Energy 174 (2019) 460–477

Contents lists available at ScienceDirect

Energy

journal homepage: www.elsevier.com/locate/energy

Review

A hybrid method based on neural network and improved environmental adaptation method using Controlled Gaussian Mutation with real parameter for short-term load forecasting

Priyanka Singh [a,*], Pragya Dwivedi [a], Vibhor Kant [b]

[a] Computer Science and Engineering Department, MNNIT Allahabad, Allahabad, 211004, India
[b] Department of Computer Science and Engineering, LNMIIT, Jaipur, 302031, India

A R T I C L E I N F O

Article history:
Received 1 June 2018
Received in revised form
18 February 2019
Accepted 19 February 2019
Available online 22 February 2019

Keywords:
Electricity load forecasting
Environmental adaptation method
Neural network
Gaussian mutation

A B S T R A C T

Load forecasting is a challenging task in power markets that require attention in generating accurate and stable load to deal with planning and management strategies. In past few years, several intelligence-based models have been introduced for precise load forecast. Among them, artificial neural network (ANN) seems more effective and capable to handle the non-linear behavior of load and generates an accurate forecast. However, it suffers from overfitting problem thus reducing the accuracy of load forecasts. To overcome this problem, a hybrid methodology namely ANN-IEAMCGM-R, for short-term load forecast is proposed in this paper. ANN is integrated with an enhanced evolutionary algorithm (IEAMCGM-R) to find optimal network weights. This evolutionary algorithm is composed of improved environmental adaptation method with real parameters (IEAM-R) and our proposed Controlled Gaussian Mutation (CGM) method to bring greater diversity within the population resulting in a higher convergence of solutions.

The electric load data from the New England Power Pool (NEPOOL, ISO New England) and Australian Energy Market Operator (New South Wales (NSW), Australia) have been used to illustrate the efficacy of the proposed hybrid methodology. Results show that the proposed hybrid methodology generates higher accuracy than other state-of-the-art algorithms.

© 2019 Elsevier Ltd. All rights reserved.

Contents

* Corresponding author.
E-mail addresses: 2015rcs53@mnnit.ac.in (P. Singh), pragyadwi86@mnnit.ac.in (P. Dwivedi), vibhor.kant@lnmiit.ac.in (V. Kant).

https://doi.org/10.1016/j.energy.2019.02.141
0360-5442/© 2019 Elsevier Ltd. All rights reserved.

1. Introduction

Several operating decisions such as management and planning, scheduling, load dispatching, and so on are made on the basis of load forecasting [1]. Thus, accurate forecasting is highly desirable to achieve reliable, secure and stable operations of electric power systems. Inaccurate forecasting results in huge economic loss to electric power companies, resulting in increased operating costs. The literature reveals that with 1% raise in forecasting error, there is the growth of 10 million operating costs every year [2].

In this modern era of technology, an accurate and effective load forecast is required for proper energy management and planning of the power system. Load forecast shows a strong relationship with weather variables on load demand such as temperature, humidity, dew point, dry bulb temperature, wind speed, rainfall. In order to achieve better forecasting results, all factors affecting the load demand can be considered as inputs for forecast models [3].

During the past few decades, several load forecasting methods have been proposed by the researchers to increase predictive accuracy. Statistical methods include linear regression [4,5], auto regressive moving average (ARIMA) [6,7], exponential technique [8] and stochastic time series [9]. These statistical methods are based on mathematical combinations of previous and current load values and most of them could not achieve the desired forecasting accuracy due to some of their own limitations. Linear regression methods basically depend on the past historical data and have a poor non-linear fitting capability, whereas ARIMA models do not consider external factors as input and generate results only on the basis of past and current data [10,11].

To overcome the limitation of statistical methods, scholars and practitioners proposed artificial intelligence-based methods. Artificial intelligence-based methods include artificial neural network (ANN) [30,31], fuzzy logic [32,33] and support vector machine (SVM) [11,22,34]. Structural risk minimization principle of SVM provides better generalization ability and solves stagnation problem. However, the main disadvantage of SVM is a high dependency between forecasting accuracy and selected SVM parameters [35]. ANN is the most common and widely accepted method for prediction problems. ANN provides least forecasting error compared to traditional forecasting models for non-linear input variables [21]. The ability to map the relationship between input and output variables without making complex dependency among the inputs is another advantage of ANN over statistical models. Neural network can easily learn complex relationships and has strong decision-making capability under uncertain conditions.

ANN with backpropagation method, based on multi-layer perceptron is one of the popular methods for solving load forecasting [29]. Due to restricted generalization ability, ANN may provide a solution which can fall in local minimum [36]. Backpropagation method has been very successful in load forecasting, its convergence rate in learning phase is very poor. Traditional ANN depends upon the initial parameters which lead to under-fitting or over-fitting problems [3].

In past few years, researchers have combined ANN with various population-based optimization learning algorithm for adjusting network structure and network parameters in order to enhance the accuracy of load forecast [37]. Neural network is good at learning when optimization algorithms such as genetic algorithm (GA) [38,39], particles swarm optimization (PSO) [40,41], fruit fly optimization [10,42], cuckoo search [43], ant colony optimization (ACO) [44], follow the leader (FTL) [29], Jaya algorithm [45], differential evolutionary (DE) [46,47] and many more are used for its training purpose. Recently, a novel optimization algorithm namely, environmental adaptation method (EAM) has been proposed to obtain global solution with higher probability [48]. EAM is different from other evolutionary algorithms in such a way that it does not use filtering methods like crossover and mutation. Rather it is based on the theory of adaptive learning. Among the population, fit individuals will survive and non-fit individuals would struggle to improve their fitness in that same environment. Singh et al. [49] successfully combined EAM with generalized neural network using wavelet transform for electricity price. However, EAM suffers from premature convergence and more processing time to deal with binary code. To overcome these shortcomings some modifications in EAM have been proposed in past few years. Mishra et al. [50] proposed an improved version of basic EAM namely improved environmental adaption method (IEAM) by embedding the role of the best particle. Particles within the population update their positions with the help of their best position in the previous and current iterations. Further, two adaptation operators (one for updating the position of the best particle and other for non-best particles) are incorporated into EAM to avoid premature convergence [51]. However, improved EAM still carries certain limitations such as, working with binary encoding and equal treatment of particles except the best ones. To remove these drawbacks of IEAM, improved environmental adaptation method with real parameter encoding (IEAM-R) is presented [48]. IEAM-R works with real data and generates high diversity along with favorable convergence growth without alteration operator. The proper mixture of exploitation and exploration in IEAM-R makes it fit for real-world applications. To further enhance its efficiency, IEAM-R can be improved by integrating with mutation operator.

Hinterding [52] implemented self-adaptive Gaussian mutation for GA to optimize numeric functions and found it superior to bit-flip mutation. Higashi and Iba [53] combined Gaussian mutation with particle swarm optimization to avoid stagnation problem. Wu [54] proposed a new forecasting model based on hybrid PSO and SVM. PSO has been combined with Gaussian and adaptive mutation to enhance the efficiency of PSO. Although several mutation operators have already been presented in past few decades, Gaussian mutation offers more flexibility and provides greater genetic diversity among particles. Gaussian mutation provides a degree of freedom by the addition of a random value from a Gaussian distribution to scale in the whole search space. By considering the effect of Gaussian mutation, we have proposed a new mutation operator namely, CGM and adapted in IEAM-R to enhance the accuracy of load forecasting.

Table 1 shows a comprehensive literature review in the area of electricity load forecasting models based on two broad categories: statistical methods and artificial intelligence-based methods. From literature, it has been observed that artificial intelligence-based methods overcome certain limitations of statistical methods and generate better results. ANN is the most widely used intelligence-based method but suffers from over-fitting problem. The findings of the literature review show that appropriate learning algorithm, pre-processed training data, and optimized network structure may increase the overall performance of neural network. From Table 1 we can deduce that learning capability of NN can be improved by population-based learning algorithms.

In this paper, we introduce a hybrid methodology to increase the accuracy of load forecasting by integrating an ANN training task with an enhanced optimization algorithm termed as IEAMCGM-R to determine ANN parameters. This enhanced optimization algorithm is composed of IEAM-R and proposed CGM to improve genetic diversity and convergence rate of the solution. To show the superiority and stability of our proposed method extensive experiments are conducted over two electricity market data sets and a statistical test is performed to ensure statistical significance of results. The major contributions of this paper can be summarized as follows:

- Proposal of a novel mutation operator, namely controlled Gaussian mutation (CGM) for enhancing the diversity of IEAM-R.
- An enhanced optimization algorithm, termed as IEAMCGM-R, composed of improved IEAM-R and CGM is proposed.
- A hybrid methodology IEAMCGM-R integrated with ANN namely (ANN-IEAMCGM-R) is presented to solve the problem of load forecasting.
- Effectiveness of the proposed hybrid methodology is shown by evaluating its performance on England electricity market, and the Australian electricity market.
- The experimental simulation and results show that the hybrid methodology generates least forecasting error compared to traditional models.
- Statistical test is performed to ensure the statistical significance of the results obtained.

The rest of this paper is organized as follows. Section 2 presents an overview of artificial neural network, Gaussian mutation, and environmental adaptation method. In section 3, the proposed algorithm is presented. Section 4 presents the proposed hybrid forecasting methodology by integrating a neural network with the proposed algorithm for electricity load forecast problem. Section 5 presents the case studies used to test and validate the proposed hybrid methodology. In section 6, the numerical results of two different data sets are presented and discussed. The last section provides the conclusion with some future remarks.

2. Background details

This section provides brief background details of artificial neural network, environmental adaptation method, and Gaussian mutation.

2.1. Artificial neural network

An artificial neural network is a mathematical tool and it is inspired by the working of biological human brain system where neurons are the basic processing units [55]. These basic units receive information at input nodes, process them internally and generate a response at the output node [30]. ANN is best known for learning capability, generalization ability and fault tolerance [32,56]. However, ANN carries certain limitations which have to be decided while implementing it.

- Selection of input variables
- Choosing training and testing data
- Selection of hidden layers
- Selection of neurons in each hidden layer
- Selection of training algorithm
- Selection of the activation function
- Choosing stopping criterion
- Selection of ANN architecture

A three layered feed-forward neural network is shown in Fig. 1. In multi-layer feed-forward (MLF) neural network, neurons are arranged into layers having one input layer, one or more hidden layers and one output layer [57–59]. Each connection within the network has a certain weight associated with it. Suppose, $W_{i,j}$ is connection weight between neurons i and j then the output value of neuron j for two consecutive layers $(k-1, k)$ having X_i input is determined by Eq. (1).

$$Y_j = \Theta_j\left(b_j + \sum_{i=1}^{m} W_{i,j} X_i\right) \quad i\varepsilon[1,m] \ \ and \ \ j\varepsilon[1,n] \tag{1}$$

where, b_j is threshold coefficient (bias).

Θ_j is activation function
m is number of neurons in layer $k-1$
n is number of neurons in layer k

2.2. Environmental adaptation method

Environmental adaptation method (EAM) is based on the theory of adaptive learning proposed by Mark Baldwin [60]. According to this theory, if an organism finds itself in a new environment where its phenotype is not appropriate then it tries to modify itself according to the new environment based on its adaptive behavior. Adaptation process thus depends on both individual and environmental conditions.

Mishra et al. [61] first introduced the concept of adaptation in the field of an evolutionary algorithm. In this proposed work, environmental conditions are taken into account to improve the fitness of species. All species within the environment try to improve their genome structures for adjusting in the new environment and those who fail to survive will die. Mutation may occur due to

Table 1
Research contributions in the field of electricity load forecasting.

