Electric Power Systems Research 211 (2022) 108351

Contents lists available at ScienceDirect

Electric Power Systems Research

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

An advanced deep neuroevolution model for probabilistic load forecasting

Seyed Mohammad Jafar Jalali [a], Parul Arora [b], B.K. Panigrahi [b], Abbas Khosravi [a], Saeid Nahavandi [a], Gerardo J. Osório [c], João P.S. Catalão [d,*]

[a] Institute for Intelligent Systems, Research and Innovation, (IISRI), Deakin University, Victoria, Australia
[b] Department of Electrical Engineering, Indian Institute of Technology, Delhi, New Delhi 110017, India
[c] Portucalense University, Infante D. Henrique (UPT), Porto, Portugal
[d] Faculty of Engineering of the University of Porto (FEUP) and INESC TEC, Porto, Portugal

ARTICLE INFO

Keywords:
Deep learning
Neuroevolution
Probabilistic load forecasting
Optimization

ABSTRACT

Probabilistic load forecasting (PLF) is necessary for power system operations and control as it assists in proper scheduling and dispatch. Moreover, PLF adequately captures the uncertainty whether that uncertainty is related to load data or the forecasting model. And there are not many PLF models, and those which exist are very complex or difficult to interpret. This paper proposes a novel neuroevolution algorithm for handling the uncertainty associated with load forecasting. In this paper, a new modified evolutionary algorithm is proposed which is used to find the optimal hyperparameters of 1D-Convolutional neural network (CNN). The probabilistic forecasts are produced by minimizing the mean scaled interval score loss function at 50%, 90% and 95% prediction intervals. The proposed neuroevolution algorithm is tested on a global energy forecasting competition (GEFCom-2014) load dataset, and two different experiments are conducted considering load only and one with load and temperature. Strong conclusions are drawn from these experiments. Also, the proposed model is compared with other benchmark models, and it has been shown that it outperforms the other models.

1. Introduction

Electricity load forecasting is vitally essential for power system operation and control. The load forecasting can be done either by deterministic/point forecasting or probabilistic forecasting. Probabilistic load forecasting [1] provides more uncertainty information, and such information is useful for power system operations and control like economic dispatch, unit commitment tasks [2] and electricity trading.

1.1. Motivation

In the recent years, many deep learning techniques [3] have been used for probabilistic load forecasting like recurrent neural networks (RNN) [4], deep belief network (DBN), convolutional neural networks (CNN) or CNN-LSTM. Convolutional neural networks are widely used to solve complex tasks like image processing, pattern recognition and energy management. Among these DNN's, CNN's automate the process of feature extraction, and they have performed very well in regression tasks [5]. Although, CNN's have offered an advanced feature selection process, but the selection of hyper-parameters of CNNs is a very cumbersome task, since it involves lots of parameters and hyper-parameters designing in convolutional layers [6]. Usually, their architecture is designed manually using trial-and-error method or with the help of an expert in that domain. These type of architecture designs have two main limitations: First, most of the effort and time is spent on tuning manual parameters. Secondly, the design proposed by the expert is for only one type of problem and not always applicable on other kind of problems and mayn't show good performance [7]. An efficient approach to train the DNN's is to combine them with the evolutionary algorithms (EA's). The evolutionary algorithms are considered as one of the most potent optimization models for evolving the architectures of several deep learning models effectively and automatically [8–10]. Such models where DNNs are combined with the EAs are known as deep neuroevolution models. Therefore, to simplify the hyperparameter selection procedure, we have integrated an evolutionary algorithm with deep CNNs [11,12]. The main advantage of using evolutionary algorithms is that they easily converge to optimal solutions and they are capable of handling complex, multi-dimensional and NPhard problems. Several evolutionary algorithms are being used for optimization problems such as differential evolution, particle swarm optimization, grasshopper optimization algorithm, flower pollination (FPA), success history-based adaptive differential evolution (SHADE) [13] and many

* Corresponding author.
E-mail addresses: sjalali@deakin.edu.au (S.M.J. Jalali), parularora@ee.iitd.ac.in (P. Arora), bkpanigrahi@ee.iitd.ac.in (B.K. Panigrahi), abbas.khosravi@deakin.edu.au (A. Khosravi), saeid.nahavandi@deakin.edu.au (S. Nahavandi), gerardo@upt.pt (G.J. Osório), catalao@fe.up.pt (J.P.S. Catalão).

https://doi.org/10.1016/j.epsr.2022.108351
Received 4 October 2021; Received in revised form 9 March 2022; Accepted 2 July 2022
Available online 13 July 2022

others. In this paper, we have focussed on FPA due to its simplicity and effectiveness for solving multi-dimensional problems in the real-world. Therefore, developing improved FPA and its integration with CNN for developing a novel deep neuroevolution model for probabilistic load forecasting problem is the main motivation behind this paper.

1.2. Key contributions

In this article, for the first time in the literature, we develop an advanced deep neuroevolution strategy based on an improved version of the flower pollination algorithm (FPA) named IFPA and the deep-learning-based convolutional neural networks (CNN) to solve the challenging probabilistic load forecasting problem. We incorporate two powerful evolutionary operators into the standard version of FPA to enhance the exploration and exploitation phases and improve the convergence speed to effectively find the best solutions for the probabilistic load forecasting problem.

