Energy 161 (2018) 130–142

Contents lists available at ScienceDirect

Energy

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

Two stage forecast engine with feature selection technique and improved meta-heuristic algorithm for electricity load forecasting

Noradin Ghadimi [a], Adel Akbarimajd [b, *], Hossein Shayeghi [b], Oveis Abedinia [a, c]

[a] Department of Engineering, Ardabil Branch, Islamic Azad University, Ardabil, Iran
[b] Department of Electrical Engineering, Faculty of Technical Engineering, University of Mohaghegh Ardabili, Ardabil, Iran
[c] Department of Electrical Engineering, Budapest University of Technology and Economics, Budapest, Hungary

ARTICLE INFO

Article history:
Received 19 April 2018
Received in revised form
20 June 2018
Accepted 14 July 2018
Available online 18 July 2018

Index Terms:
Load forecast
Feature selection
Two stage forecast engine
Ridgelet NN
Elman NN
Intelligent algorithm

ABSTRACT

Short-term load forecasting is of major interest for the restructured environment of the electricity market. Accurate load forecasting is essential for effective power system operation, but electricity load is non-linear with a high level of volatility. Predicting such complex signals requires suitable prediction tools. This paper proposes a hybrid forecast strategy including novel feature selection technique, and a complex forecast engine based on a new intelligent algorithm. The electricity load signal is first filtered by feature selection technique to select appropriate candidates as input for the forecast engine. Then, the proposed two stage forecast engine is implemented based on ridgelet and Elman neural networks. All forecast engine parameters are chosen based on a novel intelligent algorithm to improve its accuracy and capability. Different electricity markets were considered as test cases to compare the proposed method with several current algorithms. Additionally, the proposed forecasting model measures the absolute forecasting errors in this work (among seven types of measurements i.e., absolute forecasting errors, measures based on percentage errors, symmetric errors, measures based on relative errors, scaled errors, relative measures and other error measures). The results validate the effectiveness of the proposed method.

1. Introduction

Short-term load forecasting (STLF) is of major interest in the deregulated electricity market for effective and efficient dispatch of diverse energy loads [1]. Load demand monitoring is very important for optimal power system operation, and generation, distribution, and transmission aspects all require accurate and sufficient predicted electricity load. In addition to decreasing generation cost, this is also essential for power system reliability. System operators use load predictions as the basis for off-line network analysis to regulate and safeguard the system, activating corrective actions as required, such as load shedding, power purchase, bringing peaking units on line, etc. [2].

There have been many studies regarding load prediction problem, dating back to the late 1960s [3]. Traditional methods were based on mathematical methods such as regression analysis [4], but more recent approaches include Kalman filtering [5], autoregressive integrated moving average (ARIMA) [6], Box–Jenkins models [7], state space models [8], etc.

Artificial intelligence (AI) methods have been recently proposed for load forecasting, including artificial neural networks (ANNs) expert systems [9], fuzzy inference methods [10,11], genetic programming [12], evolutionary computation [13], support vector regression (SVR) [14], etc.

The various proposed methods have the advantages of being simple methods and easy-to-use, but their key disadvantage is that linear analysis is not suitable for non-linear load series prediction, which can make problems. Several new prediction methods can be effective for non-linear signals without complex mathematical formula. For example ANN [15,16] and radial basis function (RBF) NN [17] based prediction methods.

Hybrid two-stage models have also been proposed for STLF [18], using the second stage as the forecast engine incorporating linear regression; dynamic programming; and support vector machine (SVM). Fixed size least squares support vector machines (LS-SVM) using an autoregressive exogenous nonlinear autoregressive exogenous structure, have also been implemented and outperformed the linear model [19]. A new load forecasting model

* Corresponding author.
E-mail address: akbarimajd@uma.ac.ir (A. Akbarimajd).

https://doi.org/10.1016/j.energy.2018.07.088
0360-5442/

based on seasonal recurrent support vector regression (SVR) was proposed that applied the chaotic artificial bee colony (CABC) intelligent algorithm to improve forecast engine accuracy [20]. The model outperformed some well-known methods including ARIMA and trend fixed seasonally adjusted ε-SVR. Other univariate exponentially weighted methods have been proposed to address STLF using singular value decomposition [21].

Although different prediction models have been presented by various researchers, an accurate prediction model is demanded yet. For this purpose, a new prediction mode which provide better results with low error is presented in this paper based on the following contributions.

a. A new feature selection technique that filters the input load signal using appropriate criteria for candidate selection to provide correct information and simple learning data.
b. A two stage forecast engine using a hybrid model to provide accurate predictions using a ridgelet neural network (RNN) for the initial forecast block and an Elman neural network (ENN) as the final forecast block.
c. A new version of optimization algorithm is presented which is applied on load forecasting problem. Optimal selection of forecast engine parameters is addressed as an optimization problem.

This paper is organized as follows: Section 2 presents the proposed feature selection technique, and Section 3 presents the proposed forecast engine model. Section 4 presents the new optimization algorithm and Section 5 provides numerical analysis of prediction and optimization. Section 6 provides our conclusions.

2. Feature selection

2.1. Review

Many feature selection techniques have been proposed to find a minimum subset of primary candidates that preserves the main information within the original set, but allows simple and easy analysis [22]. Unimportant features can entrap data analysis and modeling without providing additional useful information. Such unimportant features are max-redundant or min-irrelative (which can be denied) [23]. In other words, more features can make a serious problem in forecast engine training process due to redundancy and relevancy, to do the prediction which can be named as candidates. By this way, the best candidates of feature selection are considered as input of forecast engine to do the prediction.

However, feature selection technique using maximum relevancy criteria only fails to select redundant or highly correlated candidates. Removing redundant candidates will not produce desirable results. Thus, the minimization process must retain appropriate redundant, as well as non-redundant, candidates.

In the minimal redundancy maximal relevance (mRMR) method [24], candidates are selected using a greedy method, one of the strong models to select good features. However, once a candidate is selected, it cannot be deleted, and some redundant features can include important information [25]. Wrapper based methods have been proposed to address such problems [25], but they can be problematic if the combination of specific classifiers are suboptimal. Thus, this model does not generally provide the best features.

Therefore, we applied two stage mutual information feature selection technique based on the transductive model (MIT-MIT), which works by data training and unlabeled test data. An optimization algorithm was applied to improve predictive efficiency as well as computational performance of the proposed method.

2.2. Proposed feature selection technique

In the proposed MIT-MIT method, we enhanced the efficiency of classic inductive learning models. Relevancy criteria (as maximization) and redundancy criteria (as minimization) were considered as independent components. Since maximization (i.e., relevancy) evaluates mutual information (MI) between the selected candidates and class values where it is not able to extract new extra unlabeled test data, it cannot improve the transductive version of the model [26]. On the other hand, minimization (i.e., redundancy) evaluates the MI of all selected candidates without considering their class value. Accordingly, unlabeled test samples in related components can be considered as the base transductive setting [22], and the objective function for feature selection process can be expressed as

$$\max_{x_j \in |X - S_{m-1}} \left[I\left(x_j^{training}; c^{training}\right) - \frac{1}{m-1} \sum_{x_j \in S_{m-1}} I\left(x_j^{training+test}; x_i^{training+test}\right) \right], \quad (1)$$

where the j^{th} feature vector is represented by $x_j^{training}$, which contains the training data; the training and test data is in $x_j^{training+test}$; the vector of class value is $c^{training}$ (with no test data); and the greedy algorithm was implemented for maximization to find near optimal candidates [23].

To evaluate the *MI* factor of features, as two vectors of *X*, *Y*,

$$I(X,Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log\left(\frac{p(x,y)}{p(x)p(y)}\right), \quad (2)$$

where marginal probability is represented by p(x), i.e., p(X = x); the joint probability by p(x,y), i.e., p(X = x, Y = y). The discrete case can be evaluated by summing the corresponding values whereas the continuous case uses the integral,

$$I(X,Y) = \int_Y \int_X p(x,y) \log\left(\frac{p(x,y)}{p(x)p(y)}\right) dxdy. \quad (3)$$

Using Equations (3) and (2), we can evaluate the relevancy of candidates, but similar phenotypes for different samples can lead to large rounding errors. For this purpose, we can calculate the z-score in phenotype value through each sample which can write as $\frac{x-\mu}{\sigma^2}$. Thus, the value of discretized can be evaluate as following relationship:

$$discretized\ value = \begin{cases} -1 & if\ z - score < -1 \\ 1 & if\ z - score > 1 \\ 0 & otherwise \end{cases}, \quad (4)$$

and calculate the systematic error for two vectors as

$$\Delta I(X,Y) \approx \frac{M_{xy} - M_x - M_y + 1}{2N}, \quad (5)$$

where M_x represents the histogram bins for *X*, and similarly for M_y and $M_{x,y}$; and *N* is the number of samples. Thus, the error is monotonic in *N*, i.e., the error decreases for large *N* and vice versa.