Technique		Reference/Year	Highlights	Remark
Statistical methods	Linear regression	[4]/2016	Linear regression models based on patterns of daily cycles for STLF.	Linear regression forecasting models are easily implemented, but they produce higher forecast error due to their inability to map the complex input and output relationship [3].
		[5]/2010	Implementing clustering model to identify a family of functional linear models for peak load forecasting.	
	Multiple regression	[12]/2011	Proposal of novel multiple linear regression benchmark for STLF.	Multiple regression learn more about the relationships between several independent variables and a dependent variable, however, in most research they are combined with other sophisticated methods to perform well [13].
	ARIMA	[6]/2011	ARIMA model embedded with the lifting scheme for STLF.	ARIMA model fit stationary time series, but load series are usually non-stationary time series. Thus, it can forecast the linear basic part of the load, but it cannot forecast the non-linear sensitive part of the load [14].
		[14]/2012	Hybrid of ARIMA and SVMs for STLF.	
	Exponential smoothing	[15]/2012	Using exponentially weighted methods for load forecasting	Exponential smoothing is faster and efficient than ARIMA, but it generates more error for a long-term load forecast.
Artificial intelligence-based methods	Fuzzy logic	[16]/2015	Combines dynamic and fuzzy time series approach for mid term load forecast	Fuzzy techniques, solves uncertainties in the load forecast and generates better forecast results than statistical based forecast models. However, fuzzy system shows strong dependency on expert systems and cannot generate good accuracy [17].
		[18]/2012	Proposal of interval Type-2 fuzzy logic systems for load forecasting	
		[19]/2010	Fuzzy inductive reasoning and evolutionary algorithm for STLF.	
	Support vector machine	[20]/2010	Modifying the risk function of SVR by locally weighted support vector regression for load forecast	SVM is one of the most robust and accurate methods and is a highly effective model in solving non-linear problems. One of the major drawbacks of SVM is higher computational time for the constrained optimization programming [21].
		[22]/2013	STLF based on the support vector regression machines to reduce the operator interaction in the model-building procedure.	
		[23]/2010	Reduced SVM training data by ant colony optimization for power load forecasting	
	Artificial neural network	[24]/2010	Use of multilayer perceptron model (MLP) for prediction of Greek long-term energy consumption.	ANN produces higher forecast accuracy due to its capability to map the complex input and output relationship by its training process. For a large neural network, high processing time is required [21].
		[25]/2012	Levenberg-Marquardt optimization technique is used as a back propagation algorithm for ANN in STLF.	
	Population-based learning algorithm for NN training	[10]/2013	Generalized regression neural network (GRNN) is combined with fruit fly optimization algorithm to automatically select the appropriate spread parameter value for the GRNN	Population based algorithm improves the training capability by optimizing the weights of neural network to enhance the forecasting accuracy such as particle swarm optimization (PSO), genetic algorithm (GA), follow the leader (FTL), etc. These evolutionary algorithms show better performance than traditional forecasting models [3].
		[26]/2010	Bacterial foraging optimization (BFO) is used to train neural network for STLF	
		[27]/2010	Due to extensive capability of global optimization of PSO, it is used to optimize the weights of radial basis function neural network (RBFNN) to forecast load	
		[28]/2015	To improve forecasting accuracy, BPNN is combined with adaptive differential evolution algorithm for time series forecasting	
		[29]/2018	Novel evolutionary algorithm FTL is proposed to optimize weights of neural network for STLF.	

environmental noise. EAM is a population-based algorithm which starts with the random initialization of population and uses three operators: adaptation; alteration; and selection. In the forthcoming section, we will discuss various variants of EAM. Table 2 describes the terminologies used throughout the paper.

2.2.1. Basic environmental adaptation method

Basic EAM uses Eq. (2) for adaptation operator.

$$X_{i+1} = \left(\alpha * (X_i)^{c_i} + \beta\right) \Big/ 2^l \tag{2}$$

where, Current environment fitness $(c_i) = \frac{F(X_i)}{F_{avg}}$

Limitations: Although, EAM generate better results than GA and PSO but it has low convergence rate and gets trapped in local optima.

2.2.2. Improved environmental adaptation method (IEAM) version 2012 [50].

To resolve the limitations of basic EAM few modifications have been done in adaptation operator. In basic EAM, each candidate updates its phenotype structure on the basis of environmental conditions without any interference of the best candidate. In improved version of EAM best candidate has been incorporated and modified adaptation operator is shown in Eq. (3).

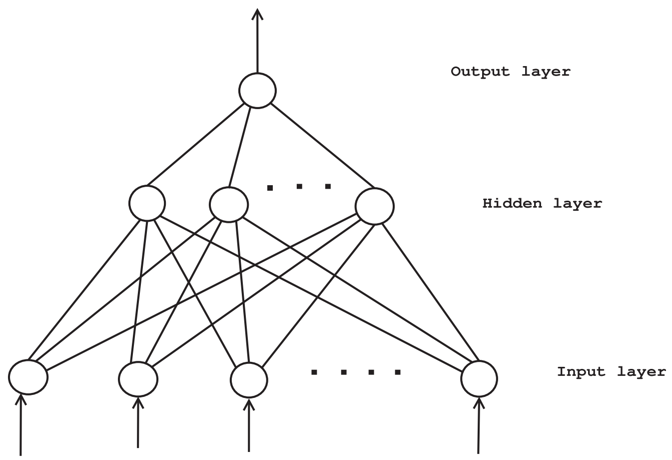

Fig. 1. Three layered feed-forward neural network.

Table 2
Terminologies.

Symbol Used	Description
α and β	Random numbers (vary according to problem)
c_i	Current environment fitness
C_1	Random candidate selected among fit candidates
C_2	Random candidate selected among struggling candidates
$F(X_i)$	Fitness value of candidate *Xi*
F_{avg}	Average fitness value of current candidates
F_{mid}	Fitness value of *mid* indexed candidate
G_i	Position of best candidate in current iteration
gen	Temporary variable
ite	Current iteration count
l	Number of bits to represent an individual candidate
L_l	Lower limit of search space range
Max_{ite}	Maximum number of iterations
mid	Index position of F_{avg}
Ps	Population size
p_m	Mutation probability
$P_{def_{mutation}}$	Predefined mutation probability
P_{i-1}	Position of best candidate in previous iteration (if exist)
rand	Random value between 0 and 1
U_l	Upper limit of search space range
X_{old}	Current candidates within population
X_{new}	Updated candidates after adaptation
X_{temp}	Temporary candidates
X_{mut}	Candidates generated after mutation
X_{i+1}	Update structure of *Xi*
Xi	Decoded decimal value of binary value of candidate *Xi*

$$X_{i+1} = \left(\alpha*(X_i)^{c_i} + \beta(P_{i-1} - X_i) + (G_i - X_i)\right) \Big/ 2^l \tag{3}$$

Limitations: The position vectors of the current best candidate and previous best candidate decide the movement of each candidate instead of the global best candidate and this method works only for binary inputs.

2.2.3. *Improved environmental adaptation method (IEAM) version 2014 [51].*

To further improve the convergence growth of both IEAM and EAM, a slight modification is done in adaptation operator of basic EAM [51]. To avoid the stagnation problem, best candidate uses Eq. (2) as adaptation operator and non-best candidates use Eq. (4).

$$X_{i+1} = \left(\alpha*(X_i)^{c_i} + \beta(G_i - X_i)\right) \Big/ 2^l \tag{4}$$

Limitations: In IEAM, only one candidate (best) within the population is used to explore the search space. Moreover, IEAM works with binary encoding and have less diversity. Also, every candidate other than the best candidate is treated equally [48].

2.2.4. *Improved environmental adaptation method with real parameter encoding (IEAM-R) [48].*

IEAM-R is an improved version of IEAM and it works with real parameter encoding that ensures easy implementation of real problems. Unlike IEAM, IEAM-R treats every candidate differently [48]. A proper combination of exploration and exploitation is required to reach the optimal solution in a minimum number of iterations. In IEAM-R, average fitness of candidates is calculated. Candidates having their fitness values above average fitness are selected for exploitation and rest candidates for exploration.

Unlike EAM, IEAM-R consists of only two operators: adaptation and selection. IEAM-R works in the following three phases:

2.2.4.1. ***Phase 1: initialization****.* Initial candidates of the population are generated randomly.

2.2.4.2. ***Phase 2: creating new population****.* Adaptation and selection operators are used to create a new population. For a maximization problem, first the average fitness (F_{avg}) of the population is computed. Based on the average fitness values, the candidates of the population are distributed into two categories such as fit candidates (fitness value greater than F_{avg}) and struggling candidates (fitness value below F_{avg}). These fit candidates are responsible to improve the convergence rate and struggling candidates brings diversity by exploring search space [48]. Two different types of adaptation operators are used in IEAM-R - uncontrolled adaptation for exploration and controlled adaptation for exploitation.

For fit candidates (exploitation), controlled adaptation is applied by moving the solutions in a specific bandwidth range. The bandwidth is a difference between the two candidates chosen randomly (one from fit candidates and other from struggling candidates). Each fit candidate is updated as follows:

$$X_{i+1} = rand + c_i*X_i \tag{5}$$

For struggling candidates (exploration), uncontrolled adaptation is applied by moving the solutions in random directions. This helps in exploring new regions and avoids premature convergence problem. Each struggling candidate is updated as follows:

$$X_{i+1} = X_i + (C_1 - C_2)*rand \tag{6}$$

Current population and temporary population (population obtained by adaptation operator) are combined and best optimal candidates of size P_s from this combination are selected by using selection operator for next generation.

2.2.4.3. ***Phase 3: optimal solution generation and termination****.* In every iteration adaptation and selection operators are used and this process will continue until it meets the termination condition.

From literature it has been observed that to maintain a balance between exploration and exploitation, mutation operator is usually considered for intelligent algorithms for providing more diversity within the population [62]. Based on the same idea, we are combining IEAM-R with a novel mutation operator proposed in this paper.

2.3. *Gaussian mutation*

Mutation makes a small random change that takes place in small parts of an individual in the population to avoid local optimal solutions [63]. Mutation concept was first introduced in GA with standard bit-flipping mutation operator [64]. This traditional mutation operator is not ideal for binary values because decoding generates very small change within a candidate. Gaussian mutation operator is very flexible and supports fine-tuning of solutions. Gaussian mutation operator provides genetic diversity by showing the suitable direction to particles. It provides stronger local exploitation capability, which leads to faster convergence rate of the algorithm [62].

Gaussian mutation adds a random value from a Gaussian distribution to each element of the parent candidate. Suppose $X \in [l, u]$ be any real variable. Gaussian mutation operator G_m changes variable X to

$$G_m(x) := \min(\max(N(X, \sigma), l), u) \quad (7)$$

where.

$\sigma$is standard deviation
N is number of element to be generated
l is lower bound of X
u is upper bound of X and

σ may depend on the length of the candidate $L = u - l$ and a number of current iteration. G_m is applied with a probability p_m to each variable.

3. Enhanced optimization algorithm (IEAMCGM-R) - IEAM-R with Controlled Gaussian Mutation

For meta-heuristic optimization algorithms, exploration and exploitation are two main concepts where exploration extends the capability of search space of the algorithm and exploitation has the capability of finding the best solution. Therefore, a balance between exploration and exploitation is required in IEAM-R algorithm. To overcome this common problem in IEAM-R, a novel mutation operator has been proposed in this paper.

IEAM-R is capable of solving the real problems of minimization or maximization functions. Although IEAM-R has a proper balance between exploration and exploitation, little more exploitation within EAM might generate better results. From the last few decades, mutation has been combined with many optimization algorithms to bring higher diversity among candidates and to increase the convergence growth within fewer iterations. Optimization algorithms such as GA, PSO, differential evolutionary (DE) algorithm and many more have already been combined with mutation and these combinations gave more reliable and efficient results [65–67]. Among available mutation operators, Gaussian mutation is superior and widely acceptable in generating high diversity.