This model tunes the architecture of deep CNNs automatically and improves its forecasting performance. The CNN produces probabilistic forecasts by optimizing the mean scaled interval score by the FPA algorithm. GEFCom-2014 load dataset is used for testing the performance of the proposed neuro-evolution model. Two different experiments are conducted, one with load and temperature and the other considering loads only. It has been observed that the model performs better in case of considering more variables i.e. load and temperature. Also, the experimental findings represent that our proposed model outperforms an excellent probabilistic forecasting accuracy compared to the other conventional state of the art algorithms.

The main key contributions of the work are summarized as below:

- An advanced deep neuroevolution model based on the improved flower pollination algorithm and CNN is proposed for the probabilistic load forecasting problem.
- Standard FPA is enhanced with the introduction of the new evolutionary operators and these enhances the convergence speed of the algorithm.
- The proposed algorithm is capable of optimizing the parameters of the deep neural network (CNN) automatically without the manual (hit-and-trial) intervention.
- The mean scaled interval score has been used as the fitness function for the evolutionary algorithm, which considers both the coverage and width of the prediction intervals.
- Evaluation of the need for exogenous variable like temperature in the case of probabilistic load forecasting

1.3. Related work

The application of CNNs in the field of time-series forecasting is limited to the best of our knowledge. In [14], Binkowski et al. have used CNN-AutoRegressive model for the electricity and financial time series data and they have shown that their model outperforms the CNN and LSTM models. Fuzzy time series and Convolutional neural network has been used for short-term load forecasting in [15]. The dilated convolutional neural network is integrated along with bidirectional long short-term memory for power supply control between consumer and supplier in [16]. Error feedback stochastic modelling (ESM) strategy has been used for constructing convolutional neural network (CNN) in [17] for time series forecasting task. This ESM strategy adds neurons and random filters incrementally to compensate for the error during the processing stage. The filter selection strategy extracts the different size temporal features at each stage. A non-linear relationship extraction method is proposed in [18]. CNN is used to extract the non-linear relationship among the past, present and future load values, and then support vector regression is used for load forecasting. Recurrent inception convolutional neural network, which is the combination of RNN and CNN, has been proposed in [19]. Here, CNN is used to calibrate the prediction time and hidden state vector values calculated from RNN and this model is tested on power usage data of South Korea. Zhang et al. [20] have proposed a CNN-Seq2Seq model with an attention mechanism for electric load forecasting, where CNN is used to extract input features, and multi-energy load forecasting is done by the Seq2Seq network. Further attention mechanisms and multi-task learning are used for the accuracy improvement of the model. In [21], a hybrid forecasting method is developed by combining CNN and K-means clustering. The large dataset obtained from the power grid is clustered into small subsets, and these subsets are used for training CNN. FPA is being used for wind forecasting [22], electrical load forecasting [23], energy flow management in microgrid [24], economic dispatch [25,26], price prediction in electricity trading markets [27]. A multi-objective flower pollination algorithm is developed in [23], in which FPA is used to optimize the weights of single models obtained for electrical load forecasting. FPA is used along with differential evolution (DE) algorithm to prevent premature convergence, and it is used along with fuzzy inference systems for electrical load dispatch. FPA, along with machine learning algorithms for unit price prediction, is proposed in [27]. Very few works have used FPA optimization algorithms and machine learning algorithms for electrical load forecasting.

1.4. Organization

In this paper, Section 2 explains the enhanced flower pollination algorithm, convolutional neural network model, and evaluation metrics used to assess the performance of the models. Section 3 presents the experimental setup and the dataset used. Section 4 explains the experimental details, and the paper concludes with the conclusion in Section 5.

2. Deep neuroevolution-based probabilistic model

This section explains how to use our proposed deep learning-based optimization model for a probabilistic forecasting model. Here are the steps of our proposed model:

2.1. Enhanced flower pollination algorithm

Flower pollination algorithm is originated from the concept of flower pollination in nature [28]. The pollination process is categorized into biotic pollination and abiotic pollination. In biotic pollination, pollen is spread by animals or insects, and it corresponds to global search, whereas abiotic pollination corresponds to pollination by wind and other natural factors, and it is considered the local search.

Four rules were considered for FPA as mentioned below:

- *Phase 1:* Biotic pollination is a global search process in which pollen-carrying pollinators obey Lévy flight
- *Phase 2:* Abiotic pollination is a local search process
- *Phase 3:* The greater the similarity between two flowers, the greater are the chances of pollination, and this probability is known as flower constancy
- *Phase 4:* There can be a transformation between local and global pollination which is given by switch probability $p \in (0, 1)$

For building a mathematical model of FPA, it is assumed that each plant has only one flower, and each flower has only one pollen particle. The global search process is given by mathematical formula as shown below :

$$x_i^{t+1} = x_i^t + \gamma L(x_i^t - g_*) \tag{1}$$

where g_* is the current best solution and, x_i^t and x_i^{t+1} are the solution vectors of pollen i at tth iteration and $(t+1)$th iteration, respectively. The parameter L denotes step size of pollen particles as given by :

$$L \sim \frac{\lambda \Gamma(\lambda) \sin\left(\frac{\lambda}{2}\pi\right)}{\pi} \frac{1}{s^{1+\lambda}}, \quad (s > 0) \tag{2}$$

where $\lambda = 1.5$ and $\Gamma(\lambda)$ is a standard gamma function where s is given by:

$$s = \frac{U}{|V|^{1/\lambda}} \tag{3}$$

In Eq. (3), U is Gaussian distribution having variance δ^2 and V is a random number following standard normal distribution, as given by:

$$U \sim N(0, \delta^2), \quad V \sim N(0,1),$$
$$\delta^2 = \left\{ \frac{\Gamma(1+\lambda)}{\lambda\Gamma[(1+\lambda)/2]} \cdot \frac{\sin(\pi\lambda/2)}{2^{(\lambda-1)/2}} \right\}^{1/\lambda}. \tag{4}$$