We calculated the objective function for each candidate based on the naïve algorithm $O\left(\frac{N^2}{2} \times M\right)$, where *M* is the number of candidates, a small value, and *N* represents the number of target features. However, the naïve greedy method is not efficient for new

redundancy evaluation of MI between selected candidates. Therefore, an effective greedy method is required. It is not necessary to calculate each x_j in $\sum_{x_i \in S_{m-1}} I(x_j; x_i)$, and from the progressive behavior of candidates, the difference of S_{m-1} and S_{m-2} is the $(m\text{-}1)^{th}$ candidate. Therefore, recalculation of the sum of MI for x_j and x_i is not necessary where $1 \le i \le m-2$. On the other hand, $\sum_{x_i \in S_{m-2}} I(x_j; x_i)$ and $\sum_{x_i \in S_{m-1}} I(x_j; x_i)$ vary only by $I(x_j; x_{m-1})$. Thus, $\sum_{x_i \in S_{m-1}} I(x_j; x_i)$ can be saved for each stage and recalculated in the next. The proposed algorithm is significantly faster than the naïve method.

3. Hybrid forecast engine

Previously we proposed a forecast engine based on hybrid blocks due to difficulties of single neural network (NN) blocks, such as inappropriate input and output mapping [27]. Different learning algorithms can envelop the leakage of each other through the prediction process [28,29]. Therefore, a two stage forecast engine based on RNN and ENN is implemented in this paper.

Fig. 1 shows the main structure of the proposed forecast engine, incorporating two main stages, RNN and ENN. Both blocks are improved by an intelligent algorithm, described in Section 4, that ensures the best results for local search. The blocks use different learning algorithms to tackle the prediction process drawbacks discussed above. To tackle error function evaluation we propose cross-validation for the forecast engine [28]. In this training mode, each block is performed by the remaining section of the training set, so the previous day becomes unseen for each block. Thus, the forecast error for each block for a given specific day can be estimated based the known error for the forecast day.

Regarding the forecasted values, evaluations from other prediction techniques can be useful information for load forecasting, and we use this strategy for the proposed hybrid model, as shown in Fig. 1. The ENN uses the load forecast for the next hour, but before that, the RNN block provides an initial forecast of the load signal for the next hour.

In the proposed model, the RNN based lagged loads and related exogenous data (e.g. temperature) were included to provide accurate predictions based on high data correlation. The second block (ENN) used $\widehat{L}_{(t+1)}$ as training. Thus, the second block can evaluate the signal prediction error in the learning phase to provide better results once $L_{(t+1)}$ is not accessible. To prevent impregnation as well as sweep effects of candidate variable ranges in the proposed model, inputs and outputs were all normalized to [0,1],

$$x_n = (x - r_{\min})/(r_{\max} - r_{\min}), \tag{6}$$

where x_n is the normalized parameter, and r_{max} and r_{min} are the upper and lower bound of the training set data, respectively.

After this process, the candidates shuffled 1000 times for each blocks separately. By this method, we can improve the randomness of data and also remove the ordinal bias efficacy so, an optimization tool is needed. The outputs of both blocks were input to the preprocessor, in the same way as other input features.

3.1. Ridgelet neural network

The proposed Ridgelet neural network (RNN) is presented in this section. The structure of RNN is depicted in Fig. 2 where, the activation function of $\psi(.)$ [30] can be written as:

$$\int \frac{\left|\widehat{\psi}(\omega)\right|^2}{|\omega|^M} d\omega < \infty, \tag{7}$$

where $\widehat{\psi}(\cdot)$ is the Fourier transform (FT) of $\psi(.)$, and M is the spatial dimension (for the load prediction procedure). Thus,

$$\psi_\gamma(Z(t)) = \frac{1}{\sqrt{a}}\psi\left(\frac{<U, Z(t)> - b}{a}\right), \tag{8}$$

where γ, a, b, and U are ridgelet parameters,

$$\begin{aligned} &\gamma = (a, b, U),\\ &a, b \in R,\\ &a > 0\\ &U \in US^M,\\ &\|U\| = 1 \end{aligned} \tag{9}$$

where the unit sphere is US^M in M dimensional space, b is the location, a defines the scale, and U defines the direction. Therefore, the RNN output is

$$WP(t) = \sum_{i=1}^{N} w_i.\psi\left(\frac{<U_i, Z(t)> - b_i}{a_i}\right), \tag{10}$$

where N defines the number of hidden layer nodes.

In this block of the forecast engine, the linear combination of ridgelets and related RNN weights, w_i, is considered to improve the prediction ability.

To train the first phase of the hybrid forecast engine, we define the parameters of RNN as

$$X = [w_1, w_2, \ldots, w_N, a_1, a_2, \ldots, a_N, b_1, b_2, \ldots, b_N, U_1, U_2, \ldots, U_N] \tag{11}$$

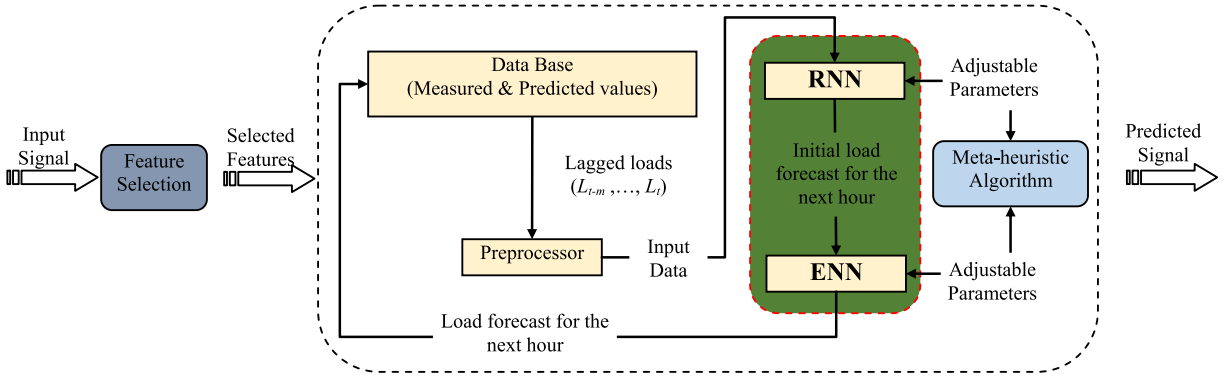


Fig. 1. Main structure of the proposed prediction model.

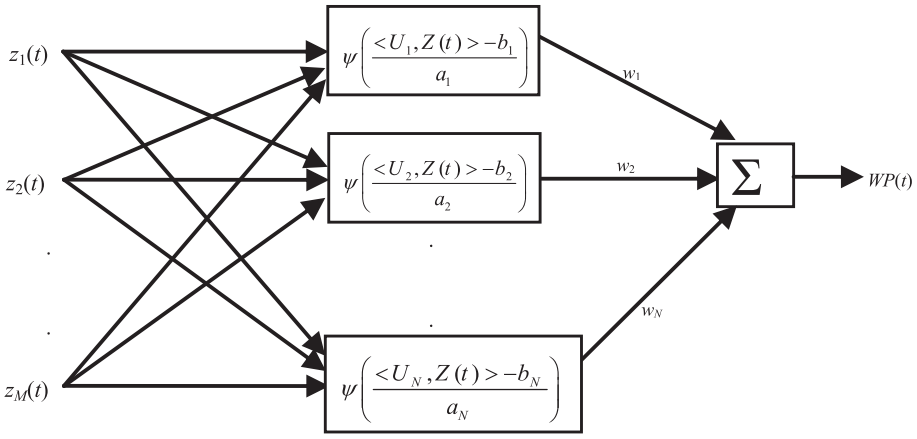


Fig. 2. Proposed RNN structure.

where *w*, *a*, and *b* are scalar variables in RNN; and the last parameter is the direction, which is based on the vectors

$$U_1 = [u_{11}, u_{12}, \ldots, u_{1M}], U_2 = [u_{21}, u_{22}, \ldots, u_{2M}], \ldots, U_N = [u_{N1}, u_{N2}, \ldots, u_{NM}]. \quad (12)$$

Following this definition, we want to train the RNN through optimizing these parameters, where the optimization objective function is to minimize the training error. We used a new intelligent algorithm to train the RNN that addressed the drawbacks of previous training algorithms [31]. In this model, each individual set was based on the proposed parameters in Equation (11). Then, the objective function was calculated and minimized, and repeated until the stop criteria was achieved. After feature selection procedure, the 50 previous days to each prediction day were considered as its corresponding training period. Hence, we had 49 days for training the 24 hourly samples of the last day. Test error was also monitored, and when error increased, RNN performance decreased due to overfitting, so the algorithm terminated.

3.2. Elnam neural network

This section presents the second forecast engine block, comprising ENN based on four main layers: input, hidden, context, and output, as shown in Fig. 3. The proposed model was based on a feed-forward loop including the input, hidden, and output layers through weighted connection between neighboring layers. *R* nodes were chosen for the first layer, *N* nodes for the hidden and context layers, and *M* nodes for the output layer [32]. The model can be expressed as follows.