3.1. Proposed mutation operator - Controlled Gaussian Mutation (CGM)

Gaussian mutation dominates for local exploitation but for global optimization, a proper balance between exploration and exploitation is required. To bring greater diversity in Gaussian mutation a little modification has been proposed in this paper.

In this method, the best candidate P_{best} generated in each iteration is used for the improvement in Gaussian mutation operator. Whenever mutation probability condition satisfies, P_{best} candidate is chosen to mutate in only one dimension. This one-dimensional value is randomly drawn from Gaussian distribution and is swapped by a randomly chosen element of the best candidate to generate a new child. Controlled Gaussian mutation operator *CGM* changes P_{best} candidate to

$$CGM(x) := \min(\max((P_{best}, \sigma), l), u) \quad (8)$$

where, σ is the standard deviation and l and u represents lower bound and upper bound of X. CGM increases the chance of getting a new and better P_{best} by adding a random Gaussian value. Here, the controlled environment meant for choosing only the best candidate

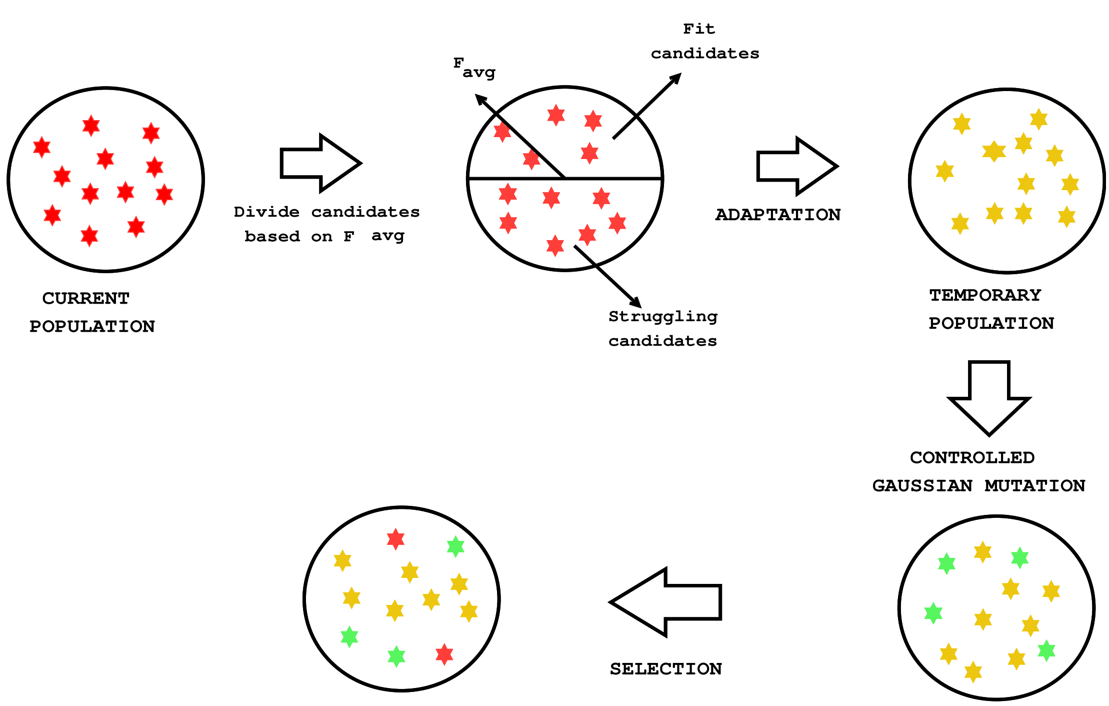

Fig. 2. IEAMCGM-R working procedure.

in every iteration for mutation. The advantage of CGM over Gaussian mutation is that CGM adds only one random noise within the best candidate while Gaussian mutation adds more than one random Gaussian distribution values to a randomly chosen candidates, hence it increases the probability of adding more noise and decreases the probability of getting a better P_{best} in forthcoming iterations.

In our proposed work, we combine IEAM-R with CGM termed as IEAMCGM-R to exploit the best candidate for generating better candidates in coming iterations. For proper visualization of our proposed work, a schematic diagram is shown in Fig. 2. This figure clearly demonstrates the working procedure of IEAMCGM-R. At the end of every iteration, X_{old} of size P_s is updated which passes through adaptation, mutation, and selection i.e. it evolves until a termination condition is reached. Final solution generated after termination condition is an optimal solution to the problem. The pseudo-code of IEAMCGM-R is given in Algorithm-1.

Algorithm 1
IEAMCGM-R

Input: Input neurons, Number of hidden layer, Hidden neurons, Output neurons, L_l, U_l
Output: Best solution of optimization problem
Begin
(1) Initialize population size, Ps:
(2) Randomly initialize the population X_{old} of size Ps within the search space of L_l and U_l
(3) Compute fitness value of each candidate within X_{old}.
(4) Set iteration count $ite = 2$.
(5) **while** terminate condition is not satisfied i.e. $ite < Max_{ite}$
(6) Compute average fitness F_{avg} value of X_{old}.
(7) Sort X_{old} according to their fitness value and name it X_{temp} and find F_{mid}, which divides X_{old} in two sets: fit candidates having fitness $> F_{avg}$ and struggling candidates with fitness $< F_{avg}$.
(8) Select two random candidates C_1 and C_2 one each from fit candidates and struggling candidates.
(9) **for** fit candidates within X_{temp} having index $i < mid$
$$X_{new,i} = X_{temp,i} + (C_1 - C_2)*rand$$
(10) **end for**
(11) **for** struggling candidates within X_{temp} having index $i > mid$ up to Ps
$$X_{new,i} = X_{temp,i} + rand*c_i$$
(12) **end for**
(13) Set $gen = 1$ and randomly generate mutation probability p_m
(14) **While** $gen < Ps$, check ***If*** $p_m < P_{def_{mutation}}$
(15) Apply ***Controlled Gaussian mutation*** to find X_{mut}
(16) **end while**(17) Check the boundary conditions of X_{mut}
(18) Compute fitness value of X_{mut}.
(19) Concatenate old X_{old} ,X_{mut} and current X_{new} population.
(20) Update C_{old} by selecting best Ps candidates according to their fitness value from merged population.
(21) **end while**
End

4. Proposed hybrid methodology for electricity load forecast

In this section, we discuss the process involved in the implementation of our proposed hybrid methodology for accurate load forecasting.

4.1. Problem statement

The process of estimating or predicting future load demand based on previous information is load forecasting. Load forecasting is a typical task, that requires high accuracy to fulfill the load demand, therefore, it is highly recommended to develop fast and precise forecasters. Although several techniques have already been introduced for electricity load forecast and ANN is one of them to resolve this issue but it does not perform well due to the overfitting problem. To overcome this problem of ANN and to achieve greater accuracy in the field of electricity forecasting, the proposed algorithm (IEAMCGM-R) is combined with ANN named ANN-IEAMCGM-R. Thus it enhances the learning ability of neural network. The process involved in our proposed work is shown in Fig. 3 which demonstrates the major steps that have to be followed during the load forecast.

4.2. Objective function

Load forecasting is a single objective problem with real parameters and without constraint. The goal is to minimize the predicted error generated through NN architecture by adjusting weights in successive generations. In our work, mean absolute error (MAE) is considered as an objective function and its mathematical equation is shown in Table 3.

Fig. 4 shows the flowchart of our proposed hybrid methodology (ANN-IEAMCGM-R), starting from load series and ending with a day-ahead load forecast. More detailed working of our proposed hybrid methodology is presented in section 4.7. Before discussing the detailed working of ANN-IEAMCGM-R, we present the following steps required for our work.

4.3. Input selection and features extraction

We collect data for training and testing purposes that enable good forecast and are computationally inexpensive in terms of size to stay away from the risk of ANN over-training. Preferably, we choose easily available data to show its integrity and standardization. We selected explanatory variables that show high correlation with the load like temperature, rainfall, wind speed, holiday, etc.

In case study I, to forecast a day ahead load on the hourly interval of $Y_{t+1}.......Y_{t+24}$ for England market following input variables have been considered: $L1$ - previous 24 h average load ($Y_{t-1} + Y_{t-2} + ... + Y_{t-24}/24$), $L2$ - 24-hr lagged load, and $L3$ - 168-hr (previous week) lagged load, T_1 - dry bulb temperature ($^\circ C$), T_2 - dew point temperature in $^\circ C$, $H_1, H_2, ...H_{24}$ - hour of day (H_1 represents 00:00 h load and H_{24} represents 23:00 h load), D_i - day of the week ($i = 1, 2, ...7, D_1$ is Sunday,.... and D_7 is Saturday) and H - holiday/weekend (H is 0 for holiday/weekend otherwise 1).

In case study II, to forecast a day ahead load on half hourly interval of $Y_{t+1}.......Y_{t+48}$, the load data incorporated in training data set consists of: $L1$ - previous 24 h average load ($Y_{t-1} + Y_{t-2} + ... + Y_{t-48}/24$), $L2$ - 24-hr lagged load, $L3$ - 336-hr (previous week) lagged load, $H_1, H_2, ...H_{48}$ - half hour of day (H_1 represents 00:00 h load and H_{48} represents 23:30 h load) and D_i - day of the week ($i = 1, 2, ...7, D_1$ is Sunday,.... and D_7 is Saturday). To build our proposed forecasting model, the input selection $I = I_1, I_2, ... I_n$ is formed where n is a number of input parameters chosen.

4.4. Data pre-processing

Before forecasting, missing and extra values are managed and data pre-processing is done to eliminate spikes and outliers. Data normalization (min-max normalization) is done to train the neural network in the interval of $[-1, 1]$. Data normalization is important as input variables are of different ranges and also it reduces the training time. To perform various experiments, we divide the data into two categories: training data and testing data. Scaling of test data is done by the same factor with which train data is normalized. The ANN output is scaled back with the same factor as target values of training data.

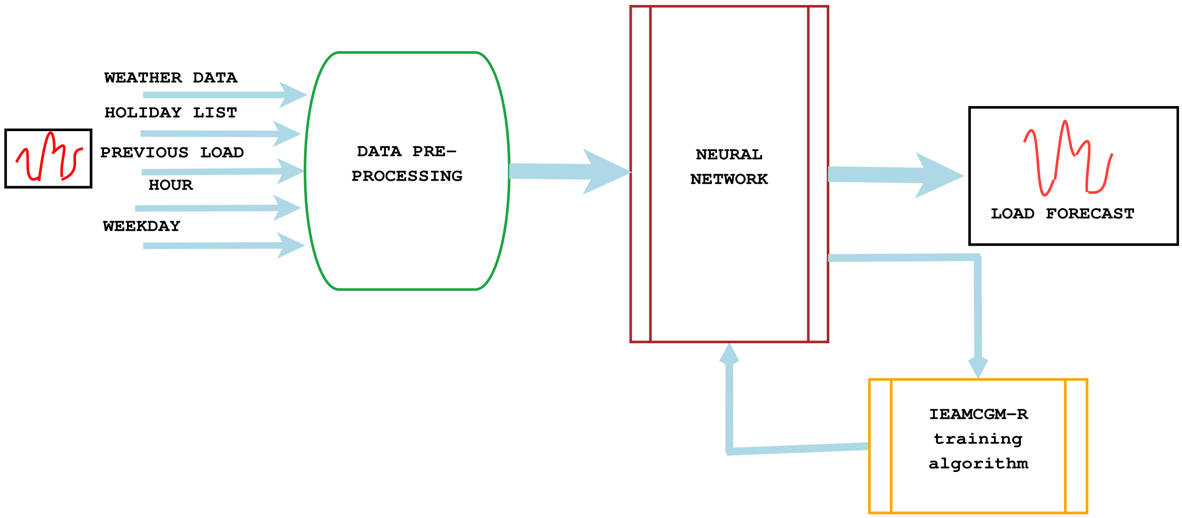

Fig. 3. ANN-IEAMCGM-R electricity load forecast process.

