learning algorithms can have an important role for enhancing the accuracy of forecasting .

The proposed strategy has been examined in PJM and AEMO electricity markets. The results illustrate that the proposed model has a high level of effectiveness and robustness. Also, results of proposed strategy show more accuracy compared with methods of recent papers.

Table (5): MAPE (%) and MAE (MW) results of MLP with LM, MLP with BFG, MLP with BR and proposed ML-STLF engine

Test week	MLP with LM		MLP with BFG		MLP with BR		Proposed (ML-STLF)	
	MAPE	MAE	MAPE	MAE	MAPE	MAE	MAPE	MAE
February	4.04	1035	4.67	1194	3.02	756	1.23	501
May	4.32	1075	5.07	1274	3.06	791	1.47	556
August	3.84	975	5.17	1318	3.03	780	1.22	499
November	3.90	995	5.23	1365	3.04	784	1.45	550
Average	4.02	1020	5.04	1287.75	3.04	777.75	1.34	526.5

Table (6): Monthly comparison of performance.

Month	ANN [31]		Hybrid [31]		Additive Model [31]		Proposed (ML-STLF)	
	MAPE	MAE	MAPE	MAE	MAPE	MAE	MAPE	MAE
October 2008	2.57	134.87	2.15	121.83	1.66	88.55	1.52	74.31
November 2008	2.63	140.52	2.12	123.50	1.74	94.33	1.67	90.74
December 2008	2.49	126.39	2.17	116.34	1.55	79.89	1.51	72.63
January 2009	2.81	168.04	2.14	126.73	1.88	110.21	1.71	93.52
February 2009	2.37	139.68	1.95	119.07	1.64	96.48	1.44	59.11
March 2009	2.29	123.21	1.94	116.49	1.59	87.45	1.43	57.81
Average	2.53	138.79	2.08	120.66	1.68	92.82	1.54	74.68

References

[1] M. Q. Raza and A. Khosravi, "A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings," Renewable and Sustainable Energy Reviews, Vol. 50, pp. 1352-1372, 2015.

[2] R. Hu, S. Wen, Z. Zeng, and T. Huang, "A short-term power load forecasting model based on the generalized regression neural network with decreasing step fruit fly optimization algorithm," Neurocomputing, Vol. 221, pp. 24-31, 2017.

[3] N. Amjady and F. Keynia, "Mid-term load forecasting of power systems by a new prediction method," Energy Conversion and Management, vol. 49, pp. 2678-2687, 2008.

[4] S. Kouhi and F. Keynia, "A new cascade NN based method to short-term load forecast in deregulated electricity market," Energy Conversion and Management, Vol. 71, pp. 76-83, 2013.

[5] H. Nie, G. Liu, X. Liu, and Y. Wang, "Hybrid of ARIMA and SVMs for Short-Term Load Forecasting," Energy Procedia, Vol. 16, pp. 1455-1460, 2012.

[6] F. M. Tuaimah and H. M. A. Abass, "Short-term electrical load forecasting for Iraqi power system based on Multiple Linear Regression method," International Journal of Computer Applications, Vol. 100, 2014.

[7] A. Kavousi-Fard, H. Samet, and F. Marzbani, "A new hybrid modified firefly algorithm and support vector regression model for accurate short term load forecasting," Expert systems with applications, Vol. 41, pp. 6047-6056, 2014.

[8] S. Park, S. Ryu, Y. Choi, and H. Kim, "A framework for baseline load estimation in demand response: Data mining approach," in Smart Grid Communications (SmartGridComm), 2014 IEEE International Conference on, 2014, pp. 638-643.

[9] N. Amjady, "Short-term hourly load forecasting using time-series modeling with peak load estimation capability," Power Systems, IEEE Transactions on, Vol. 16, pp. 798-805, 2001.

[10] J. W. Taylor, "Short-term electricity demand forecasting using double seasonal exponential smoothing," Journal of the Operational Research Society, Vol. 54, pp. 799-805, 2003.

[11] M. F. Sabahi and S. Sarrfian, "A Cell Phone Positioning Method Based on the Intelligent Received Signal Strength Pattern Recognition in GSM Network for Implementing Mobile Advertisement Services," Computational Intelligence in Electrical Engineering, Vol. 4, pp. 25-42, 2013.