The local search model is given by:

$$x_i^{t+1} = x_i^t + \epsilon(x_j^t - x_k^t) \tag{5}$$

where x_i^t is the solution vector of pollen i at tth iteration, x_k^t and x_j^t are pollens yielded by the same plant species but separate flowers, ϵ is a random number following uniform distribution in [0,1].

FPA works in both global search and local search domains, and we can switch between both through switch probability $p \in (0,1)$, and in various models after testing, it has been observed that 0.8 is the apt switch probability.

To enhance the searching capabilities of FPA, we apply two powerful evolutionary operators over the basic FPA as follows. We name this enhanced strategy on FPA as improved FPA (IFPA).

-First modification:

The first adjustment is to proceed with the best solution while evaluating the opposing assumptions concurrently. As a result, the fitter (estimation or opposite estimation) can be selected as the initially best alternative. Starting with the fitter of the two, estimation or opposite estimation, can provide a point of reference closest to the optimal solution. To provide the solution, the opposite vector needs to be determined.

Assume x is a current solution (flower). The elements of the opposite solution vector $\hat{x}$ are as follows:

$$\hat{x}_i = a_i + b_i - x_i, \quad i = 1, 2, \ldots, n \tag{6}$$

where the lower and upper bounds of x_i is defined by $[a_i, b_i]$. Now, it is time to define the strategy of the Quasi-oppositional vector. Such solution vector is derived from the preceding opposite point description and has been verified to outperform the standard opposite vector. Let $\hat{x} \in \mathbb{R}^n$ be its opposite solution point and $x \in \mathbb{R}^n$ be a solution point and $x_m = \frac{a+b}{2}$ is the middle point between $[\boldsymbol{a}, \boldsymbol{b}]$. Then the solution vectors $\hat{x}_Q$ for quasi-oppositional operator are defined as : if $x_i < x_{m_i}$ else

$$\hat{x}_{Q_i} = x_{m_i} + \epsilon\left(\hat{x}_i - x_{m_i}\right)$$
$$\hat{x}_{Q_i} = \hat{x}_i + \epsilon\left(x_{m_i} - \hat{x}_i\right)$$

where $\epsilon \sim [0,1]$ and $i = 1, 2, \ldots, n$

-Second modification: In this modification stage, we utilize a dynamic switching probability technique to make a better balance between the exploration and exploitation phases of FPA. Thus, we apply the different dynamic switching probability given by:

$$p = p_{\min} + \exp\left(-10 * \frac{t}{Max\_iter}\right)(p_{\max} - p_{\min}) \tag{7}$$

where $p_{\max}$, represents the upper limit which is set at 0.8, and $p_{\min}$ corresponds to the lower limit and is set at 0.2. In the initial stages of the search, p is set at a high value, meaning pollens move in the complete search space to find the optimal solution. Later on, the value of p is set at a lower value so that pollens settle down around themselves and find the more optimal solution accurately (see Fig. 1).

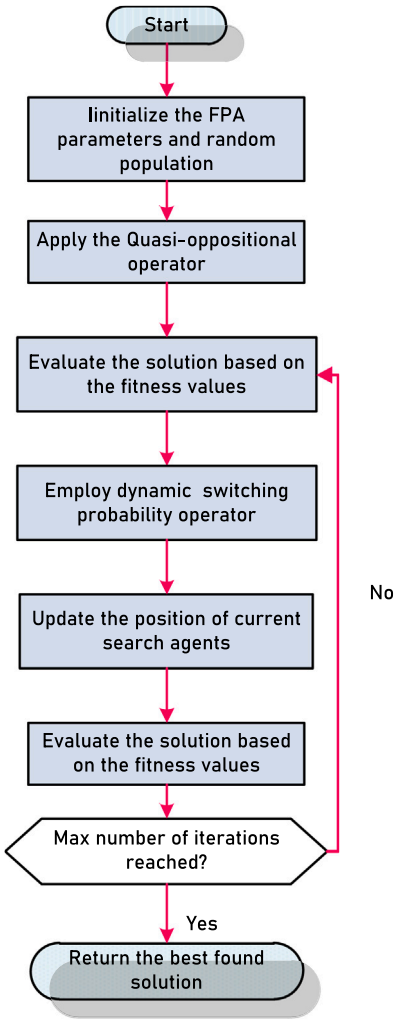

Fig. 1. The flowchart of proposed evolutionary IFPA.

2.2. 1D-Deep convolutional neural networks

1D convolutional neural networks are a type of deep learning neural network that uses feed-forward learning in their structures. Although 2D-CNNs have been widely employed for image classification applications, they are not generally appropriate for time series classification. 1D-CNNs and their variants have been successfully used in energy forecasting problems such as photovoltaic power forecasting [29], solar irradiance forecasting [12,30] and electricity load forecasting [31]. 2D-CNNs also take more time to train compared to 1D-CNN. According to [32], 2D-CNNs demand between 1M and 10M parameters to be trained, as opposed to 1D-CNNs, which typically contain approximately 10k trainable parameters.