Table 3
Evaluation metrics.

S.No.	Metric	Error metrics equation
1	Mean absolute error	$MAE = \frac{1}{N}\sum_{j=1}^{N}\left\|y_j - y'_j\right\|$
2	Mean absolute percentage error	$MAPE = \frac{1}{N}\sum_{j=1}^{N}\frac{\left\|y_j - y'_j\right\|}{y_j}*100$
3	Daily peak mean absolute percent error	$Daily\ Peak\ MAPE = \frac{1}{N}*\frac{\|max(TL_m) - max(FL_m)\|}{max(TL_m)}*100$
4	Pearson's correlation coefficient	$r = \frac{\sum_{j=1}^{N}(y_j - \overline{y})(y'_j - \overline{y'})}{\sqrt{\sum_{j=1}^{N}(y_j - \overline{y})^2\sum_{j=1}^{N}(y'_j - \overline{y'})^2}}$
5	Directional change	$DC = \frac{100}{N-1}\sum_{j=1}^{N-1}a_t\ ,\ a_t = \left\{\begin{matrix} 0, otherwise \\ 1, if(y'_{j+1} - y_j)(y_{j+1} - y_j) > 0\end{matrix}\right\}$
6	Index of agreement	$IA = 1 - \frac{\sum_{j=1}^{N}(\overline{y}_j - y_j)^2}{\sum_{j=1}^{N}(y'_j - \overline{y})(y_j - \overline{y})^2}$

Note: y_j = actual value of day j, y'_j = predicted value of day j, $\overline{y}$ = mean of actual value, $\overline{y'}$ = mean of Predicted value, FL_m = forecast load value for every 24 h, TL_m = target load for every 24 h, N = number of elements in training data.

4.5. ANN training

While using ANN as a training tool, it is important to extract its existing dependencies and relations from the train data, without losing its generalization ability to stay out of the over-fitting problem. To deal with these issues, a supervised learning process i.e. ANN is coupled with IEAMCGM-R to find optimal network weights.

4.6. ANN architecture tuning

The set of parameters involved in network training includes weight adjustments to generate an optimal solution and we updated these weights by our proposed algorithm.

4.7. Procedure

During network training, the error is minimized, weights are adjusted and optimal weights are generated for load prediction. The details of the ANN-IEAMCGM-R model are as follows:

- **Data splitting:** Load data is divided into training data and testing data and normalized.
- **Initialization:** Population size (P_s) and termination condition (Max_{ite}) are set to 100 and 10,000 respectively. Number of input nodes (m) is same as a number of input parameters and number of hidden neurons (n) is set to 10.
- **Evolution starting:** We initialized current generation as $ite = 0$ and network weights $W_{old} = W_1, W_2, W_3.......W_D$ randomly within the range of [-1,1] of size P_s. Here, D represents dimension of a candidate, $D = m*n + 2*n + o$.
- **Preliminary calculations:** Training input is given to ANN model for load forecasting and according to the results obtained, fitness value (MAE) of each candidate within P_s is calculated. W_{old} is sorted on the basis of their MAE and average fitness F_{avg} is computed. Sorted population is divided into two halves: H_1 and H_2. H_1 has fit candidates (fitness values greater than F_{avg}) and H_2 contains struggling candidates (fitness values below F_{avg}).
- **Offspring generation:** Fit candidate H_1 and struggling candidate H_2 are updated according to Eq. (5) and Eq. (6) respectively. Updated H_1 and H_2 are combined to form W_{temp}. CGM is applied to a candidate of W_{temp} whenever mutation condition is satisfied to get W_{mut}. Fitness value ofW_{mut} is calculated. W_{old}, W_{temp} and

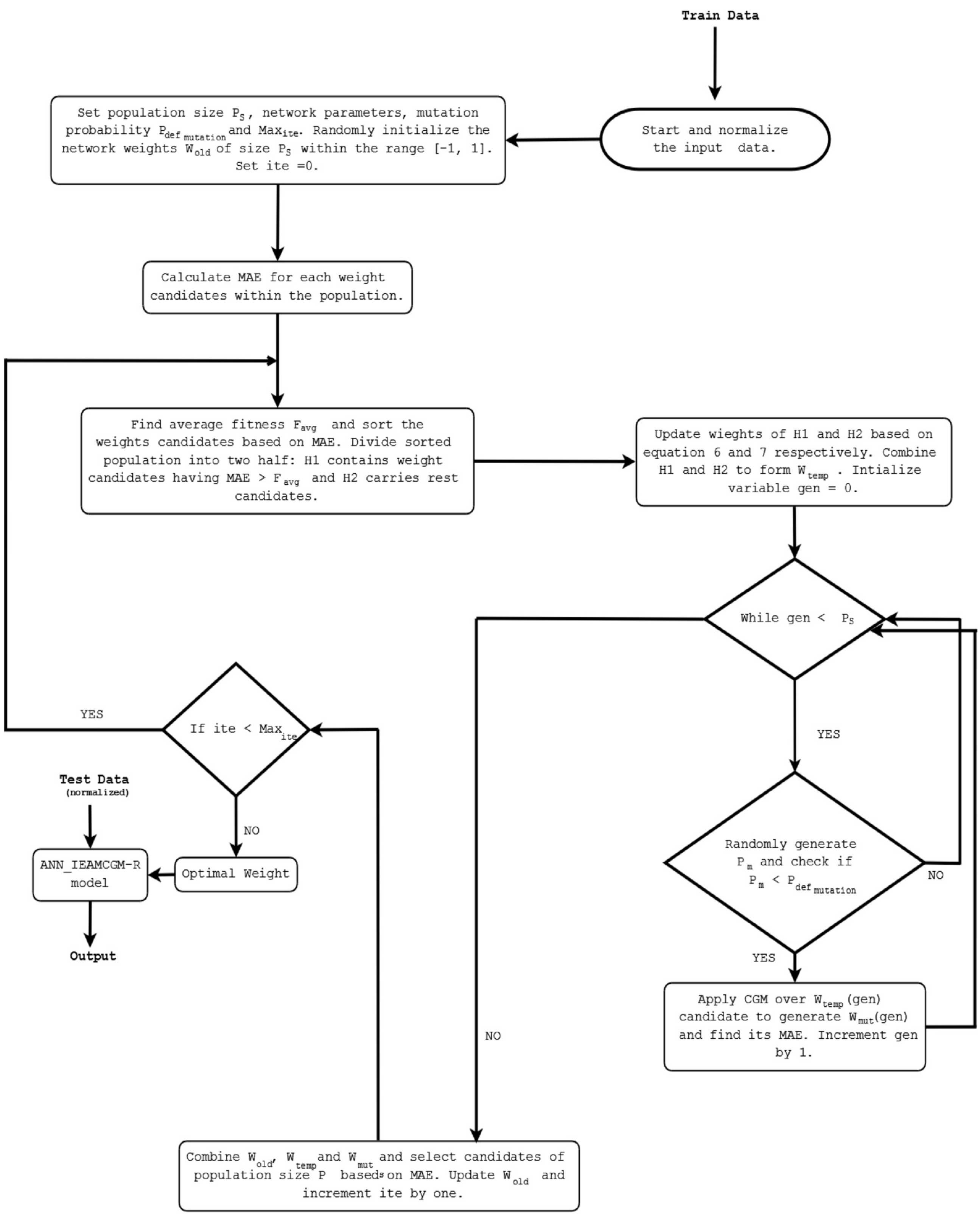

Fig. 4. The flowchart of the ANN-IEAMCGM-R model.

W_{mut} are combined and sorted according to their MAE. Best candidate of size P_s is selected on the basis of MAE to update W_{old}. *ite* is incremented by one.

- **Circulation stops:** When current iteration count, *ite* reaches Max_{ite} (stop criterion satisfied) then best weight vector is selected from the current generation otherwise we will continue the process from initialization step.
- **Test and evaluation:** Obtained best weight candidate is used to test the model and generate the output. Output load generated is denormalized to achieve its actual load value.

5. Experiments

In this section, we discuss the error metrics which verify the effectiveness and performance of our proposed hybrid methodology along with the data description of two electricity markets used in our case studies.

5.1. Evaluation metrics

Since past few decades, several evaluation metrics have been developed to evaluate the performance of load forecast. However, there are no predefined rules for selecting these metrics. In this paper, we use the six commonly used evaluation metrics shown in Table 3. Mean absolute error (MAE) indicates the overall level of errors and is powerful enough to evaluate total absolute forecasting error [68]. Mean absolute percentage error (MAPE) expresses accuracy as a percentage and is most frequently used performance metric of forecast error. Directional Change (DC) exhibits the movement directions or turning points of prediction. Pearson's correlation coefficient (r) shows the correlation between the predicted value and the observed value. Lastly, the index of agreement (IA) is a standard metric of the degree of model prediction error [69].

Model implementation is performed on MATLAB R2015a software on a Windows 8.1, 64-bit machine with Intel(R) Core(TM) i5 CPU 760 @ 2.80 GHz with 1 processor. Here we set population size as 100, maximum number of iteration as 10,000 and number of hidden neurons as 10, the number of input neurons is same as the number of input parameters taken in the case study and one output neuron.

In order to validate the performance of ANN-IEAMCGM-R, we have compared our approach to following forecasting approaches.

- Linear regression (LR)
- Generalized regression neural network (GRNN)
- ANN with Broyden Fletcher Goldfarb Shanno Backpropagation (ANN-BFGS)
- ANN-Jaya (Neural network with Jaya algorithm)
- ANN-IEAM-R (Neural network with Improved environmental adaptation method with real parameter encoding algorithm)
- ANN-IEAMGM-R (Neural network with IEAM-R with Gaussian mutation)

5.2. Statistical test

Furthermore, to verify the forecasting accuracy improvement of our proposed hybrid methodology, a statistical test is performed. The forecasting results of ANN-IEAMCGM-R, ANN-IEAMGM-R, ANN-IEAM-R, ANN-Jaya, ANN-BFGS, LR and GRNN are conducted by the Friedman statistical test. The Friedman test is a non-parametric, multiple comparisons test that aims to detect significant differences between the results of two or more forecasting models [70,71]. The statistic F of Friedman test is shown as follows:

$$F = \frac{12N}{q(q+1)}\left[\sum_{j=1}^{q} Rank_j^2 - \frac{q(q+1)^2}{4}\right] \tag{9}$$

where N is the total number of forecasting results; q is the number of forecasting models compared; $Rank_j$ is the average rank sum received from each forecasting value for each model. The null hypothesis for the Friedman test is expressed as - *all forecasting models generate the same forecasting errors and the alternative hypothesis is the negation of the null hypothesis.*

5.3. Case study I (ISO data set)

To evaluate the performance of ANN-IEAMCGM-R, we use this hybrid model for predicting ISO New England electricity. The models are trained on hourly data from the NEPOOL region from 2004 to 2007 (training samples-35064) and tested on out-of-sample data from 2008 to 2009 (testing samples-17544). The independent variables given as input to a neural network are dry bulb temperature, dew point temperature, hour of day, day of the week, the holiday/weekend indicator (0 or 1), previous 24-hr average load, 24-hr lagged load, 168-hr (previous week) lagged load. In this experiment, the ANN approach uses these eight parameters as input data vectors and generates target output corresponding to the load evaluated at the time given i.e., the load generated at every hour.

Although, there are several factors that affects the load forecast, we have chosen these parameters on the basis of their availability. Table 4 shows that the chosen input variables are correlated with the load and are relevant factors which are affecting the load forecast of NEPOOL.