3.2.1. Input layer

$$u_q(k) = e_q(k), \quad q = 1, 2, \ldots, R, \quad (13)$$

where,$e_q(k)$ and $u_q(k)$ present the input and output of the first layer, respectively, in the *k*th iteration step.

3.2.2. Hidden layer

Input of the j^{th} hidden layer neuron can be expressed as

$$v_j(k) = \sum_{l=1}^{N} \omega_{jl}^1(k) x_l^c(k) + \sum_{q=1}^{R} \omega_{jq}^2(k) u_q(k), \quad j = 1, 2, \ldots, N, \quad (14)$$

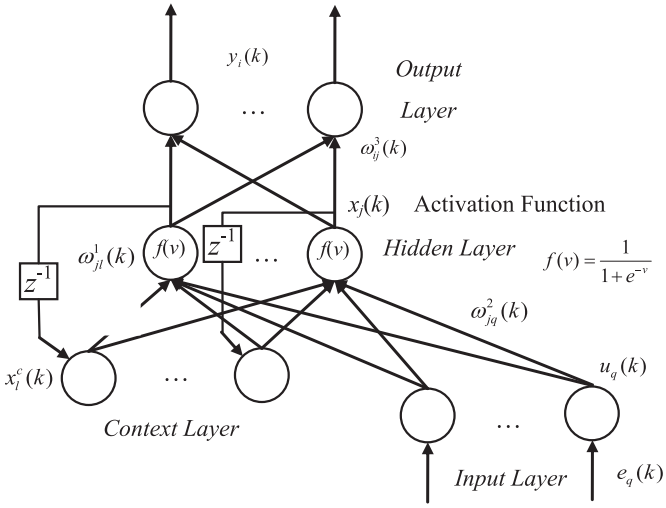


Fig. 3. Proposed ENN block in the forecast engine.

where $x_l^c(k)$is the output of l^{th} context layer, and $\omega_{jl}^1(k)$is the weight of the l^{th} context layer to the j^{th} hidden layer. Thus, in the *j*th input layer neuron, the weight for the j^{th} hidden layer neuron is $\omega_{jq}^2(k)$ [33]. For the Chebyshev orthogonal function on the interval [−1, 1], the input of the hidden layer was normalized,

$$\bar{v}_j(k) = \frac{v_j(k)}{\max_{1 \le p \le N}(|v_p(k)|)}, \quad j = 1, 2, \ldots, N, \quad (15)$$

and the output of j^{th} hidden layer neuron is

$$x_j(k) = f_i[\bar{v}_j(k)], \quad j = 1, 2, \ldots, N. \quad (16)$$

3.2.3. Context layer

This output from this layer is

$$x_l^c(k) = \alpha x_l^c(k-1) + x_1(k-1), \quad l = 1, 2, \ldots, N, \quad (17)$$

where x_1 is self-connection feedback gain between [0, 1]. When $\alpha = 0$, this network is reduced to a classic ENN [34].

3.2.4. Output layer

The final output is

$$\tilde{y}_i(k) = \sum_{j=1}^{N} \omega_{ij}^3(k) x_j(k), \quad i = 1, 2, ..., M, \tag{18}$$

where $\omega_{ij}^3(k)$ is the weight of j^{th} hidden layer to i^{th} output layer neuron. All weights of the proposed block were optimally chosen by a proposed meta-heuristic algorithm.

3.3. Optimization in forecast engine

In this paper, to increase the prediction accuracy of RNN and ENN based forecasting engine, an optimization tool is needed for fine tuning the free parameters. By this optimization, we can increase the prediction accuracy of proposed prediction model and reduce the error. Moreover, application of proposed algorithm on training of RNN and ENN, optimizes the adjustable parameters of forecast engine to learn its associated input/output function. The stochastic search method is a modified version of SSO algorithm. To effectively train each block by proposed algorithm, its generalization capability should be carefully monitored along the training process to avoid overfitting/underfitting, which is a serious problem for training.

4. Proposed meta-heuristic algorithm

This section presents the proposed chaotic binary shark smell optimization (CBSSO) algorithm. We first describe the classic version of the method, and then improvements are added to increase the search capability. The proposed algorithm is based on search model of shark to find the prey. Regarding this fact that prey location is distinguished by odor particles in search environment, the gradient of particles is the best way to find the prey. In other hand, the gradient ability is not the only operator of SSO and other operators improve its abilities based on; local search, velocity and position updates of odors. Also, in this paper the chaotic nature and binary ability of proposed method improve the basic SSO algorithm.

4.1. Shark smell optimization

This algorithm is introduced by Abedinia et al. in Ref. [35], which motivated by the shark's ability for finding prey. In this process, the sharks movement is based on odor concentration growth, to find the prey. This movement is modeled mathematically as an optimization problem to find the best solution. Additional information of this algorithm can be found in Ref. [35]. This algorithm consists of main parts as follows.

4.1.1. Initialization

The optimization algorithm starts by generating initial decision variable values determined based on shark position in an SSO population,

$$\mathrm{SX_i} = \left[\mathrm{sx}_1^1, \mathrm{sx}_2^1, \ldots, \mathrm{sx}_{NP}^1\right], \quad NP = \text{Population Size}, \tag{19}$$

where sx_i^1 is the ith initial position vector, i.e., the ith initial candidate solution. The velocity for each individual is

$$\mathrm{SP}_i = \left[\mathrm{sp}_{i,1}^1, \mathrm{sp}_{i,2}^1, \ldots, \mathrm{sp}_{i,ND}^1\right], \quad i = 1, ...NP, \tag{20}$$

where $sp_{i,j}^1$ is the jth decision variable randomly generated in each individual valid range. Thus, each objective function for related individuals can be defined by $OF(SX_{NP})$, and will be saved for each one after calculation. The algorithm starts at iteration zero.

4.2. Movement process

After initialization, the shark moves to attain prey. This movement is based a combination of "*forward*" and "*rotational*" movement. The shark direction based on high odor concentration is evaluated by

$$sp_{i,j}^m = \mu_m \cdot R1 \cdot \frac{\partial(OF)}{\partial x_j} \left| sx_{i,j}{}^m + \alpha_m \cdot R2 \cdot sp_{i,j}^{m-1} \quad j = 1, ..., ND, i = 1, ...NP, m = 1, ..., M, \right. \tag{21}$$

where $\nabla(OF)$ is the objective function gradient, μ_m is the gradient constant, m is the stage number, M is the maximum number of stages through forward movement process, μ_m and $\alpha_m \in (0,1]$, and $R1$ and $R2$ are generated in the [0, 1] interval.

For the shark speed limitation, we can express (21) as

$$\left|sp_{i,j}^m\right| = Min\left[\left|\mu_m \cdot R1 \cdot \frac{\partial(OF)}{\partial x_j}\right| sx_{i,j}^m + \alpha_m \cdot R2 \cdot sp_{i,j}^{m-1}\right|, \left|\gamma_m \cdot sp_{i,j}^{m-1}\right|\Bigg] \quad j = 1, ..., ND, i = 1, ..., NP, m = 1, ..., M, \tag{22}$$

where γ_m is the upper bound of current velocity in terms of the previous one. Every element $sp_{i,j}^m$ of the vector SP_i^m is determined through (22). For the global search of shark (forward movement) the new position evaluation can be presented by

$$\mathrm{GY}_i^{m+1} = \mathrm{SX}_i^m + SP_i^m \cdot \Delta t_m \quad i = 1, ..., NP \quad m = 1, ..., M, \tag{23}$$

where Δt_m is the time interval for m^{th} stage. Thus, the local shark search can be expressed as

$$NX_i^{m+1,l} = GY_i^{m+1} + R3 \cdot \Delta GY_i^{m+1} \quad l = 1, ..., L \quad i = 1, ..., NP \quad m = 1, ..., M, \tag{24}$$

where $l = 1, \ldots, L$; $R3$ is generated randomly between $(-1,+1)$; and L is the number of points in the local search of every stage.