The most significant advantage of 1D convolutional neural nets is that they automatically retrieve valuable features for 1D time-series patterns. Such a key characteristic renders them ideal for analysing time series in the manner of regression and classification. By considering a signal a with the length of n, a convolution operator with a filter f and size m is given by the following formula:

$$\text{Conv}\,1D(a, f) \quad = \left[\text{Conv}\,1D(a \mid i, f), i \in \left(1, \frac{n-m}{s} + 1\right)\right] \tag{8}$$

where $Conv1D(a[i], f) = \sum_{k=-m}^{m} a[i-k]f[k]$.

1D-CNNs are made up of several or singular convolutional layers, which are then accompanied by pooling, dropout, and flattening layers. In the context of succeeding layers, the convolution agent concatenated a filter of a certain length with the input time series or the input series of a specific layer. Let us assume the ith neuron output in layer $l-1$ is denoted by y_i^{l-1} and considers as the input of kth neuron of layer l. After this procedure, by following the convolution operator and the

implementation of the activation function $(f(.))$, the output of neuron k in layer l is calculated by the following formula:

$$y_k^l = f\left(b_k^l + \sum_{i=1}^{N_{l-1}} \text{Conv1}D\left(w_{ik}^{l-1}, y_i^{l-1}\right)\right) \tag{9}$$

The flattening layer is succeeded by applying the above operators by a fully connected layer. The final layer employs tanh or ReLU for regression applications such as time series forecasting.

2.3. Evaluation metrics

2.3.1. Prediction interval coverage probability (PICP)

The coverage of prediction intervals is assessed via Prediction Interval Coverage Probability (PICP). The quantiles $(0, 1)$ represent the coverage probability from (0 to 100%) . For any quantile α, predictions must lie within $(1-\alpha)\%$ confidence interval.

$$PICP = \frac{1}{n}\sum_{m=1}^{n} p_m \tag{10}$$

where $p_m = 1$ if mth predictions lie within upper bound U_m and lower bound L_m, else $p_m = 0$. In this paper, we have considered confidence levels at 90% and 95% for evaluating the performance of our model.

2.3.2. Skill score

For the complete assessment of probability forecasts, skill scores are used to cover both reliability and sharpness. For evaluation of prediction intervals, we have used mean scaled interval score (MSIS) as the skill score [33]. For representing quantiles at level α, lower bound is considered at $\frac{\alpha}{2}$, upper bound at $1-\frac{\alpha}{2}$ and center at (1-α) ×100%. If the predictions are outside the lower and upper bound, then MSIS imposes $\frac{2}{\alpha}$ penalty. Therefore, MSIS considers both the coverage and width of prediction intervals. In this paper, we have considered skill score as our fitness function, aiming to minimize it.

$$IS = (U_t - L_t) \qquad (L_t \le y_t \le U_t) \tag{11}$$

$$= (U_t - L_t) + \frac{2}{\alpha}(L_t - y_t) \qquad (y_t \le L_t) \tag{12}$$

$$= (U_t - L_t) + \frac{2}{\alpha}(y_t - U_t) \qquad (y_t \ge U_t) \tag{13}$$

MSIS is the mean scaled score of the IS.

$$MSIS = \frac{IS}{h(n-m)\sum_{t=m+1}^{n}|y_t - y_{t-m}|} \tag{14}$$

where m is the frequency of data, n is the number of samples, and h is the forecast horizon.

2.4. Pinball loss (PL)

Pinball loss which is twice the quantile loss evaluates the loss of the given quantile, while considering the under bias and over bias [34]. Pinball loss is represented as:

$$underbias = q(y - \hat{y}_{t,q}) \qquad (y_t \ge \hat{y}_{t,q}) \tag{15}$$

$$overbias = (1-q)(\hat{y}_{t,q} - y) \qquad (\hat{y}_{t,q} < y_t) \tag{16}$$

$$PL = (underbias + overbias) \tag{17}$$

There can be parametric or non-parametric ways to produce probabilistic forecasts. This paper has adopted a non-parametric method of obtaining forecasts through quantile regression. For producing quantile forecasts, the decoder of CNN directly outputs the set of quantile values for every time-step in the forecasting horizon and training of this model is done using the quantile loss function. The optimum hyperparameters of the CNN are obtained through FPA optimization.

Table 1
Symbols used in the optimization process of CNN network architecture.

Hyperparameters	Symbol
No. convolutional layers	N_c
Kernel size	$kS_{i,i\in N_c}$
Batch size	S_{batch}
Dropout rate	R_{dropout}

Table 2
List of range of hyperparameter values during CNN architecture design.

Expression	Valuation list
$N_c = \psi$	$\overline{N_c(\psi)}_{\psi\in[1,4]} \in [1, 2, \ldots]$
$kS_{1<i<N_c} = (2 \times \psi) + 1$	$\overline{kS(\psi)}_{\psi\in[1,4]} \in [3, 5, 7, 9]$
$R_{\text{dropout}} = (\psi + 3) \times 0.05$	$\overline{R_{\text{dropout}}(\psi)}_{\psi\in[1,8]} \in [0.2, 0.25, \ldots]$
$S_{\text{batch}} = 10 \times \psi$	$\overline{S_{\text{batch}}(\gamma)}_{\gamma E[1,10]} \in [10, 20, \ldots]$

Table 3
Statistical parameters of GEFCom-2014 data.