5.4. Case study II (NSW data set)

For showing the effectiveness of our algorithm we have also used NSW electricity data set for load forecast. For NSW region, 2 years electricity data set from Australian Energy Market Operator (AEMO) is divided on the basis of seasons namely autumn (March–May), winter (June–August), spring (September–November), and summer (December–February). For each season, 4 months data is taken as the train data set, and the remaining 2 months as the test data set. Previous load (24-hr average load, 24-hr lagged load, 168-hr lagged load), an hour of day and day of the week are input parameters to the neural network. The correlation between load and these input parameters is shown in Table 5.

6. Results and discussions

Here, we discuss the results obtained from our proposed hybrid methodology with other state-of-the-art models in each case study.

Table 4
Correlation coefficients between the input parameters and load for years 2004–2007.

S. No.	Input parameter	Correlation
1	Dry Bulb Temperature	0.19
2	Dew Point Temperature	0.07
3	Hour of the Day	0.51
4	Day of the Week	0.03
5	Holiday/Weekend indicator (0 or 1)	0.26
6	168-hr (previous week) Lagged Load	0.85
7	24-hr Lagged Load	0.90
8	Previous 24-hr Average Load	0.56

Table 5
Correlation coefficients between the input parameters and load.

S. No.	Input parameter	Correlation
1	Hour of the Day	0.4638
2	Day of the Week	0.0342
3	168-hr (previous week) Lagged Load	0.6983
4	24-hr Lagged Load	0.8348
5	Previous 24-hr Average Load	0.5496

6.1. Results for case study I

Table 6 shows the goodness of fit (statistical model) between actual load and forecasted load obtained after excluding an input variable from load data for 100 and 500 iterations. The goodness of fit is based on the population discrepancy where normalized root mean square error (NRMSE) as a cost function is used. From Table 6 it can be clearly visualized that ANN-IEAMCGM-R for Parameter (all input variables considered) generated higher goodness of fit compared to ANN-IEAMGM-R, ANN-IEAM-R, ANN-Jaya and ANN-BFGS. With increasing iterations, the goodness of fit obtained by Parameter improves. Results shown in Table 6 describe that inclusion of all relevant input information (Parameter) generates better forecasting results than leaving behind any of them. We can conclude that better network training and better forecasting results can be achieved based on the selection of more relevant input variables. Therefore, in both case studies, we have included all the parameters taken into consideration and initialized 10,000 as maximum iterations for network training over 100 populations.

Fig. 5 shows the improvement of IEAMCGM-R over IEAMGM-R and IEAM-R for 5000 iterations by including all input variables. Improved network learning and better convergence rate of IEAMCGM-R over other two evolutionary models verify that our developed model has a higher convergence rate than the other two.

The results obtained by ANN-IEAMCGM-R, ANN-IEAMGM-R, ANN-IEAM-R, ANN-Jaya, ANN-BFGS, LR and GRNN for the years

Table 6
Performance comparison of goodness of fit of all forecasting models at 100 and 500 iteration by including all input parameters (Parameter) and by excluding dry bulb temperature (Parameter1), dew point temperature (Parameter2), hour of the Day (Parameter3), day of the Week (Parameter4), holiday/weekend indicator (Parameter5), 168-hr lagged load (Parameter6), 24-hr lagged load (Parameter7) and previous 24-hr avg load (Parameter8) for England load forecast.

Algorithm	*Ite*	Parameter	Parameter1	Parameter2	Parameter3	Parameter4	Parameter5	Parameter6	Parameter7	Parameter8
ANN-IEAMCGM-R	100	0.708,955	0.696,592	0.690,377	0.701,916	0.690,278	0.698,889	0.699,565	0.702,393	0.693,306
	500	0.718,814	0.700,694	0.707,908	0.706,061	0.712,307	0.704,324	0.703,058	0.710,605	0.706,252
ANN-IEAMGM-R	100	0.692,844	0.678,933	0.688,304	0.689,869	0.68,417	0.678,527	0.691,339	0.679,902	0.688,652
	500	0.704,673	0.698,943	0.697,693	0.693,386	0.69,666	0.694,223	0.69,419	0.699,318	0.697,693
ANN-IEAM-R	100	0.672,225	0.571,451	0.642,658	0.649,489	0.653,318	0.667,667	0.599,017	0.6543	0.643,594
	500	0.699,635	0.642,316	0.651,519	0.652,576	0.69,238	0.689,429	0.632,698	0.685,556	0.659,413
ANN-Jaya	100	0.568,558	0.52,437	0.482,749	0.559,286	0.537,019	0.480,333	0.479,143	0.44,469	0.491,059
	500	0.61,021	0.588,476	0.590,416	0.586,906	0.586,444	0.601,494	0.542,955	0.523,528	0.604,621
ANN-BFGS	100	0.621,376	0.545,633	0.611,927	0.608,595	0.600,935	0.601,714	0.557,533	0.569,284	0.562,002
	500	0.677,789	0.662,714	0.650,742	0.672,397	0.676,839	0.652,397	0.650,777	0.674,916	0.676,206

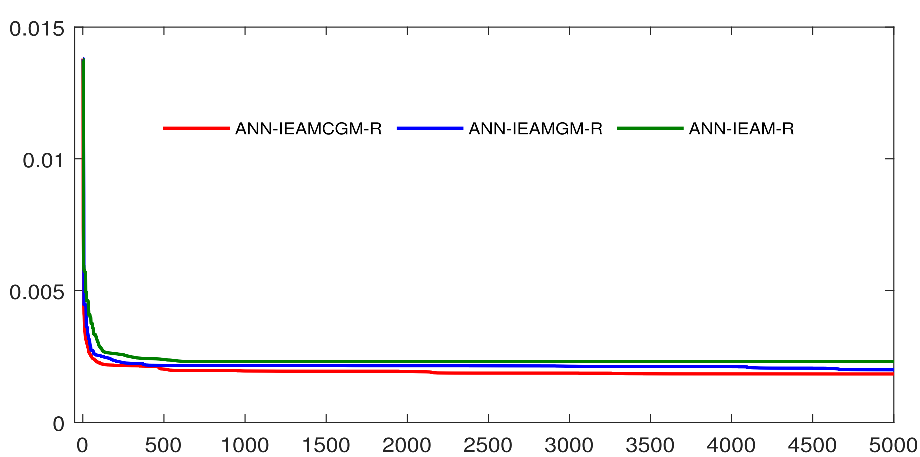

Fig. 5. Comparison of MAE generated by ANN-IEAMCGM-R, ANN-IEAMGM-R and ANN-IEAM-R during training.

Table 7
Forecasting results of all models for ISO New England electricity market for years 2008 and 2009.

Algorithm	MAE	MAPE	Daily Peak MAPE	r	DC	IA
ANN-IEAMCGM-R	***527.0614***	***3.586,657***	***3.413,847***	***0.968,189***	***74.115***	***0.983,460***
ANN-IEAMGM-R	549.566	3.724,259	3.78,645	0.964,945	74.2974	0.981,882
ANN-IEAM-R	604.5601	4.064,793	3.857,194	0.956,768	72.45,625	0.977,716
ANN-Jaya	1073.954	7.42,015	5.92,617	0.872,385	65.81,542	0.926,461
ANN-BFGS	680.4431	4.623,383	4.281,381	0.946,891	70.78,607	0.972,462
LR	645.5463	4.347,394	3.979,897	0.950,899	69.86,262	0.974,388
GRNN	840.4085	5.614,846	4.497,605	0.911,327	67.61,671	0.953,862

Note:Bold - italic values indicate the best values of error/performance metrics.

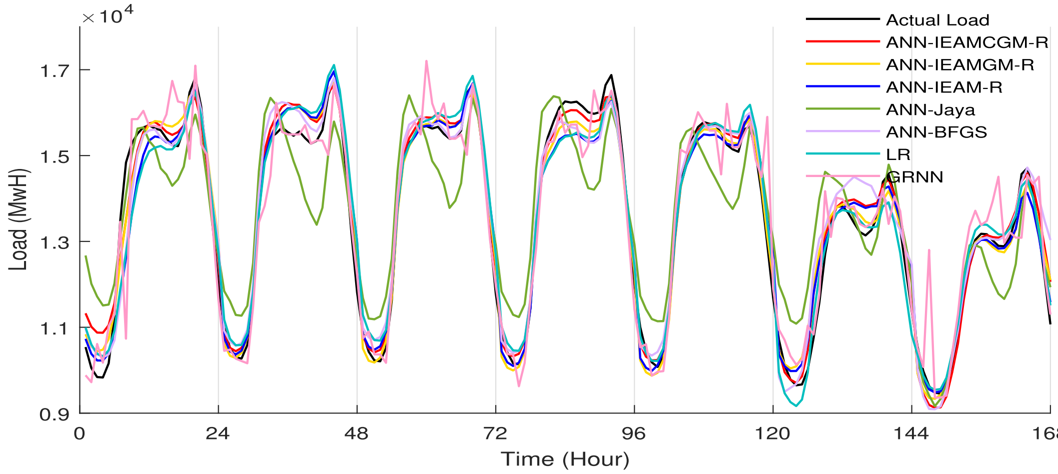

Fig. 6. Load forecast of ISO New England electricity market generated on an hourly basis (a) comparing actual (black solid line) and forecasted load (red solid line: ANN-IEAMCGM-R) with ANN-IEAMGM-R, ANN-IEAM-R, ANN-Jaya, ANN-BFGS, LR and GRNN. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

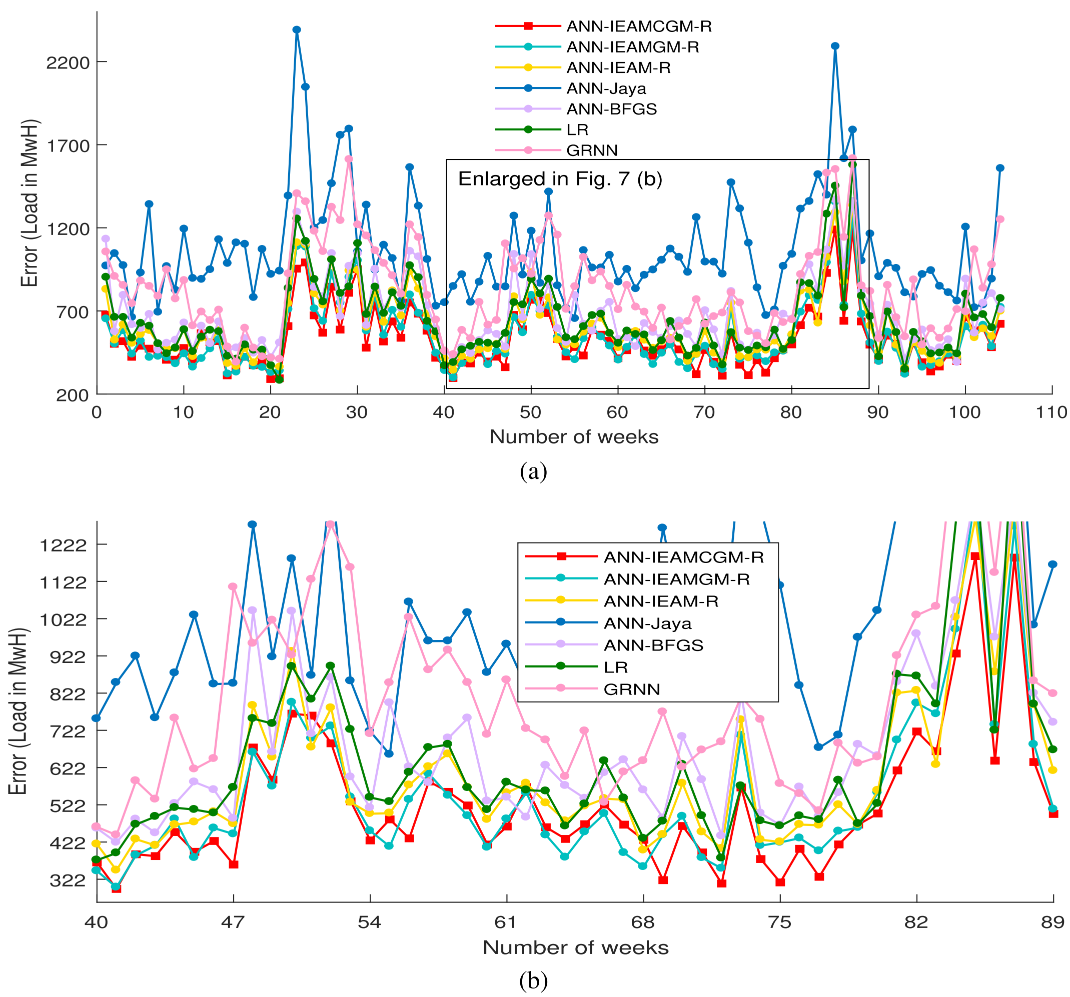

Fig. 7. (a) Comparing weekly average error for years 2008 and 2009 of ISO New England electricity market for all forecasting models (red solid line: ANN-IEAMCGM-R). (b) The enlargement comparison of weekly load of Fig. 7(a) from the compared models. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