[12] H. Nosrati nahook and M. Eftekhari, "A New Method for Feature Selection Based on Fuzzy Logic," Computational Intelligence in Electrical Engineering, Vol. 4, pp. 71-84, 2013.

[13] Y. Hong-Tzer and H. Chao-Ming, "A new short-term load forecasting approach using self-organizing fuzzy ARMAX models," Power Systems, IEEE Transactions on, Vol. 13, pp. 217-225, 1998.

[14] N. Ding, C. Benoit, G. Foggia, Y. Bésanger, and F. Wurtz, "Neural network-based model design for short-term load forecast in distribution systems," IEEE Transactions on Power Systems, Vol. 31, pp. 72-81, 2016.

[15] K. Tayebeh, M. Ataei, and P. Moallem," Wind Speed Prediction Based on Chaos Theory using RBF Neural Networks," Computational Intelligence in Electrical Engineering, Vol. 7, pp. 87-96, 2016.

[16] G.-C. Liao and T.-P. Tsao, "Application of a fuzzy neural network combined with a chaos genetic algorithm and simulated annealing to short-term load forecasting," IEEE Transactions on Evolutionary Computation, Vol. 10, pp. 330-340, 2006.

[17] N. Amjady and F. Keynia, "Short-term load forecasting of power systems by combination of wavelet transform and neuro-evolutionary algorithm," Energy, Vol. 34, pp. 46-57, 1// 2009.

[18] P. J. Santos, A. G. Martins, and A. J. Pires, "Designing the input vector to ANN-based models for short-term load forecast in electricity distribution systems," International Journal of Electrical Power & Energy Systems, Vol. 29, pp. 338-347, 5// 2007.

[19] N. Amjady, "<Short-Term Hourly Load Forecasting Using.pdf>," pp. 798-805, 2001.

[20] S. Haykin, Neural networks: a comprehensive foundation: Prentice Hall PTR, 1994.

[21] H. Taherian, I. Nazer-Kakhki, M. R. Aghaebrahimi, M. Farshad, and S. R. Goldani, "Short Term Price Forecasting in Electricity Market Considering the Effect of Wind Units' Generation," Computational Intelligence in Electrical Engineering, Vol. 5, pp. 105-122, 2014.

[22] F. Shu, J. R. Liao, R. Yokoyama, C. Luonan, and L. Wei-jen, "Forecasting the Wind Generation Using a Two-Stage Network Based on Meteorological Information," Energy Conversion, IEEE Transactions on, Vol. 24, pp. 474-482, 2009.

[23] S. Z. Seyyedsalehi and S. A. Seyyedsalehi, "Bidirectional Layer-by-layer Pre-training Method," Computational Intelligence in Electrical Engineering, vol. 6, pp. 1-10, 2015.

[24] N. Amjady, A. Daraeepour, and F. Keynia, "Day-ahead electricity price forecasting by modified relief algorithm and hybrid neural network," Generation, Transmission & Distribution, IET, Vol. 4, pp. 432-444, 2010.

[25] C. P. Rodriguez and G. J. Anders, "Energy price forecasting in the Ontario competitive power system market," Power Systems, IEEE Transactions on, Vol. 19, pp. 366-374, 2004.

[26] N. Amjady, Introduction to intelligent systems. Semnan,Iran: Semnan University Press, 2002.

[27] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," Evolutionary Computation, IEEE Transactions on, Vol. 10, pp. 281-295, 2006.

[28] N. Amjady and F. Keynia, "Day-ahead price forecasting of electricity markets by a new feature selection algorithm and cascaded neural network technique," Energy Conversion and Management, Vol. 50, pp. 2976-2982, 2009.

[29] N. Amjady, "<Short-Term Bus Load Forecasting of Power.pdf>," pp. 333-341, 2007.

[30] PJM. Available: http://www.pjm.com

[31] F. Shu and R. J. Hyndman, "Short-Term Load Forecasting Based on a Semi-Parametric Additive Model," Power Systems, IEEE Transactions on, Vol. 27, pp. 134-141, 2012.

[32] AEMO. Available: http://www.aemo.com.au/Electricity/Data/Price-and-Demand/Aggregated-Price-and-Demand-Data-Files