	Mean	Std	Min	Max
Load (MW)	145.64	46.66	48.4	315.6
Temp (F)	61.33	16.16	12.64	97.68

3. Dataset selection

The proposed neuroevolution model is tested on GEFCom-2014 dataset [35]. In this dataset, we have considered two different variables: load and temperature, and we formulate two different case studies for day-ahead probabilistic load forecasting, one while considering load only and the other while considering load and temperature both. GEFCom-2014 data is provided by [36] for load forecasting competition with hourly resolution. Table 3 presents the statistical parameters of data. Data from 1 January 2005 to 31 August 2009 is considered for training, while from 1 September 2009 to 30 September 2010 is considered for testing. We have conducted two different experiments on this dataset, experiment 1, where the temperature is taken as the exogenous variable along with load and in experiment 2, only load is considered (univariate case). To have fair comparison with all models, data is normalized in the range of (0,1). This study aims to get probabilistic load forecasts for 90% and 95% prediction interval such that forecasts lie within these intervals. In Tables 1 and 2, the list of CNN hyperparameters that are evolved in the optimization procedure is tabulated.

4. Experimental results

Electricity load displays hourly as well as daily seasonality patterns. We have considered dynamic features and time features in our case study. Dynamic features consist of the temperature variable, and time features consist of hour-of-the-day and day-of-the-week seasonality. Hour-of-the day captures peak and low loads information, and day-of-the-week stores information about weekends and weekdays. We have tested our proposed evolutionary-based deep CNN model (IFPA-CNN) on the GEFCom-2014 dataset. Our proposed probabilistic load forecasting model optimizes the generated prediction intervals (PI) performance using the Mean Scaled Interval Score (MSIS).

To examine the efficiency of our proposed model, we compare it with two deep learning models named as DeepAr and DeepState, which are the state of the art models in the literature for electricity load forecasting. Besides, to evaluate the searching capabilities of our proposed model, since we have the elements of evolutionary algorithms into that, we compare it with three powerful evolutionary algorithms, which are hybridized with the CNN model. These models are the standard version of the Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Flower Pollination Algorithm (FPA). We have initially examined these

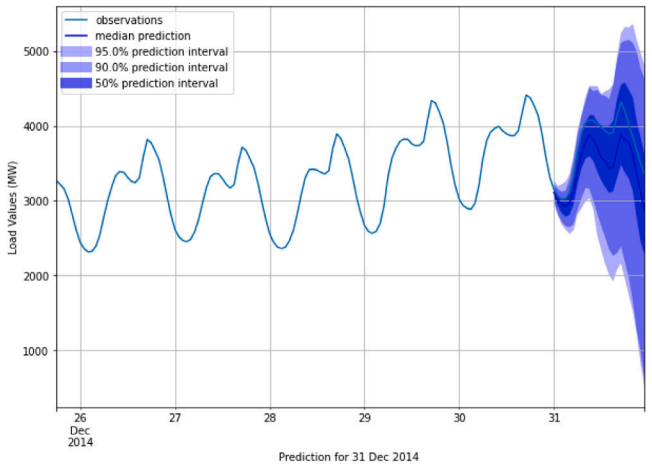

Fig. 2. The prediction intervals obtained by the proposed model for the 90% and 95% intervals in case of experiment 1.

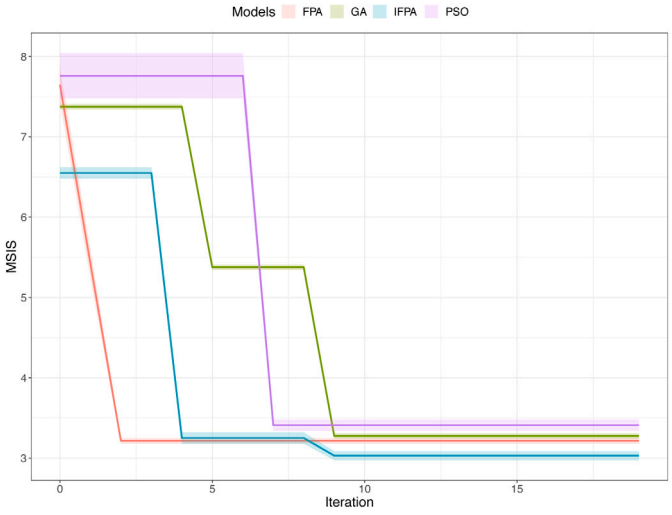

Fig. 3. The convergence profiles of different deep-neuroevolution based models based on the PSO,GA, FPA and IFPA.

models based on their reference articles and then conducted trial and error simulations are done to report their best performances. We also set the number of population equal to 10 and the number of iterations similar to 20 for all evolutionary-based deep CNN models to have a fair comparison with our proposed model. Also, all the algorithms have been performed by ten independent runs.