2008 and 2009 are shown in Table 7. This error metric criterion reveals that the proposed hybrid methodology generates a better result than others. The result shows that ANN-IEAMCGM-R produced least MAPE (3.586%), MAE (527.0614 Mwh), DC (74.115) and IA (0.98,346). Among all comparing algorithms, ANN-Jaya generates maximum performance error. Fig. 6 shows the

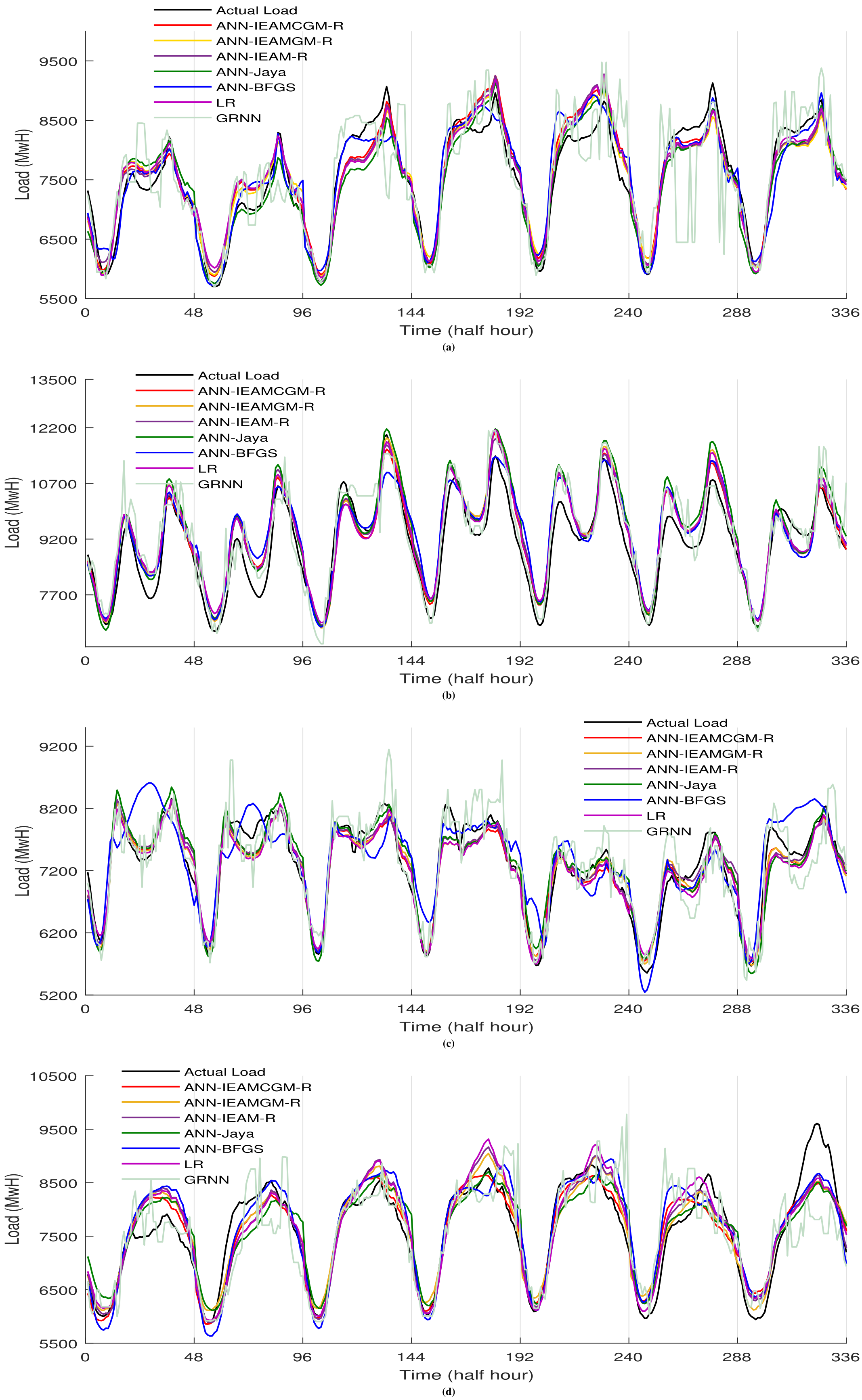

Fig. 8. Load forecast of NSW electricity market generated on half hourly basis ((a), (b), (c), (d)) comparing actual (black solid line) and forecasted load (red solid line: ANN-IEAMCGM-R) with ANN-IEAMGM-R, ANN-IEAM-R, ANN-Jaya, ANN-BFGS, LR and GRNN of four seasons namely; autumn, winter, spring and summer for a selected week. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

Table 8
Result of Friedman test for case study I [70,71].

Compared Models W = 17,544	Friedman Test $\alpha = 0.05$
ANN-IEAMCGM-R vs. ANN-IEAMGM-R	$H_0 : e_1 = e_2 = e_3 = e_4 = e_5 = e_6 = e_7$
ANN-IEAMCGM-R vs. ANN-IEAM-R	
ANN-IEAMCGM-R vs. ANN-Jaya	F = 9331.167
ANN-IEAMCGM-R vs. ANN-BFGS	$P = 0.000$ (Reject H_0)
ANN-IEAMCGM-R vs. LR	
ANN-IEAMCGM-R vs. GRNN	

Table 9
Forecasting results of all models for NSW, Australia electricity market (AEMO) for autumn season.

Algorithm	MAE	MAPE	Daily Peak MAPE	r	DC	IA
ANN-IEAMCGM-R	***251.2368***	***3.180,717***	4.121,703	***0.939,976***	***71.882,473***	***0.9,627,117***
ANN-IEAMGM-R	263.4911	3.330,922	4.32,736	0.93,949	70.17,424	0.958,847
ANN-IEAM-R	267.4049	3.388,427	4.275,341	0.936,385	69.28,596	0.957,917
ANN-Jaya	291.54,377	3.695,573	4.771,710	0.931,670	69.422,617	0.952,780
ANN-BFGS	283.7115	4.067,188	***3.628,918***	0.916,357	69.046,805	0.952,197
LR	263.5268	3.348,635	4.094,978	0.937,936	69.86,676	0.958,415
GRNN	361.051	4.553,385	4.968,636	0.860,899	69.86,676	0.922,552

Note:Bold - italic values indicate the best values of error/performance metrics.

Table 10
Forecasting results of all models for NSW, Australia electricity market (AEMO) for winter season.

Algorithm	MAE	MAPE	Daily Peak MAPE	r	DC	IA
ANN-IEAMCGM-R	***335.7683***	***3.893,514***	4.044,175	***0.928,077***	66.074,478	***0.961,833***
ANN-IEAMGM-R	341.0382	3.945,375	4.320,714	0.9,258,001	65.596,173	0.960,592
ANN-IEAM-R	349.1137	4.040,854	4.379,562	0.922,405	65.52,784	0.958,819
ANN-Jaya	379.22,741	4.341,623	5.090,108	0.913,113	65.903,655	0.954,464
ANN-BFGS	345.5719	4.011,711	4.066,895	0.921,973	66.10,864	0.9584
LR	349.0812	4.04447	4.356,431	0.922,664	65.73,283	0.958,849
GRNN	461.9426	5.368,509	5.73,951	0.864,557	64.91,288	0.927,998

Note:Bold - italic values indicate the best values of error/performance metrics.

Table 11
Forecasting results of all models for NSW, Australia electricity market (AEMO) for spring season.

Algorithm	MAE	MAPE	Daily Peak MAPE	r	DC	IA
ANN-IEAMCGM-R	***276.4673***	***3.55,338***	4.301,902	***0.903,480***	61.42,705	***0.945,440***
ANN-IEAMGM-R	282.6346	3.643,058	4.380,081	0.901,388	60.15,542	0.944,075
ANN-IEAM-R	283.2444	3.650,604	4.344,611	0.898,063	61.60,367	0.943,386
ANN-jaya	311.5199	4.073,987	4.088,852	0.88,533	57.71,812	0.939,804
ANN-BFGS	387.3879	5.026,833	5.079,942	0.844,172	58.98,976	0.911,644
LR	291.2717	3.738,498	4.424,185	0.893,694	60.26,139	0.938,179
GRNN	359.9562	4.653,698	6.040,867	0.847,067	60.5793	0.917,666

Note:Bold - italic values indicate the best values of error/performance metrics.

Table 12
Forecasting results of all models for NSW, Australia electricity market (AEMO) for summer season.

Algorithm	MAE	MAPE	Daily Peak MAPE	r	DC	IA
ANN-IEAMCGM-R	***653.3145***	***7.282,034***	8.80,283	***0.855,984***	54.6,173,198	***0.919,002***
ANN-IEAMGM-R	698.9133	7.761,013	9.003034	0.837,184	49.402,118	0.908,934
ANN-IEAM-R	712.5485	7.890,375	9.417,321	0.831,918	51.09594	0.899,632
ANN-Jaya	780.8032	8.78,944	9.65,132	0.810,218	45.31,081	0.887,409
ANN-BFGS	739.4184	8.331,234	***7.866,904***	0.803,915	52.13,798	0.887,367
LR	756.2111	8.454,861	10.00321	0.812,654	50.19,763	0.894,325
GRNN	746.695	8.258,361	10.44,684	0.802,609	56.77,327	0.891,396

Note:Bold - italic values indicate the best values of error/performance metrics.

forecasting results of all forecasting models along with ANN-IEAMCGM-R for a randomly selected week. Also, Fig. 8 shows a weekly average error analysis of all forecasting models for the years 2008 and 2009.