The best of the points searched in the forward movement and local search is selected by shark, which is modeled in the SSO algorithm as

$$\mathrm{SX}_i^{k+1} = \mathrm{argmax}\left\{OF\left(\mathrm{GY}_i^{m+1}\right), OF\left(\mathrm{NX}_i^{m+1,1}\right), ..., OF\left(\mathrm{NX}_i^{m+1,L}\right)\right\} \quad i = 1, ...NP \tag{25}$$

4.3. Enhanced shark smell optimization

4.3.1. Chaotic enhancement

SSO was improved to increase its local and global search capability. The chaotic algorithm was added based on a logistic map that randomly generated various areas in solution space [36]. Therefore, after calculating (23) for each SSO individual generation of $\mathrm{GY}_{\mathrm{iit}+1}$, other candidate solutions were estimated as

$$ChaoGY_{i,j}^{m+1,1} = GY_{i,j}^{m+1} + chm_{i,h}^{m} \cdot \left(GY_{i,j}^{m+1} - SX_{i,j}^{m}\right), \quad j = 1,2,...,ND, \quad i = 1,2,...,NP \tag{26}$$

$$ChaoGY_{i,j}^{m+1,2} = GY_{i,j}^{m+1} + chm_{i,h}^{m} \cdot \left(SX_{i,j}^{m} - GY_{i,j}^{m+1}\right), \quad j = 1,2,...,ND, \quad i = 1,2,...,NP \tag{27}$$

$$ChaoGY_{i,j}^{m+1,3} = SX_{i,j}^{m} + chm_{i,h}^{m} \cdot \left(SX_{i,j}^{m} - GY_{i,j}^{m+1}\right), \quad j = 1,2,...,ND, \quad i = 1,2,...,NP \tag{28}$$

and

$$ChaoGY_{i,j}^{m+1,1} = SX_{i,j}^{m} + chm_{i,h}^{m} \cdot \left(GY_{i,j}^{m+1} - SX_{i,j}^{m}\right), \quad j = 1,2,...,ND, \quad i = 1,2,...,NP \tag{29}$$

where $j = 1, 2, \ldots, ND$; $i = 1, 2, \ldots, NP$; and $chm^{it}_{i,j}$ is the chaotic number produced for the j^{th} decision variable for the i^{th} individual in the m^{th} iteration. This operator was implemented based on the logistic map function,

$$chm_{i,j}^{m} = 4.chm_{i,j}^{m-1} \cdot \left(1 - chm_{i,j}^{m-1}\right) \quad 0 < chm_{i,j}^{m} < 1$$
$$chm_{i,j}^{0} = md_4, \quad chm_{i,j}^{0} \notin \{0.25, 0.5, 0.75\}, \tag{30}$$

where the proposed logistic map function generates more diverse values for $chm^{it}_{i,j}$ between (0,1). Therefore, (25) can be expressed as

$$SX_i^{m+1} = \text{argmax}\left\{ \begin{array}{l} OF\left(GY_i^{m+1}\right), OF\left(ChaoGY_i^{m+1,1}\right), OF\left(ChaoGY_i^{m+1,2}\right), \\ OF\left(ChaoGY_i^{m+1,3}\right), OF\left(ChaoGY_i^{m+1,4}\right), OF\left(NX_i^{m+1,1}\right), \ldots, OF\left(NX_i^{m+1,L}\right) \end{array} \right\}. \tag{31}$$

4.4. Binary enhancement

SSO is a continuous based optimization algorithm due to the gradient calculation in (21). To increase the ability of this algorithm, we changed it to a binary version using binary integer decision variables, as shown in Fig. 4. Accordingly, the proposed transformation to binary domain Bin(.) for the binary variables was

$$bsx_{i,j} = Bin(sx_{i,j}) = \left\{ \begin{array}{ll} 0.5.Rand(0,1) & if \quad sx_{i,j} = 0 \\ 1 - 0.5.Rand(0,1) & if \quad sx_{i,j} = 1 \end{array} \right., \tag{32}$$

where $Rand(0,1)$ is a random parameter generated between (0,1). Using (32), we can generate each individual in the binary domain and map them to $bsx_{i,j}$ between (0,1) in real format. Hence, all binary individuals were converted to the desired format in all stages of forward movement, rotational movement, and chaotic enhancement. Finally, all of the evolved individuals were returned to the appropriate format using the sigmoid function,

$$Sig(bsx_{i,j}) = \frac{1}{1 + \exp(-bsx_{i,j})}, \tag{33}$$

where $Sig(bsx_{i,j}) \in (0,1)$ and $Sig(bsx_{i,j})$ are transformed into the binary domain as

$$sx_{i,j} = \left\{ \begin{array}{ll} 1 & Rand(0,1) < Sig(bsx_{i,j}) \\ 0 & Rand(0,1) > Sig(bsx_{i,j}) \end{array} \right.. \tag{34}$$

4.5. Training

To train the proposed forecast engine based intelligent algorithm, we implemented the following steps.

1. Select the training period, i.e., 49 previous days, and remove the target day before forecasting this day for cross-validation. This day's data must be unseen in the forecast and preprocessor steps.

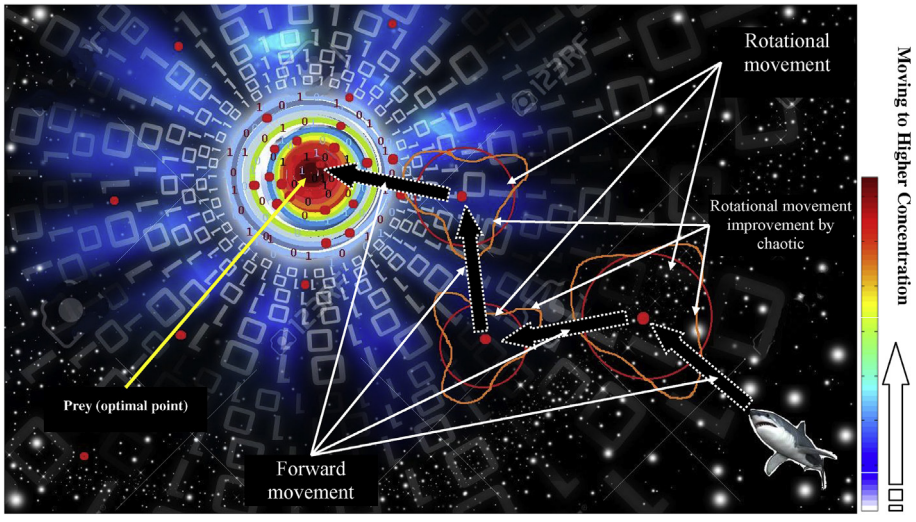


Fig. 4. Proposed Chaotic-binary SSO structure.

2. Evaluate candidate inputs, which consisted of lagged load from 200 previous hours.
3. Select the best features using the proposed feature selection technique filter applied to the signal.
4. Consider the preprocessor output as training samples for input features, normalized for next step.
5. Train the output of step 4 using the hybrid mechanism of the RNN and ENN blocks based intelligent algorithm.
6. Optimize all free RNN and ENN parameters using the proposed meta-heuristic algorithm.

The signal predicted from these steps was then used to evaluate forecast error and accuracy, and compared with other methods.

To validate the proposed model, this model is compared with well-known prediction strategies, and analyzed the proposed optimization algorithm efficacy using standard test cases and benchmark functions.

4.6. Application of proposed algorithm on forecast engine

To effectively train of ENN and RNN, its generalization capability should be carefully monitored along the training process to avoid overfitting, which is a serious problem for training. By occurring the overfitting/underfitting in training phase, the prediction error will be increased while it means the prediction accuracy reduction. But, as the prediction error isn't accessible in the training stage, the error of validation sample is considered for its estimation. The obtained error from validation samples can be considered as prediction of ENN error for hidden prediction sample i.e., load signal. This criterion is an appropriate metric tool for testing the ENN/RNN ability in overfitting problem.

To improve the efficiency of test error, its samples must be same as prediction samples where in this state the test error can provide close estimate of the prediction error. By this model, the SSO trained the ENN and RNN in proposed forecast engine structure. The proposed SSO mechanism is considered as an optimization problem where the error of the constructed training samples or training error is considered as objective function. And the weights and neurons are considered as decision variables by the following steps:

- First, the SSO population is initialized randomly and the initial value for decision variables will be considered.
- In each iteration, the decision variables change based on training error improvement.
- In final iteration, the evaluation process of test error will done.
- Once, the test error began to rise, the generality performance of forecast engine starts to reduce (the overfitting/underfitting began to effect the training process) so, the training should be transferred to the matching iteration.

Furthermore, the weights of forecast engine will be set based on decision variables due to best individual of the optimization process of SSO. So, the trained forecast engine can forecast the future values of load for the next horizon.

5. Numerical results and analysis

The proposed prediction model was applied to real-world engineering test cases from the Australian energy market commission (AEMC), Pennsylvania-New Jersey-Maryland (PJM), and North American (NA) electric utilities. The AEMC and PJM datasets consisted of historical load and temperature data, and the NA dataset consisted of historical load, temperature, and humidity data. To ensure a fair comparison, the same forecast step and horizon were considered in all test cases.