For evaluating the performance of our proposed model, we have conducted two experiments, as explained below:

Experiment 1: Considering load and temperature

In experiment 1, the load is considered the static feature, whereas temperature is regarded as the dynamic feature. Time features like hour-of-day and day-of-week are also considered. We have trained the model on this data and evaluated the model for the day-ahead prediction. Fig. 2 shows the prediction interval coverage at 90% and 95% prediction interval while considering both load and temperature. As we can see, the median values, which usually represents the deterministic forecasts, follow the test dataset, and all other observations lie within the 95% coverage.

The IFPA-CNN model has been compared with other benchmark models which have been employed in the field of probabilistic forecasting like DeepAr [37], DeepState [38], flower pollination algorithm (FPA) optimized CNN, genetic algorithm (GA) optimized CNN, particle swarm optimization (PSO) optimized CNN, Deep Ensemble [39] and quantile regression neural network with skip connections(QLNN) [40].

Table 4
Comparison of the proposed model with other models in terms of PICP and MSIS scores.

Model	PICP (0.9)	PICP (0.95)	MSIS	PL
Proposed (IFPA-CNN)	0.921	0.962	2.99	200.14
DeepAr	0.908	0.949	3.95	263.98
DeepState	0.891	0.932	4.21	272.45
FPA-CNN	0.913	0.955	3.21	285.19
GA-CNN	0.918	0.952	3.32	246.15
PSO-CNN	0.898	0.945	3.46	232.16
Deep ensemble	0.891	0.949	3.25	253.78
QLNN	0.902	0.958	3.88	225.18

Table 5
Comparison of the proposed model with other models in terms of training, testing and optimization time (seconds).

Model	Training time (s)	Testing time (s)	Optimization time (s)
Proposed (IFPA-CNN)	62.3	35.7	271.5
DeepAr	68.9	37.2	–
DeepState	78.4	42.6	–
FPA-CNN	66.8	39.8	294.4
GA-CNN	65.3	38.6	279.5
PSO-CNN	67.9	37.8	289.2
Deep ensemble	64.3	36.98	–
QLNN	63.9	36.12	–

For proper coverage of prediction intervals, the value of PICP must be equal to or higher than the α. As we can see from the Table 4, DeepAr, FPA-CNN, GA-CNN and QLNN have good coverage probabilities (PICP), but the IFPA-CNN outperforms the others. In the case of MSIS, the proposed model has a minimum skill score of 2.99, followed by FPA-CNN and others. Also, in terms of pinball loss the proposed model has minimum quantile loss (200.14) followed by QLNN, PSO-CNN, GA-CNN, Deep Ensemble, DeepAr and DeepState.

We can see from Fig. 3, that convergence of IFPA is very fast as compared to the GA, FPA and PSO. This finding shows that by incorporating two powerful evolutionary operators into the basic version of FPA, the proposed IFPA can have an excellent performance in finding the best solution to solve the probabilistic load forecasting problem (see Table 5).

Experiment 2: Considering only load

In experiment 2, the only load is considered, and the proposed neuroevolution algorithm only looks at the load values and learns features and patterns from load data. In this experiment, hour-of-the-day and day-of-the-week are considered, whereas temperature is not considered. 1D-CNN extracts the parameters from these features and gives the probabilistic forecasts by minimizing the quantile loss function. The coverage probability, in this case, is represented by Fig. 4.

If we compare Figs. 2 and 4, we can say that while considering both load and temperature variables, better coverage and minimum MSIS is obtained in comparison to considering load only. Thus, two major conclusions are drawn from these experiments:

- The proposed neuroevolution algorithm (IFPA-CNN) outperform other benchmark algorithms.
- Better probabilistic forecasts are given in the presence of load and temperature, whereas only load data cannot capture all the information present in the seasonal data.

Thus, for single loads, both load and temperature are must be required for good predictions and this fact is also highlighted in most of the load forecasting literature [35].

5. Conclusion

This paper proposes a novel neuro-evolution algorithm that integrates the flower pollination algorithm (FPA) with a convolutional

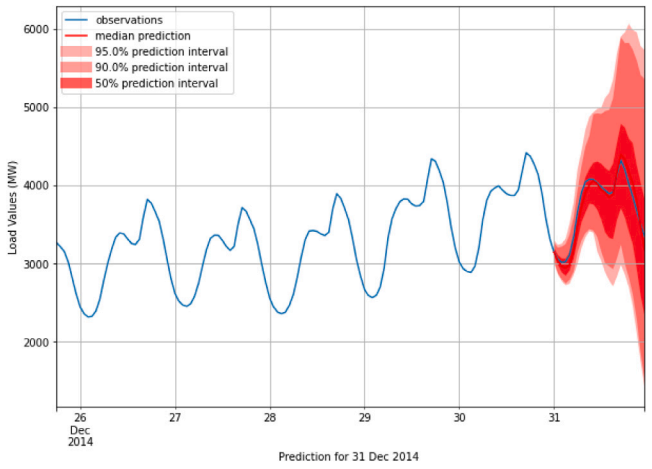

Fig. 4. The prediction intervals obtained by the proposed model for the 90% and 95% intervals in case of experiment 2.

neural network (CNN). CNN's are good at extracting high-level features from the data, but selecting their parameters is cumbersome. This paper has modified the flower pollination algorithm and named it improved FPA (IFPA). The proposed model is tested on the GEFCom-2014 dataset. Two different experiments are conducted, one considering both load and temperature and the other considering load only. The 1D-CNN produces the probabilistic forecasts, and the IFPA does parameter optimization by optimizing the mean scaled interval score loss function. The IFPA-CNN model has been compared with several other state-of-the-art algorithms also, and it outperforms the others in terms of probabilistic coverage and skill score. The proposed model has been tested on the load dataset, but it can be easily applied to any other time-series dataset. The only limitation of the proposed model is that it uses a bit amount of GPU when compared to run over the CPU. In future, work will be done to improve this aspect further.