The results shown in Table 8 state that the null hypothesis of Friedman test fails for $\alpha = 0.05$ and it verifies that the forecasted

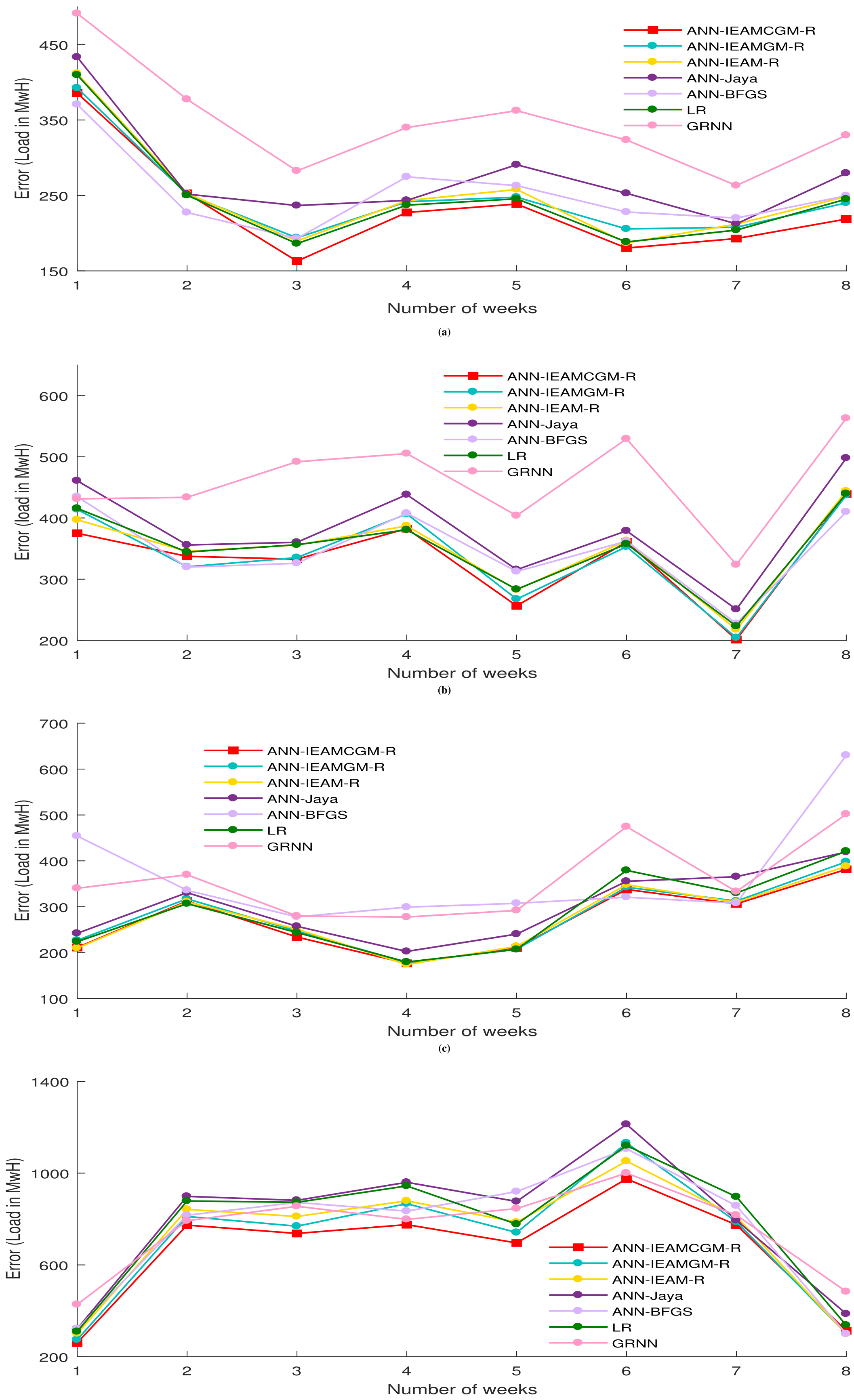

Fig. 9. Weekly average error of NSW electricity market ((a), (b), (c), (d)) comparing forecasted load (red solid line: ANN-IEAMCGM-R) with ANN-IEAMGM-R, ANN-IEAM-R, ANN-Jaya, ANN-BFGS, LR and GRNN of four seasons namely; autumn, winter, spring and summer. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

load of all models are different.

6.2. Results for case study II

The performance of ANN-IEAMCGM-R is compared with six state-of-the-art algorithms mentioned in the earlier subsection. The proposed methodology achieves superior results and significantly outperforms the ANN-IEAM-R and other forecasting methods in all four seasons. The results obtained from all state-of-the-art forecasting models for each season data set, namely autumn, winter, spring, and summer are shown in Table 9, Table 10, Table 11 and Table 12 respectively. ANN-IEAMCGM-R outperforms for almost all performance metrics. Fig. 8 shows the forecasting results of a randomly chosen week for all forecasting models along with ANN-IEAMCGM-R in all four seasons. Fig. 9 shows the weekly average error analysis of all the four seasons of NSW load data set. Table 13 shows the results obtained from the Friedman test for $\alpha = 0.05$. Result describes that all the forecast models generate unique load.

6.3. Findings from case study I and case study II

- Among four evolutionary based hybrid models (ANN-IEAM-R, ANN-Jaya, ANN-IEAMGM-R, and ANN-IEAMCGM-R), ANN-IEAMCGM-R achieves higher forecasting accuracy in both electricity markets. Mutation delivers high diversity leading to a better optimal solution than IEAM-R.
- Our proposed hybrid methodology is not dependent on seasonal changes as it generates superior results in all the cases. e.g. MAE (251.2368 Mwh) and MAPE (3.1807%) in autumn, MAE (335.7683 Mwh) and MAPE (3.8935%) in winter, MAE (276.4673 Mwh) and MAPE (3.55,338%) in spring; and MAE (653.3145 Mwh) and MAPE (7.2820%) in summer. In other words, the proposed hybrid methodology outperforms the other models by generating a minimal forecast error by maintaining its stability in different seasons.
- All effective error metrics reveal that ANN-IEAMCGM-R hybrid methodology is superior not only to evolutionary hybrid models, but also with other state-of-the-art forecasting models. GRNN and ANN-Jaya model generate the maximum error within all comparable models in all seasons showing inefficient of the load forecast.
- Results also show that ANN-IEAMCGM-R performed better than ANN-IEAMGM-R which verifies that CGM brings better diversity than GM and adding a random Gaussian value to P_{best} generates superior results than adding more than one Gaussian distribution values to randomly chosen candidates. Numerical results show improvement in the accuracy of forecasting by 11.76% and 3.69% MAPE over ANN-IEAM-R and ANN-IEAMGM-R for ISO electricity market. Moreover, for NSW electricity market, improvement in MAPE of 6.13%, 3.64%, 2.66% and 7.7% for autumn, winter, spring and summer seasons by ANN-IEAMCGM-R over ANN-IEAM-R have been seen in the results.

7. Conclusions and future work

Accurate and reliable load forecasting plays a very important role in the management of the power market. Due to the high fluctuation in load curves, it is strongly recommended to use effective forecasting models. Although, ANN itself is sufficient to deal with complex problems and is reliable but has restricted generalization ability. To overcome this problem a hybrid approach is implemented in this paper where an improved short-term load forecasting model is introduced to enhance the prediction accuracy by combining our proposed controlled Gaussian mutation with IEAM-R termed as IEAMCGM-R and then by integrating it with ANN. Our proposed controlled Gaussian mutation increases exploration capability of the algorithm and also restricts to stay away from getting stuck in local optima.

Table 13
Result of Friedman test for case study II [70,71].

Compared Models	Friedman Test $\alpha = 0.05$
Autumn (W = 2928)	
ANN-IEAMCGM-R vs. ANN-IEAMGM-R	$H_0: e_1 = e_2 = e_3 = e_4 = e_5 = e_6 = e_7$
ANN-IEAMCGM-R vs. ANN-IEAM-R	
ANN-IEAMCGM-R vs. ANN-Jaya	F = 333.016
ANN-IEAMCGM-R vs. ANN-BFGS	$P = 0.000$ (Reject H_0)
ANN-IEAMCGM-R vs. LR	
ANN-IEAMCGM-R vs. GRNN	
Winter (W = 2928)	
ANN-IEAMCGM-R vs. ANN-IEAMGM-R	$H_0: e_1 = e_2 = e_3 = e_4 = e_5 = e_6 = e_7$
ANN-IEAMCGM-R vs. ANN-IEAM-R	
ANN-IEAMCGM-R vs. ANN-Jaya	F = 387.681
ANN-IEAMCGM-R vs. ANN-BFGS	$P = 0.000$ (Reject H_0)
ANN-IEAMCGM-R vs. LR	
ANN-IEAMCGM-R vs. GRNN	
Spring (W = 2928)	
ANN-IEAMCGM-R vs. ANN-IEAMGM-R	$H_0: e_1 = e_2 = e_3 = e_4 = e_5 = e_6 = e_7$
ANN-IEAMCGM-R vs. ANN-IEAM-R	
ANN-IEAMCGM-R vs. ANN-Jaya	F = 475.383
ANN-IEAMCGM-R vs. ANN-BFGS	$P = 0.000$ (Reject H_0)
ANN-IEAMCGM-R vs. LR	
ANN-IEAMCGM-R vs. GRNN	
Summer (W = 2928)	
ANN-IEAMCGM-R vs. ANN-IEAMGM-R	$H_0: e_1 = e_2 = e_3 = e_4 = e_5 = e_6 = e_7$
ANN-IEAMCGM-R vs. ANN-IEAM-R	
ANN-IEAMCGM-R vs. ANN-Jaya	F = 570.611
ANN-IEAMCGM-R vs. ANN-BFGS	$P = 0.000$ (Reject H_0)
ANN-IEAMCGM-R vs. LR	
ANN-IEAMCGM-R vs. GRNN	

To validate the effectiveness of our proposed hybrid model, we have compared our proposed model with six state-of-the-art forecasting models using two load data sets from two different countries, namely England (New Pool) and Australia (NSW). From results drawn, ANN-IEAMCGM-R outperformed other algorithms in terms of forecasting accuracy and generalization ability. Statistical analysis of ANN-IEAMCGM-R and state-of-the-art algorithms shown in tables and figures reveals that ANN-IEAMCGM-R generates the least error. Moreover, IEAM-R combined with the proposed mutation operator proved that forecasting requires a good combination of exploration and exploitation and has yielded accuracy improvement in almost all error metrics. In our future work, we will plan to implement our proposed hybrid methodology on different data sets.

Abbreviations

ACO	Ant Colony Optimization
ANN	Artificial Neural Network
BFGS	Broyden Fletcher Goldfarb Shanno Backpropagation
CGM	Controlled Gaussian Mutation
DC	Directional Change
DE	Differential Evolutionary
EAM	Environmental Adaptation Method
FTL	Follow The leader
GA	Genetic Algorithm
GRNN	Generalized Regression Neural Network
IEAM	Improved Environmental Adaptation Method
IEAM-R	Improved Environmental Adaptation Method with real parameter encoding
IEAMGM-R	IEAM-R with Gaussian mutation
IEAMCGM-R	IEAM-R with controlled Gaussian mutation
IA	Index of Agreement
LR	Linear Regression
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MLF	Multi-Layer Feed-forward
MTLF	Medium Term Load Forecaster
NN	Neural Network
PSO	Particle Swarm Optimization
STLF	Short Term Load Forecasting
SVM	Support Vector Machine