5.1. Australian energy market commission

The AEMC data was detailed in Ref. [37]. The proposed prediction strategy for day-ahead forecasting was compared with artificial neural network (ANN), hybrid, and additive models taken from Refs. [38,39]. The mean absolute percentage error (MAPE) and mean absolute error (MAE) forecast error criteria were considered,

$$MAPE(\%) = \frac{1}{NH}\sum_{t=1}^{NH}\frac{\left|L_{act}(t) - L_{for}(t)\right|}{L_{act}(t)} \times 100 \tag{35}$$

and

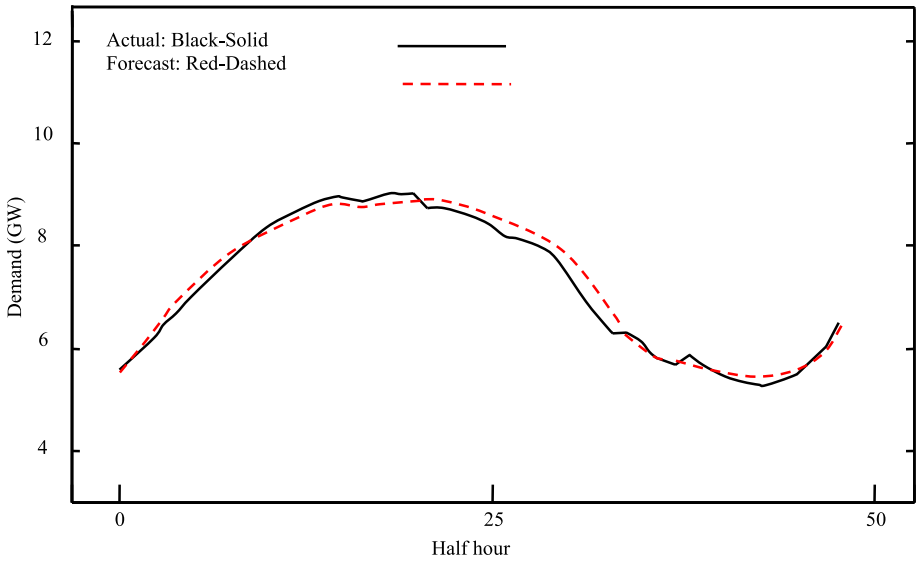


Fig. 5. Proposed method prediction and actual load for January 18, 2009, AEMC test case.

$$MAE = \frac{1}{NH}\sum_{t=1}^{NH}\left|L_{act}(t) - L_{for}(t)\right|, \quad (36)$$

where $L_{act}(t)$ and $L_{for}(t)$ are the actual and forecasted signals, respectively, for time period t; and NH is the number of time periods. In this test case, the time period was 0.5 h.

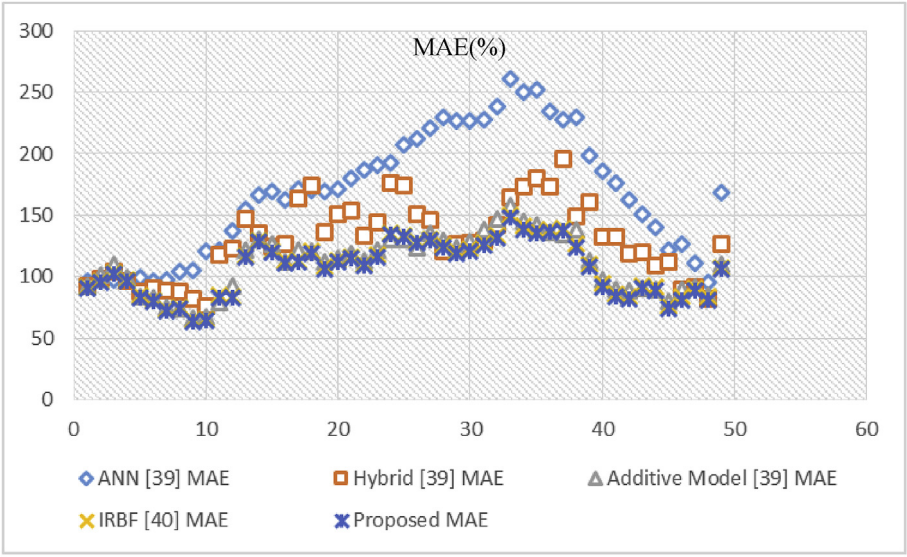
Fig. 5 compares the proposed method prediction and actual loads for a sample day. The proposed method provides better MAE and MAPE than the four comparison methods, and follows the actual signal well for each time period (Fig. 5). Also, Fig. 6, shows the results for January 2009 day-ahead load prediction.

Furthermore, Fig. 6 provides the scatter view for MAE and 2-D area comparison for MAPE. As shown in this figure, all points of proposed method in scatter figure (Fig. 6-a) is lower than all other compared methods in all times and the error area in Fig. 6-b is smaller than all other methods. This comparison demonstrates the validity of proposed method in comparison with other four prediction models.

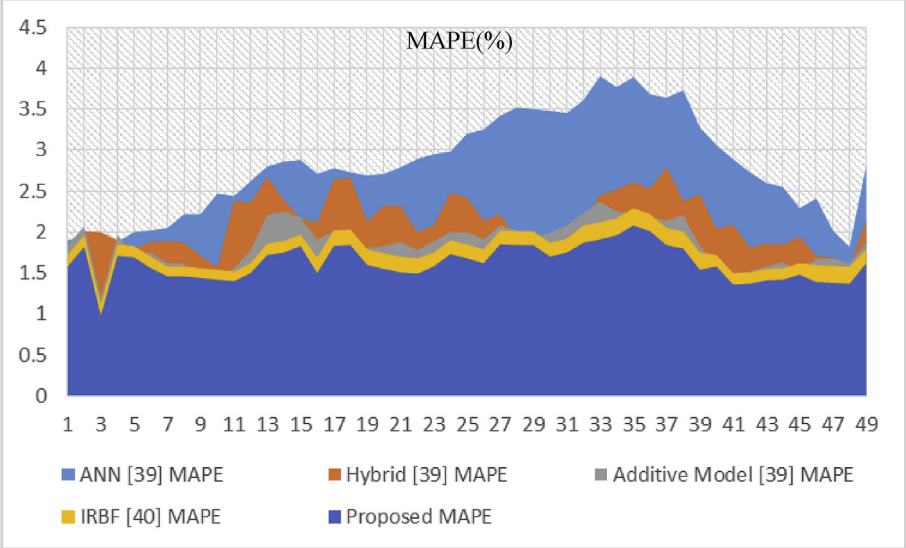
5.1.1. Pennsylvania-new Jersey-Maryland

The PJM data was detailed in Refs. [40,41]. The proposed prediction strategy for day ahead forecasting was compared with other well-known strategies: RBF neural network; multi-variate ARMA time series; MLP neural network with Broyden, Fletcher, Goldfarb, Bayesian regularization (BR), Shanno (BFGS), and Levenberg-Marquardt (LM) learning algorithms; and ANN forecasting engine trained by modified harmony search algorithm (MHSA) [42] and improved RBF (IRFB) [39]. To ensure a fair comparison, the same conditions were considered for all methods, and the forecast periods were the third week of February, May, August, and November of 2009.

Table 1 shows the different model results, furthermore, the bar comparison of the four test weeks of 2009 based on PIM test case is



(a)



(b)

Fig. 6. The graphical comparison; (a): MAE(%), (b): MAPE(%).

Table 1
Prediction error metrics for the four test weeks of 2009, PJM data based on MAPE error.

Test Week	Multi-Variate ARMA [43]	RBF [42]	MLP + BR [42]	MLP + BFGS [42]	MLP + LM [42]	ANN + HAS [42]	IRBF [39]	Proposed
Feb.	4.10	2.38	2.55	2.76	2.17	1.44	1.51	1.42
May	4.45	3.06	2.63	2.85	2.44	1.91	1.72	1.65
Aug.	3.09	2.01	2.61	2.59	1.86	1.10	1.12	1.04
Nov.	3.01	2.24	2.28	2.63	1.75	1.13	1.12	1.09
Ave.	3.66	2.42	2.52	2.71	2.05	1.39	1.37	**1.30**

presented in Fig. 7. This figure is graphical comparison of Table 1 which demonstrate that the proposed method provide better results in comparison with other methods. And Fig. 8 compares a particular prediction case for this test case. The proposed prediction method is significantly superior to all the other compared models (Table 1), and has excellent accuracy for each half-hour prediction period.

Table 2 shows the comparison of the proposed prediction model on PJM data over a long period (12 months) with ISO, ANN [42], and IRBF [39] models. In this table, the proposed method without implementation of feature selection, without proposed optimization and without both optimization and feature selection is presented. This comparison is done to show the improvement of proposed prediction tool through the proposed model step by step. So, we can see the improvement of prediction process in each step and analysis the effects of each prediction tool. For instance, in whole year, the improvement of MAPE error by implementation of feature selection is 14%, i.e., (1.53−1.32)/1.53 = 0.14. this

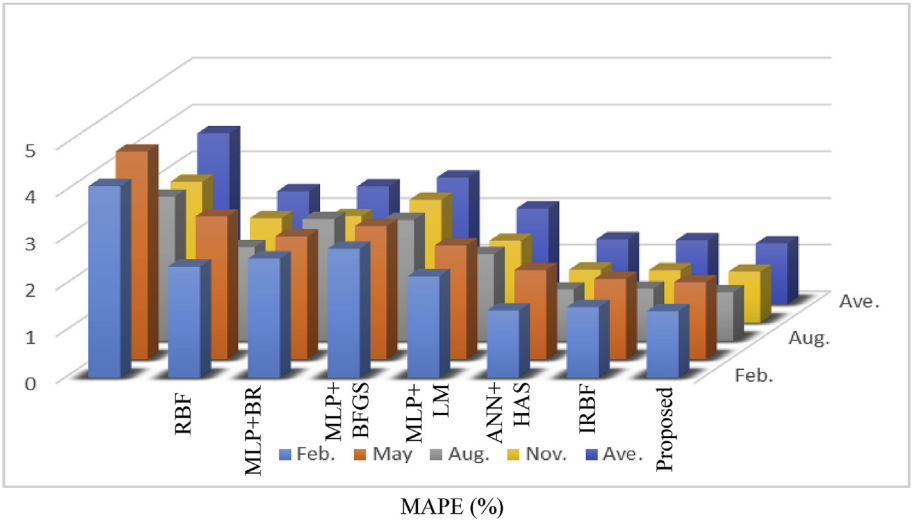


Fig. 7. The bar comparison of MAPE error in four test weeks of 2009, PJM data in Table 1.