CRediT authorship contribution statement

Seyed Mohammad Jafar Jalali: Investigation, Software, Writing – original draft. **Parul Arora:** Data curation, Formal analysis. **B.K. Panigrahi:** Formal analysis, Visualization. **Abbas Khosravi:** Supervision, Methodology. **Saeid Nahavandi:** Methodology, Conceptualization. **Gerardo J. Osório:** Funding acquisition, Validation, Visualization. **João P.S. Catalão:** Validation, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

J.P.S. Catalão acknowledges the support by FEDER funds through COMPETE 2020 and by Portuguese funds through FCT, under POCI-01-0145-FEDER-029803 (02/SAICT/2017). Also, G.J. Osório acknowledges the financial support from REMIT through the UIDB/05105/2020 Program Contract, funded by national funds through the FCT I.P.

References

[1] P. Arora, B. Panigrahi, P. Suganthan, Shallow neural networks to deep neural networks for probabilistic wind forecasting, in: 2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), IEEE, 2021, pp. 377–382.

[2] Y. Li, R. Wang, Z. Yang, Optimal scheduling of isolated microgrids using automated reinforcement learning-based multi-period forecasting, IEEE Trans. Sustain. Energy 13 (1) (2021) 159–169.

[3] P. Arora, A. Khosravi, B. Panigrahi, P. Suganthan, Remodelling state-space prediction with deep neural networks for probabilistic load forecasting, IEEE Trans. Emerg. Top. Comput. Intell. (2021).

[4] P. Arora, H. Kumar, B.K. Panigrahi, Prediction and analysis of COVID-19 positive cases using deep learning models: A descriptive case study of India, Chaos Solitons Fractals 139 (2020) 110017.

[5] X. Guo, Q. Zhao, D. Zheng, Y. Ning, Y. Gao, A short-term load forecasting model of multi-scale CNN-LSTM hybrid neural network considering the real-time electricity price, Energy Rep. 6 (2020) 1046–1053.

[6] A. Baldominos, Y. Saez, P. Isasi, Evolutionary convolutional neural networks: An application to handwriting recognition, Neurocomputing 283 (2018) 38–52.

[7] M. Suganuma, S. Shirakawa, T. Nagao, A genetic programming approach to designing convolutional neural network architectures, in: Proceedings of the Genetic and Evolutionary Computation Conference, 2017, pp. 497–504.

[8] S.M.J. Jalali, S. Ahmadian, M. Khodayar, A. Khosravi, V. Ghasemi, M. Shafiekhah, S. Nahavandi, J.P. Catalão, Towards novel deep neuroevolution models: chaotic levy grasshopper optimization for short-term wind speed forecasting, Eng. Comput. (2021) 1–25.

[9] M. Khodayar, M.E. Khodayar, S.M.J. Jalali, Deep learning for pattern recognition of photovoltaic energy generation, Electr. J. 34 (1) (2021) 106882.

[10] S.M.J. Jalali, P.M. Kebria, A. Khosravi, K. Saleh, D. Nahavandi, S. Nahavandi, Optimal autonomous driving through deep imitation learning and neuroevolution, in: 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC), IEEE, 2019, pp. 1215–1220.

[11] S.M.J. Jalali, M. Ahmadian, S. Ahmadian, A. Khosravi, M. Alazab, S. Nahavandi, An oppositional-Cauchy based GSK evolutionary algorithm with a novel deep ensemble reinforcement learning strategy for COVID-19 diagnosis, Appl. Soft Comput. 111 (2021) 107675.

[12] S.M.J. Jalali, S. Ahmadian, A. Kavousi-Fard, A. Khosravi, S. Nahavandi, Automated deep cnn-lstm architecture design for solar irradiance forecasting, IEEE Trans. Syst. Man Cybern.: Syst. (2021).

[13] P.P. Biswas, P. Arora, R. Mallipeddi, P. Suganthan, B. Panigrahi, Optimal placement and sizing of FACTS devices for optimal power flow in a wind power integrated electrical network, Neural Comput. Appl. 33 (12) (2021) 6753–6774.

[14] I. Koprinska, D. Wu, Z. Wang, Convolutional neural networks for energy time series forecasting, in: 2018 International Joint Conference on Neural Networks (IJCNN), IEEE, 2018, pp. 1–8.

[15] H.J. Sadaei, P.C.d.L. e Silva, F.G. Guimarães, M.H. Lee, Short-term load forecasting by using a combined method of convolutional neural networks and fuzzy time series, Energy 175 (2019) 365–377.

[16] N. Khan, I.U. Haq, S.U. Khan, S. Rho, M.Y. Lee, S.W. Baik, DB-Net: A novel dilated CNN based multi-step forecasting model for power consumption in integrated local energy systems, Int. J. Electr. Power Energy Syst. 133 (2021) 107023.

[17] X. Zhang, K. He, Y. Bao, Error-feedback stochastic modeling strategy for time series forecasting with convolutional neural networks, Neurocomputing 459 (2021) 234–248.