References

[1] Fan G-F, Peng L-L, Hong W-C, Sun F. Electric load forecasting by the SVR model with differential empirical mode decomposition and auto regression. Neurocomputing 2016;173:958–70.
[2] Xiao L, Wang J, Yang X, Xiao L. A hybrid model based on data preprocessing for electrical power forecasting. Int J Electr Power Energy Syst 2015;64:311–27.
[3] Raza MQ, Khosravi A. A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings. Renew Sustain Energy Rev 2015;50:1352–72.
[4] Dudek G. Pattern-based local linear regression models for short-term load forecasting. Electr Power Syst Res 2016;130:139–47.
[5] Goia A, May C, Fusai G. Functional clustering and linear regression for peak load forecasting. Int J Forecast 2010;26(4):700–11.
[6] Lee C-M, Ko C-N. Short-term load forecasting using lifting scheme and ARIMA models. Expert Syst Appl 2011;38(5):5902–11.
[7] Lee Y-S, Tong L-I. Forecasting time series using a methodology based on autoregressive integrated moving average and genetic programming. Knowl Base Syst 2011;24(1):66–72.
[8] Chan KY, Dillon TS, Singh J, Chang E. Neural-network-based models for short-term traffic flow forecasting using a hybrid exponential smoothing and Levenberg–Marquardt algorithm. IEEE Trans Intell Transp Syst 2012;13(2):644–54.
[9] Chakhchoukh Y, Panciatici P, Mili L. Electric load forecasting based on statistical robust methods. IEEE Trans Power Syst 2011;26(3):982–91.
[10] Li H-Z, Guo S, Li C-J, Sun J-Q. A hybrid annual power load forecasting model based on generalized regression neural network with fruit fly optimization algorithm. Knowl Base Syst 2013;37:378–87.
[11] M. Barman, N. D. Choudhury, S. Sutradhar, A regional hybrid Goa-SVM model based on similar day approach for short-term load forecasting in Assam, India, Energy .
[12] Hong T, Wang P, Willis HL. A naïve multiple linear regression benchmark for short term load forecasting. In: Power and energy society general meeting, 2011 IEEE; 2011. p. 1–6. IEEE.
[13] Weron R. Electricity price forecasting: a review of the state-of-the-art with a look into the future. Int J Forecast 2014;30(4):1030–81.
[14] Nie H, Liu G, Liu X, Wang Y. Hybrid of ARIMA and SVMs for short-term load forecasting. Energy Procedia 2012;16:1455–60.
[15] Taylor JW. Short-term load forecasting with exponentially weighted methods. IEEE Trans Power Syst 2012;27(1):458–64.
[16] Lee W-J, Hong J. A hybrid dynamic and fuzzy time series model for mid-term power load forecasting. Int J Electr Power Energy Syst 2015;64:1057–62.
[17] Che J, Wang J, Wang G. An adaptive fuzzy combination model based on self-organizing map and support vector regression for electric load forecasting. Energy 2012;37(1):657–64.
[18] Khosravi A, Nahavandi S, Creighton D, Srinivasan D. Interval type-2 fuzzy logic systems for load forecasting: a comparative study. IEEE Trans Power Syst 2012;27(3):1274–82.
[19] Hinojosa V, Hoese A. Short-term load forecasting using fuzzy inductive reasoning and evolutionary algorithms. IEEE Trans Power Syst 2010;25(1):565–74.
[20] Elattar EE, Goulermas J, Wu QH. Electric load forecasting based on locally weighted support vector regression. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 2010;40(4):438–47.
[21] Ahmad A, Hassan M, Abdullah M, Rahman H, Hussin F, Abdullah H, Saidur R. A review on applications of ANN and SVM for building electrical energy consumption forecasting. Renew Sustain Energy Rev 2014;33:102–9.
[22] Ceperic E, Ceperic V, Baric A, et al. A strategy for short-term load forecasting by support vector regression machines. IEEE Trans Power Syst 2013;28(4):4356–64.
[23] Niu D, Wang Y, Wu DD. Power load forecasting using support vector machine and ant colony optimization. Expert Syst Appl 2010;37(3):2531–9.
[24] Ekonomou L. Greek long-term energy consumption prediction using artificial neural networks. Energy 2010;35(2):512–7.
[25] Buhari M, Adamu SS, et al. Short-term load forecasting using artificial neural network. In: Proceedings of the international MultiConference of engineers and computer scientists, vol. 1; 2012. p. 806–11.
[26] Zhang Y, Wu L, Wang S. Bacterial foraging optimization based neural network for short-term load forecasting. J Comput Inf Syst 2010;6(7):2099–105.
[27] Lu N, Zhou J-z, He Y-y. Particle swarm optimization-based neural network model for short-term load forecasting. Power System Prot Contr 2010;38(12):65–8.
[28] Wang L, Zeng Y, Chen T. Back propagation neural network with adaptive differential evolution algorithm for time series forecasting. Expert Syst Appl 2015;42(2):855–63.
[29] Singh P, Dwivedi P. Integration of new evolutionary approach with artificial neural network for solving short term load forecast problem. Appl Energy 2018;217:537–49.
[30] Hippert HS, Pedreira CE, Souza RC. Neural networks for short-term load forecasting: a review and evaluation. IEEE Trans Power Syst 2001;16(1):44–55.
[31] Wang J, Yang W, Du P, Li Y. Research and application of a hybrid forecasting framework based on multi-objective optimization for electrical power system. Energy 2018a;148:59–78.
[32] Song K-B, Baek Y-S, Hong DH, Jang G. Short-term load forecasting for the holidays using fuzzy linear regression method. IEEE Trans Power Syst 2005;20(1):96–101.
[33] Ying L-C, Pan M-C. Using adaptive network based fuzzy inference system to forecast regional electricity loads. Energy Convers Manag 2008;49(2):205–11.
[34] Cao G, Wu L. Support vector regression with fruit fly optimization algorithm for seasonal electricity consumption forecasting. Energy 2016;115:734–45.
[35] Hong W-C. Application of chaotic ant swarm optimization in electric load forecasting. Energy Policy 2010;38(10):5830–9.
[36] Zhang X, Wang J, Zhang K. Short-term electric load forecasting based on singular spectrum analysis and support vector machine optimized by Cuckoo search algorithm. Electr Power Syst Res 2017;146:270–85.
[37] Ghayekhloo M, Menhaj M, Ghofrani M. A hybrid short-term load forecasting with a new data preprocessing framework. Electr Power Syst Res 2015;119:138–48.
[38] Azadeh A, Ghaderi S, Tarverdian S, Saberi M. Integration of artificial neural networks and genetic algorithm to predict electrical energy consumption. Appl Math Comput 2007;186(2):1731–41.
[39] Ling S-H, Leung FH-F, Lam H-K, Lee Y-S, Tam PK-S. A novel genetic-algorithm-based neural network for short-term load forecasting. IEEE Trans Ind Electron 2003;50(4):793–9.
[40] Quan H, Srinivasan D, Khosravi A. Short-term load and wind power forecasting using neural network-based prediction intervals. IEEE Trans Neural Networks Learn Sys 2014;25(2):303–15.
[41] Bashir Z, El-Hawary M. Applying wavelets to short-term load forecasting using PSO-based neural networks. IEEE Trans Power Syst 2009;24(1):20–7.
[42] L. Wang, S. Lv, Y.-R. Zeng, Effective sparse adaboost method with ESN and FOA for industrial electricity consumption forecasting in China, Energy .
[43] Chaowanawatee K, Heednacram A. Implementation of cuckoo search in RBF

neural network for flood forecasting. In: Computational intelligence, communication systems and networks (CICSyN), 2012 fourth international conference on, IEEE; 2012. p. 22–6.
[44] ZOU Z-d, SUN Y-m, ZHANG Z-s. Short-term load forecasting based on recurrent neural network using ant colony optimization algorithm. Power Syst Technol 2005;3:59–63.
[45] Singh P, Mishra K, Dwivedi P. Enhanced hybrid model for electricity load forecast through artificial neural network and Jaya algorithm. In: Intelligent computing and control systems (ICICCS), 2017 international conference on, IEEE; 2017. p. 115–20.
[46] Kouhi S, Keynia F, Ravadanegh SN. A new short-term load forecast method based on neuro-evolutionary algorithm and chaotic feature selection. Int J Electr Power Energy Syst 2014;62:862–7.
[47] L. Peng, S. Liu, R. Liu, L. Wang, Effective long short-term memory with differential evolution algorithm for electricity price prediction, Energy .
[48] Sharma B, Prakash R, Tiwari S, Mishra K. A variant of environmental adaptation method with real parameter encoding and its application in economic load dispatch problem. Applied Intelligence; 2017. p. 1–21.
[49] Singh N, Mohanty SR, Shukla RD. Short term electricity price forecast based on environmentally adapted generalized neuron. Energy 2017b;125:127–39.
[50] Mishra K, Tiwari S, Misra AK. Improved environmental adaption method for solving optimization problems. Comput Intell Intelligent Sys 2012:300–13.
[51] Mishra K, Tiwari S, Misra AK. Improved environmental adaption method and its application in test case generation. J Intell Fuzzy Syst 2014;27(5):2305–17.
[52] Hinterding R. Gaussian mutation and self-adaption for numeric genetic algorithms. In: Evolutionary computation, 1995., IEEE international conference on, vol. 1; 1995. p. 384. IEEE.
[53] Higashi N, Iba H. Particle swarm optimization with Gaussian mutation. In: Swarm intelligence symposium, 2003. SIS'03. Proceedings of the 2003 IEEE; 2003. p. 72–9. IEEE.
[54] Wu Q. Power load forecasts based on hybrid PSO with Gaussian and adaptive mutation and Wv-SVM. Expert Syst Appl 2010;37(1):194–201.
[55] Bento P, Pombo J, Calado M, Mariano S. A bat optimized neural network and wavelet transform approach for short-term price forecasting. Appl Energy 2018;210:88–97.
[56] Hernández L, Baladrón C, Aguiar JM, Carro B, Sánchez-Esguevillas A, Lloret J. Artificial neural networks for short-term load forecasting in microgrids environment. Energy 2014;75:252–64.
[57] Han J, Pei J, Kamber M. Data mining: concepts and techniques. Elsevier; 2011.
[58] Işık E, Inallı M. Artificial neural networks and adaptive neuro-fuzzy inference systems approaches to forecast the meteorological data for HVAC: the case of cities for Turkey. Energy 2018;154:7–16.
[59] M. Gülüm, F. K. Onay, A. Bilgin, Comparison of viscosity prediction capabilities of regression models and artificial neural networks, Energy .
[60] Baldwin JM. A new factor in evolution (Continued). Am Nat 1896;30(355):536–53.
[61] Mishra K, Tiwari S, Misra A. A bio inspired algorithm for solving optimization problems. In: Computer and communication technology (ICCCT), 2011 2nd international conference on, IEEE; 2011. p. 653–9.
[62] Li C, Zhang N, Lai X, Zhou J, Xu Y. Design of a fractional-order PID controller for a pumped storage unit using a gravitational search algorithm based on the Cauchy and Gaussian mutation. Inf Sci 2017;396:162–81.
[63] Deng Y, Liu Y, Zhou D. An improved genetic algorithm with initial population strategy for symmetric TSP. Math Probl Eng 2015;2015:212794.
[64] Cha SK, Woo M, Brumley D. Program-adaptive mutational fuzzing. In: Security and privacy (SP), 2015 IEEE symposium on, IEEE; 2015. p. 725–41.
[65] Zhang Y, Wang S, Phillips P, Ji G. Binary PSO with mutation operator for feature selection using decision tree applied to spam detection. Knowl Base Syst 2014;64:22–31.
[66] Das S, Abraham A, Chakraborty UK, Konar A. Differential evolution using a neighborhood-based mutation operator. IEEE Trans Evol Comput 2009;13(3):526–53.
[67] Zhao N, Wu Z, Zhao Y, Quan T. Ant colony optimization algorithm with mutation mechanism and its applications. Expert Syst Appl 2010;37(7):4805–10.
[68] Wang J, Du P, Niu T, Yang W. A novel hybrid system based on a new proposed algorithmMulti-Objective Whale Optimization Algorithm for wind speed forecasting. Appl Energy 2017;208:344–60.
[69] Holley JW, Guilford JP. A note on the G index of agreement. Educ Psychol Meas 1964;24(4):749–53.
[70] Dong Y, Zhang Z, Hong W-C. A hybrid seasonal mechanism with a chaotic cuckoo search algorithm with a support vector regression model for electric load forecasting. Energies 2018;11(4):1009.
[71] Fan G-F, Peng L-L, Hong W-C. Short term load forecasting based on phase space reconstruction algorithm and bi-square kernel regression model. Appl Energy 2018;224:13–33.