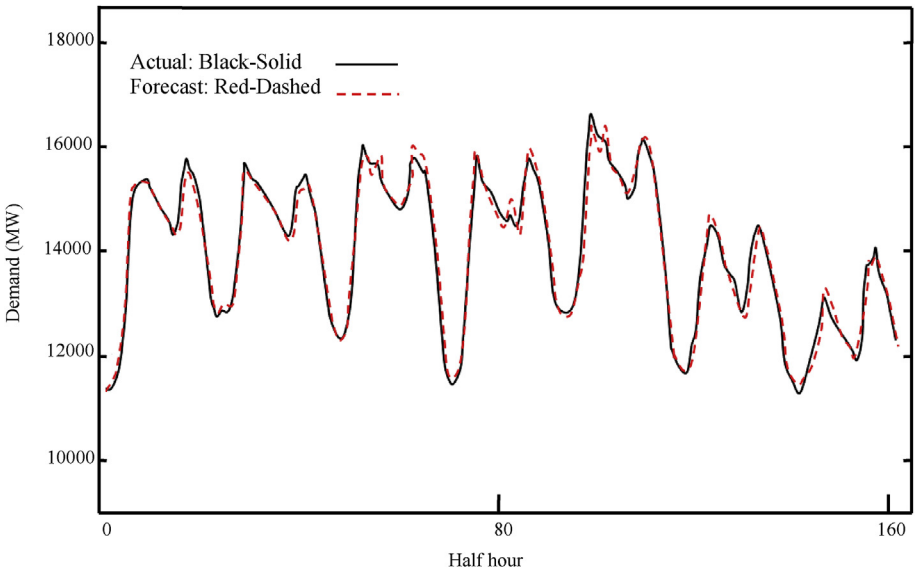


Fig. 8. Curves of actual and forecast values obtained by the proposed method prediction and actual load for PJM.

Table 2
Prediction error metrics for (MAPE) for 12 months of 2009, PJM data.

Test Month	PJM ISO [42]	ANN + HSA [42]	IRBF [39]	Proposed without optimization and feature selection	Proposed without feature selection	Proposed without optimization	Proposed
Jan.	5.25	1.67	1.52	1.68	1.51	1.36	1.28
Feb.	3.73	1.18	1.22	1.23	1.17	1.09	0.98
Mar.	4.67	1.43	1.37	1.42	1.35	1.21	1.13
Apr.	4.27	1.32	1.30	1.34	1.28	1.19	1.06
May	4.95	1.62	1.58	1.68	1.55	1.44	1.34
Jun.	5.44	1.73	1.67	1.82	1.63	1.51	1.43
Jul.	4.92	1.53	1.56	1.67	1.50	1.43	1.32
Aug.	4.29	1.35	1.43	1.46	1.32	1.22	1.19
Sep.	4.72	1.54	1.46	1.63	1.38	1.30	1.22
Oct.	4.36	1.33	1.47	1.52	1.30	1.26	1.23
Nov.	4.28	1.32	1.35	1.47	1.29	1.21	1.11
Dec.	4.71	1.53	1.40	1.57	1.37	1.24	1.16
Whole year	4.64	1.46	1.44	1.53	1.40	1.32	**1.20**

Table 3
Average DMAPE (%) for the STLF of the PJM electricity market.

Test Week	1	2	3	4	Mean
NSA	1.69	1.43	1.74	1.57	1.61
PCA	1.6	1.29	1.66	1.58	1.53
CA	1.45	1.27	1.58	1.40	1.43
MI	1.34	1.24	1.63	1.40	1.40
MR	1.28	1.19	1.53	1.44	1.36
CA + CA	1.33	1.24	1.49	1.39	1.36
Proposed	1.23	1.10	1.18	1.21	1.10

improvement for optimization algorithm and its effects on training mechanist is 8.5%, i.e., (1.53−1.40)/1.53 = 0.085. The proposed model has high accuracy and is superior to the comparison current methods for long period load prediction.

In this section, an additional analysis is presented to show the proposed feature selection method. In this section the proposed feature selection method is compared with numerical sensitivity analysis (NSA), principal component analysis (PCA), correlation analysis (CA), mutual information (MI), modified relief (MR), and CA + CA [26]. These models have been used several times on load and price forecasting problems. All feature selection techniques of Table 3 have the same forecaster and the same set of candidate inputs and historical data for the sake of a fair comparison. Obtained results are based on simple forecasting engine of multi-layer perceptron (MLP) neural network with Levenberg-Marquardt (LM) learning algorithm. In this process 400 candidate inputs are considered from the PJM through the daily mean absolute percentage error (DMAPE). Obtained results in this table proofs the validity of proposed method. The DMAPE is defined as: $DMAPE = \frac{1}{N}\sum_{i=1}^{N}\frac{|S_{iACT}-S_{iFOR}|}{S_{iACT}}$, where S_{iACT} and S_{iFOR} indicate the actual and forecast values for the ith hour of the time series.

5.1.2. North American

The North America data covered January 1, 1988 to October 12, 1992, and was detailed in Ref. [43]. As with the previous test cases, we considered the same prediction conditions for all compared methods. We first compared the proposed model for 1-h-ahead and one-day-ahead predictions with eight current prediction techniques: two stage MLP neural network (MLPNN) [44], combination of wavelet transform (WT) with MLPNN [44], combination of WT with MLPNN and evolutionary search (ES) [45], IRBF [39], echo state network (ESN) [46], and wavelet echo state network (WESN) [47].

Table 4 shows the MAPE forecasting error for the considered models. The proposed method is superior to all comparison methods. Furthermore, same as Table 2, we added the proposed method improvement in each step to this table. By this way, we can see the improvement of proposed prediction model in each step and the forecasting tool effects in this process. Furthermore, the graphical view of proposed method over 12 August of 1990 in comparison with proposed without optimization and feature selection, proposed without feature selection, and proposed without optimization is presented in Fig. 9. In this figure the comparison of each prediction method by their errors have been presented separately where, the proposed method's fitting with low error, demonstrate its high accuracy than the other models.

We then compared the proposed model for load prediction against the multiple regression model called (EGRV) [48], NN with extended Bayesian (EB) [49], NN with Level 2 support vector machine (L2-SVM) [49], local support vector regression (LSVR) [50],

Table 4
Prediction error metrics (MAPE) for the North American data.

Model	One-hour ahead	One-day ahead	Time (s) for one-day ahead
MLPNN 1 [44]	2.10	3.58	–
MLPNN 2 [44]	1.10	3.41	–
WT + MLPNN 1 [44]	1.12	3.16	–
WT + MLPNN 2 [44]	1.99	2.64	–
WT + MLPNN + ES [45]	0.99	2.04	–
ESN [46]	1.0480	2.1174	–
WESN [47]	0.6826	1.7089	–
IRBF [39]	0.67	1.58	–
Proposed without optimization and feature selection	0.70	1.72	108
Proposed without feature selection	0.67	1.63	306
Proposed without optimization	0.63	1.50	132
Proposed	0.57	1.43	436