[18] M. Imani, Electrical load-temperature CNN for residential load forecasting, Energy 227 (2021) 120480.

[19] J. Kim, J. Moon, E. Hwang, P. Kang, Recurrent inception convolution neural network for multi short-term load forecasting, Energy Build. 194 (2019) 328–341.

[20] G. Zhang, X. Bai, Y. Wang, Short-time multi-energy load forecasting method based on CNN-Seq2Seq model with attention mechanism, Mach. Learn. Appl. (2021) 100064.

[21] X. Dong, L. Qian, L. Huang, Short-term load forecasting in smart grid: A combined CNN and K-means clustering approach, in: 2017 IEEE International Conference on Big Data and Smart Computing (BigComp), IEEE, 2017, pp. 119–125.

[22] W. Zhang, Z. Qu, K. Zhang, W. Mao, Y. Ma, X. Fan, A combined model based on CEEMDAN and modified flower pollination algorithm for wind speed forecasting, Energy Convers. Manage. 136 (2017) 439–451.

[23] L. Xiao, W. Shao, M. Yu, J. Ma, C. Jin, Research and application of a combined model based on multi-objective optimization for electrical load forecasting, Energy 119 (2017) 1057–1074.

[24] M. De, G. Das, K. Mandal, An effective energy flow management in grid-connected solar–wind-microgrid system incorporating economic and environmental generation scheduling using a meta-dynamic approach-based multiobjective flower pollination algorithm, Energy Rep. 7 (2021) 2711–2726.

[25] H.M. Dubey, M. Pandit, B. Panigrahi, Hybrid flower pollination algorithm with time-varying fuzzy selection mechanism for wind integrated multi-objective dynamic economic dispatch, Renew. Energy 83 (2015) 188–202.

[26] S. Velamuri, S. Sreejith, P. Ponnambalam, Static economic dispatch incorporating wind farm using flower pollination algorithm, Perspect. Sci. 8 (2016) 260–262.

[27] S. Sahoo, S. Swain, R. Dash, P. Sanjeevikumar, J. Reddy, V. Subburaj, Novel Gaussian flower pollination algorithm with IoT for unit price prediction in peer-to-peer energy trading market, Energy Rep. (2021).

[28] M. Abdel-Basset, L.A. Shawky, Flower pollination algorithm: a comprehensive review, Artif. Intell. Rev. 52 (4) (2019) 2533–2557.
[29] M. Saffari, M. Khodayar, S.M.J. Jalali, M. Shafie-khah, J.P. Catalão, Deep convolutional graph rough variational auto-encoder for short-term photovoltaic power forecasting, in: 2021 International Conference on Smart Energy Systems and Technologies (SEST), IEEE, 2021, pp. 1–6.
[30] S.M.J. Jalali, M. Khodayar, S. Ahmadian, M. Shafie-khah, A. Khosravi, S.M.S. Islam, S. Nahavandi, J.P. Catalão, A new ensemble reinforcement learning strategy for solar irradiance forecasting using deep optimized convolutional neural network models, in: 2021 International Conference on Smart Energy Systems and Technologies (SEST), IEEE, 2021, pp. 1–6.
[31] S.M.J. Jalali, S. Ahmadian, A. Khosravi, M. Shafie-khah, S. Nahavandi, J.P. Catalao, A novel evolutionary-based deep convolutional neural network model for intelligent load forecasting, IEEE Trans. Ind. Inf. (2021).
[32] S.M.H. Rizvi, Time series deep learning for robust steady-state load parameter estimation using 1D-CNN, Arab. J. Sci. Eng. (2021) 1–14.
[33] S. Ma, A hybrid deep meta-ensemble networks with application in electric utility industry load forecasting, Inform. Sci. 544 (2021) 183–196.
[34] A. Bracale, P. Caramia, P. De Falco, T. Hong, Multivariate quantile regression for short-term probabilistic load forecasting, IEEE Trans. Power Syst. 35 (1) (2019) 628–638.
[35] T. Hong, S. Fan, Probabilistic electric load forecasting: A tutorial review, Int. J. Forecast. 32 (3) (2016) 914–938.
[36] T. Hong, P. Pinson, S. Fan, H. Zareipour, A. Troccoli, R.J. Hyndman, Probabilistic energy forecasting: Global energy forecasting competition 2014 and beyond, Int. J. Forecast. 32 (3) (2016) 896–913.
[37] D. Salinas, V. Flunkert, J. Gasthaus, T. Januschowski, DeepAR: Probabilistic forecasting with autoregressive recurrent networks, Int. J. Forecast. 36 (3) (2020) 1181–1191.
[38] S.S. Rangapuram, M.W. Seeger, J. Gasthaus, L. Stella, Y. Wang, T. Januschowski, Deep state space models for time series forecasting, Adv. Neural Inf. Process. Syst. 31 (2018) 7785–7794.
[39] Y. Yang, W. Hong, S. Li, Deep ensemble learning based probabilistic load forecasting in smart grids, Energy 189 (2019) 116324.
[40] W. Zhang, H. Quan, O. Gandhi, R. Rajagopal, C.-W. Tan, D. Srinivasan, Improving probabilistic load forecasting using quantile regression NN with skip connections, IEEE Trans. Smart Grid 11 (6) (2020) 5442–5450.