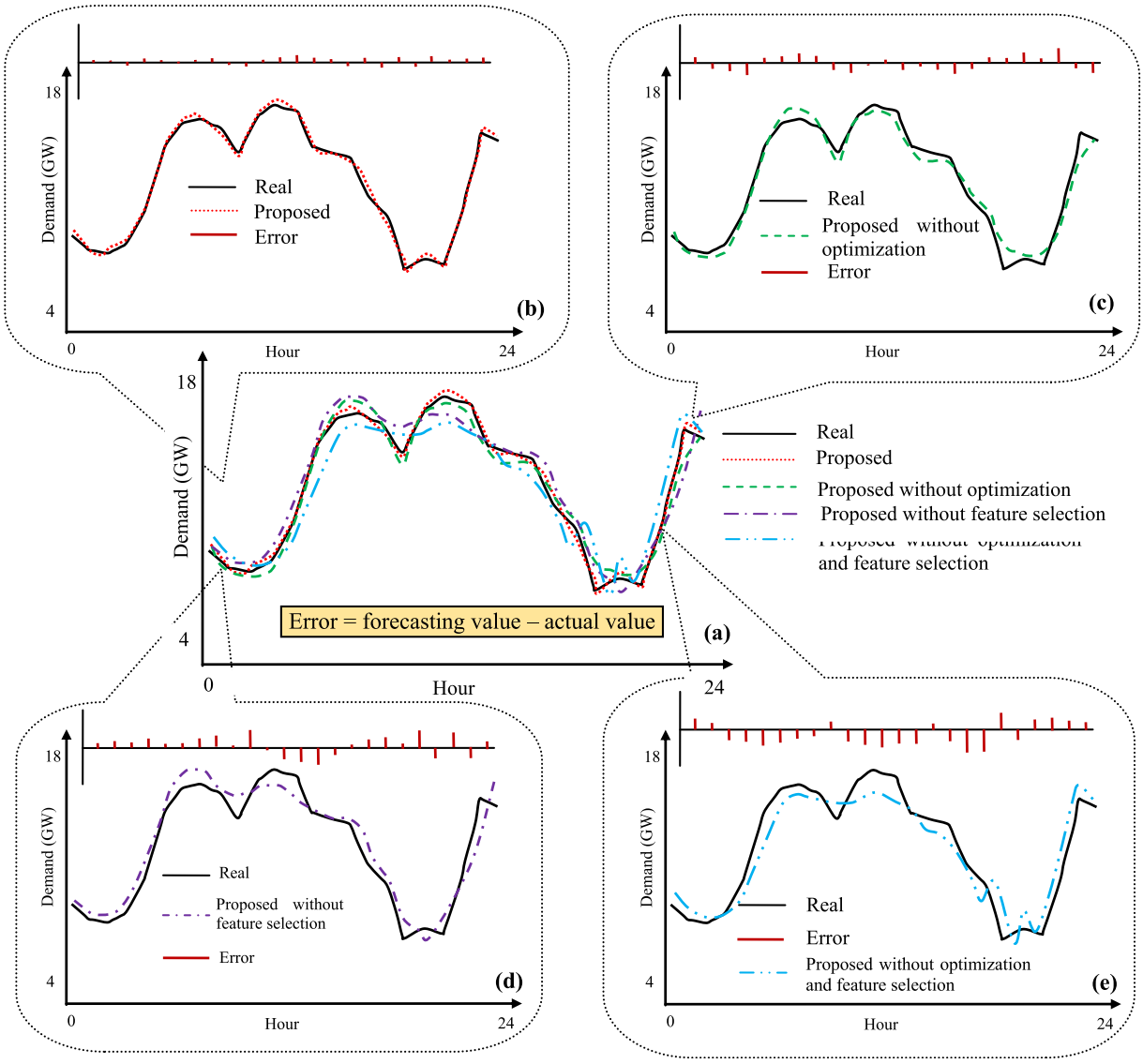


Fig. 9. The forecasting results and actual values for August 12 in comparison with different version of proposed model: (a) real signal with all models, (b) comparison of proposed method by real signal, (c) comparison of proposed method without optimization by real signal, (d) comparison of proposed without feature selection by real signal, (e) comparison of proposed without optimization and feature selection by real signal.

locally weighted support vector regression (LWSVR) [50], locally weighted regression (LWR) [50], WESN [49], IRBF [39], as shown in Table 5.

Finally, we compared the proposed method over two years data with six current methods: auto-regressive moving average with exogenous variable (ARMAX) [51], adaptive neuro-fuzzy inference system (ANFIS) [52], wavelet fuzzy NN (WFNN) [52], ESN [46], WESN [47] and IRBF [39], as shown in Table 6.

Tables 5 and 6 compare the proposed method to a total of 14 current prediction methods over north American test case and long running time, and show the proposed model was superior to all current forecast models.

For the last results, an additional analysis of the proposed structure is presented. This section answers our question for choosing the proposed structure based on RNN and ENN. Accordingly, we will re-examine the results of Table 6 through the comparison with different scenarios as following:

a) Application of two neural networks: In this case, the first neural network is used as the preprocessor block, and the second as the main predictor.
(b) Neural network plus RNN: This scenario involves the conventional neural network as a preprocessor and the ridgelet neural network as the main predictor.

Table 5
Prediction error metrics for North American data.

Error	EGRV [48]	NN + L2-SVM [49]	NN + EB [49]	LWR [50]	LSVR [50]	LWSVR [50]	WESN [47]	IRBF [39]	Proposed
MAPE	4.73	4.88	4.89	4.71	4.08	3.62	3.27	3.12	2.87
MAE	–	–	–	139.11	121.84	101.02	89.56	86.42	83.54

Table 6
Prediction error metrics for the final two years of North American data.

Days	Error	ARMAX [51]	WFNN [52]	ANFIS [52]	ESN [46]	WESN [47]	IRBF [39]	Proposed
Monday	MAPE	1.62	1.46	3.08	1.18	0.74	0.79	0.71
	MAE	36.88	33.40	69.59	26.22	16.35	15.93	14.54
Tuesday	MAPE	1.57	1.36	2.80	1.10	0.69	0.71	0.65
	MAE	35.47	31.56	63.86	24.69	15.01	14.25	13.54
Wednesday	MAPE	1.57	1.41	2.70	1.07	0.65	0.61	0.54
	MAE	35.31	32.36	61.68	24.13	14.54	14.27	13.52
Thursday	MAPE	1.65	1.44	3.03	1.16	0.68	0.62	0.54
	MAE	37.23	33.78	68.57	25.90	15.01	14.73	13.65
Friday	MAPE	1.60	1.46	3.16	1.15	0.72	0.67	0.56
	MAE	36.06	33.44	71.10	25.63	15.80	14.94	14.04
Saturday	MAPE	1.64	1.46	2.90	1.10	0.67	0.61	0.57
	MAE	36.90	33.17	64.87	24.50	14.81	14.26	12.42
Sunday	MAPE	1.58	1.43	2.87	1.20	0.73	0.70	0.63
	MAE	35.53	32.17	64.35	26.44	15.86	15.03	14.14
Average	MAPE	1.60	1.43	2.93	1.14	0.70	0.67	0.6
	MAE	36.20	32.70	66.29	25.36	15.34	14.77	13.69

(c) Neural network plus ENN: This scenario involves the conventional neural network as a preprocessor and the neural network of the element as the main predictor.
d) RNN plus NN: This scenario includes the RNN as a preprocessor and the NN as the main predictor.
e) ENN plus NN: This scenario includes the ENN as a preprocessor and the NN as the main predictor.
f) NN plus RNN: This scenario includes the NN as the preprocessor and the RNN as the main predictor.
g) proposed structure.

As the mentioned obtained results in Table 7, shows the efficiency of each model in prediction process through two different errors. As shown in this table, the proposed structure could provide better results in all week days and average values as well.

6. Conclusions

This paper proposes a new two stage forecast engine based on RNN and ENN, with a novel feature selection technique stage providing the input signal for the forecast engine. The first block consists of pre-forecasting, and ENN predicts the RNN output. Both blocks were improved using a new intelligent algorithm.

Optimization analysis verified the proposed intelligent algorithm, and the overall effectiveness of the proposed model was validated using three real test case datasets, ensuring fair comparison by applying the same prediction conditions for all methods for a given case. In summary, the proposed system was verified using 36 prediction methods over three real-world power system datasets. Although some of these methods have been duplicated in different tables, this comparison is done in different test cases and forecasting horizons.

Furthermore, to show the effectiveness of proposed feature selection as well as optimization algorithm effect, some separate numerical analysis in second and third test cases are presented. Obtained results demonstrate the validity of proposed feature selection as well as optimization algorithm in proposed prediction process improvement. In other words, the improvement of proposed prediction process is analyzed step by step, to show each sections efficiency. Finally, the proposed method could outperform the presented prediction models in this paper, over short and long-term data prediction of load signal.

Table 7
Proposed structure analysis.

Days	Error	a	b	c	d	e	f	g
Monday	MAPE	2.15	1.76	1.82	1.68	0.99	1.08	0.71
	MAE	16.77	15.59	15.65	15.51	14.82	14.91	14.54
Tuesday	MAPE	2.09	1.65	1.65	1.62	1.02	1.02	0.65
	MAE	15.43	14.59	13.09	14.51	13.82	13.91	13.54
Wednesday	MAPE	1.99	1.59	1.72	1.48	0.89	0.91	0.54
	MAE	14.96	15.33	14.63	14.77	14.12	13.89	13.52
Thursday	MAPE	2.02	1.59	1.65	1.51	0.82	0.91	0.54
	MAE	15.09	14.66	14.76	14.62	13.93	14.02	13.65
Friday	MAPE	2.08	1.61	1.58	1.34	0.84	0.93	0.56
	MAE	15.48	15.09	15.15	15.01	14.32	14.41	14.04
Saturday	MAPE	2.22	1.62	1.68	1.54	0.85	0.91	0.57
	MAE	13.86	13.47	13.53	13.39	12.76	12.79	12.42
Sunday	MAPE	2.07	1.72	1.74	1.54	0.93	1.10	0.63
	MAE	15.63	15.19	15.25	15.11	14.42	14.51	14.14
Average	MAPE	2.09	1.65	1.69	1.53	0.90	0.98	0.6
	MAE	15.32	14.84	14.58	14.70	14.03	14.06	13.69

References

[1] Shahidehpour M, Yamin H, Li Z. Market operations in electric power systems. New York, NY, USA: Wiley; 2002.
[2] Chaturvedi DK. Soft Computing: applications to electrical engineering problem. Berlin: Springer Verlag; 2008.
[3] Ghadimi Noradin, et al. A new prediction model based on multi-block forecast engine in smart grid. Journal of Ambient Intelligence and Humanized Computing 2017:1–16.
[4] Liu Yang, Wang Wei, Ghadimi Noradin. Electricity load forecasting by an improved forecast engine for building level consumers. Energy 2017;139:18–30.
[5] Zhang M, Bao H, Yan L, Cao J, Du J. Research on processing of short-term historical data of daily load based on Kalman filter. Power Syst Technol 2003:10–9.
[6] Wei L, Zhen-gang Z. Based on time sequence of ARIMA model in the application of short-term electricity load forecasting. In: Research challenges in computer science, 2009. ICRCCS'09. International conference on. IEEE; 2009. p. 11–4.
[7] Meslier F. New advances in short term load forecasting using Box and Jenkins approach. In: Proceedings; 1978. p. 51–5.
[8] Akbary Paria, et al. Extracting appropriate nodal marginal prices for all types of committed reserve. Comput Econ 2017:1–26.
[9] Gheydi Milad, Nouri Alireza, Ghadimi Noradin. Planning in microgrids with conservation of voltage reduction. IEEE Syst J 2016. https://doi.org/10.1109/JSYST.2016.2633512.
[10] Ghadimi Noradin, et al. Application of new hybrid forecast engine with feature selection algorithm in power system. Int J Ambient Energy 2017:1–13. just-accepted.
[11] Morsali Roozbeh, et al. A new multiobjective procedure for solving nonconvex environmental/economic power dispatch. Complexity 2014;20(2):47–62.
[12] Sharifi Shokoufeh, et al. Environmental economic dispatch using improved artificial bee colony algorithm. Evolving Systems 2017:1–10.
[13] Unsihuay-Vila C, Zambroni de Souza AC, Marangon-Lima JW, Balestrassi PP. Electricity demand and spot price forecasting using evolutionary computation combined with chaotic non-linear dynamic model. Int J Electr Power Energy Syst 2010;32:108–16.
[14] Gollou Abbas Rahimi, Ghadimi Noradin. A new feature selection and hybrid forecast engine for day-ahead price forecasting of electricity markets. J Intell

Fuzzy Syst Preprint 2017;32(6):4031–45.
[15] Jalili Aref, Ghadimi Noradin. Hybrid harmony search algorithm and fuzzy mechanism for solving congestion management problem in an electricity market. Complexity 2016;21(S1):90–8.
[16] Sousa JC, Neves LP, Jorge HM. Assessing the relevance of load profiling information in electrical load forecasting based on neural network models. Int J Electr Power Energy Syst 2012;40:85–93.
[17] Xia C, Wang J, McMenemy K. Short, medium and long-term load forecasting model and virtual load forecaster based on radial basis function neural networks. Int J Electr Power Energy Syst 2010;32:743–50.
[18] Mohammadi Mohsen, et al. Small-scale building load forecast based on hybrid forecast engine. Neural Process Lett 2017:1–23. https://doi.org/10.1007/s11063-017-9723-2.
[19] Ghadimi Noradin, Hosseini Firouz Mansour. Short-term management of hydro-power systems based on uncertainty model in electricity markets. J Power Technol 2015;95(4):265.
[20] Hong W-C. Electric load forecasting by seasonal recurrent SVR (support vector regression) with chaotic artificial bee colony algorithm. Energy Sep. 2011;36(9):5568–78.
[21] Ahmadian Iraj, Abedinia Oveis, Ghadimi Noradin. Fuzzy stochastic long-term model with consideration of uncertainties for deployment of distributed energy resources using interactive honey bee mating optimization. Front Energy 2014;8(4):412–25.
[22] Ghadimi Noradin, Afkousi-Paqaleh Mohammad, Nouri Alireza. PSO based fuzzy stochastic long-term model for deployment of distributed energy resources in distribution systems with several objectives. IEEE Systems Journal 2013;7(4):786–96.
[23] Ebrahimian Homayoun, et al. The price prediction for the energy market based on a new method. Economic Research-Ekonomska Istraživanja 2018;31(1):313–37.
[24] Noruzi Alireza, et al. A new method for probabilistic assessments in power systems, combining Monte Carlo and stochastic-algebraic methods. Complexity 2015;21(2):100–10.
[25] Peng H, Long F, Ding C. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. Pattern Analysis and Machine Intelligence. IEEE Transactions on 2005;27(8):1226–38.
[26] Abedinia O, Amjady N, Zareipour H. A new feature selection technique for load and price forecast of electrical power systems. IEEE Trans Power Syst 2017;32(1):62–74.
[27] Abedinia Oveis, Amjady Nima. Short-term wind power prediction based on Hybrid Neural Network and chaotic shark smell optimization. Int J Precision Eng Manufact-Green Technol 2015;2(3):245–54.
[28] Abedinia Oveis, Amjady Nima. Short-term load forecast of electrical power system by radial basis function neural network and new stochastic search algorithm. Int Trans Electr Energy Syst 2016;26(7):1511–25.
[29] Abedinia Oveis, Naslian Morteza Dadash, Bekravi Masoud. A new stochastic search algorithm bundled honeybee mating for solving optimization problems. Neural Comput Appl 2014;25(7–8):1921–39.
[30] Yang S, Wang M, Jiao L. A linear ridgelet network. Neurocomputing 2009;73:468–77.
[31] Hush DR, Horne BG. Progress in supervised neural networks. IEEE Signal Process Mag 1993;10(1):8–39.
[32] Cheng Y-C, Qi W-M, Zhao J. A new Elman neural network and its dynamic properties. In: Proceedings of the IEEE international conference on cybernetics and intelligent systems (CIS '08); September 2008. p. 971–5.
[33] Song Q. On the weight convergence of Elman networks. IEEE Trans Neural Network 2010;21(3):463–80.
[34] Lin F-J, Teng L-T, Chu H. Modified Elman neural network controller with improved particle swarm optimization for linear synchronous motor drive. IET Electr Power Appl 2008;2(3):201–14.
[35] Abedinia O, Amjady N, Ghasemi A. A new meta-heuristic algorithm based on shark smell optimization. Complexity Journal 2014. https://doi.org/10.1002/cplx.21634.
[36] Abedinia O, Amjady N. Short-term wind power prediction based on hybrid neural network and chaotic shark smell optimization. Int J Precision Eng Manuf Green Technol (Springer) July 2015;2(3):245–54.
[37] http://www.aemo.com.au/Electricity/Data/Price-and-Demand.
[38] Fan S, Hyndman Rob J. Short-term load forecasting based on a semi-parametric additive model. IEEE Trans Power Syst February 2012;27(1):134–41.
[39] O. Abedinia, N. Amjady, Short term load forecast of electrical power system by radial basis function neural network and new stochastic search algorithm, Eur Trans Electr Power, DOI: 10.1002/etep.2160.
[40] Pennsylvania-New Jersey-Maryland system operator. Available online: http://www.pjm.com (accessed on December 1, 2010).
[41] Pennsylvania State Climatologist. Available online: http://climate.met.psu.edu (accessed on December 5, 2010).
[42] Abedinia O, Amjady N, Ghadimi N. Solar energy forecasting based on hybrid neural network and improved metaheuristic algorithm. Comput Intell 2017:1–20. https://doi.org/10.1111/coin.12145.
[43] https://www.ee.washington.edu/class/559/00_archived.
[44] AJR R, Silva AP. Feature extraction via multiresolution analysis for short-term load forecasting. IEEE Trans Power Syst February 2005;20(1):189–98.
[45] Poghosyan A. Long term individual load forecast under different electrical vehicles uptake scenarios. Appl Energy 2015;157:699–709.
[46] Deihimi A, Showkati H. Application of echo state networks in short-term load forecasting. Energy March 2012;39(1):327–40.
[47] Deihimi A, Orang O, Showkati H. Short-term electric load and temperature forecasting using wavelet echo state networks with neural reconstruction. Energy August 2013;57:382–401.
[48] Ahadi Amir, Ghadimi Noradin, Mirabbasi Davar. An analytical methodology for assessment of smart monitoring impact on future electric power distribution system reliability. Complexity 2015;21(1):99–113.
[49] Ferreira VH, da Silva APA. Toward estimation autonomous neural network-based electric load forecasters. IEEE Trans Power Syst November 2007;22:1554–62.
[50] Elattar EE, Goulermas JY, Wu QH. Electric load forecasting based on locally weighted support vector regression. IEEE Trans Syst Man Cybern C Appl Rev February 2010;40:438–47.
[51] Huang CM, Huang CJ, Wang ML. A particle swarm optimization to identifying the ARMAX model for short-term load forecasting. IEEE Trans Power Syst May 2005;20:1126–33.
[52] Hanmandlu M, Chauhan BK. Load forecasting using hybrid models. IEEE Trans Power Syst May 2011;26:20–9.