Scientia Iranica E (2023) 30(2), 803–813

SCIENTIA
IRANICA

Sharif University of Technology
Scientia Iranica
Transactions E: Industrial Engineering
http://scientiairanica.sharif.edu

ParDeeB: A graph framework for load forecasting based on parallel DeepNet branches

N. Neshat[a,*], M. Sardari Zarchi[b], and H. Mahlooji[c]

a. *Department of Industrial Engineering, Meybod University, Yahyazadeh Blvd., Khorramshahr Blvd., Meybod, Yazd, Iran.*
b. *Department of Computer Engineering, Meybod University, Yahyazadeh Blvd., Khorramshahr Blvd., Meybod, Yazd, Iran.*
c. *Department of Industrial Engineering , Sharif University of Technology, Azadi Ave, Tehran, Iran.*

Received 13 July 2020; received in revised form 3 January 2021; accepted 1 March 2021

KEYWORDS
Load forecasting;
Deep neural networks;
Parallel deep networks;
Residential load demand.

Abstract. In recent times, energy demand forecasting has become an important area of research due to its significant impact on Greenhouse Gas (GHG) emissions and global warming. Long-term historical dependence, complexity, and nonlinearity are all characteristics of load prediction problems. Numerous methodologies, ranging from statistical to computational intelligence techniques, have been used up to this point in this research field. According to the literature, deep learning can handle these properties better than other approaches. However, the recent state-of-the-art deep network models are not robust against various historical dependencies. In this study, we propose a graph framework based on parallel DeepNet branches to tackle this challenge. This framework consists of multi-parallel branches in which different kinds of networks can be integrated. Parallel branches individually represent the historical dependency of determinants and improve performance when different historical dependencies exist in the data. In this case study, the performance of the proposed model is examined through a comparative study with state-of-the-art deep network models. According to the comparison, the proposed framework can improve load forecasting by a significant margin on average.

1. Introduction

The core of energy management is the forecasting of energy demand, which enables the suppliers to manage numerous routine operational decisions. However, there are some problems with accurate forecasting due to its complicated and uncertain nature. In short-term energy demand forecasting, dependence on time, consumer choice, and external weather conditions make load forecasting a complex research area with rare features and important consequences.

Studies suggest that households are major contributors to total global energy demand [1,2]. Therefore, residual load forecasting is central to this research area, both from an academic and a practical point of view. Different forecasting approaches, from statistical to computational intelligence have been applied in the past [3,4]. The related literature is typically divided into two main categories based on the number of techniques used: stand-alone and hybrid methods, each involving one or more techniques. Depending on the type of underlying technique, stand-alone methods are further divided into three categories: statistical, casual, and computational intelligence methods.

The following statistical techniques simulate the

[*] *Corresponding author.*
E-mail addresses: neshat@meybod.ac.ir (N. Neshat); sardari@meybod.ac.ir (M. Sardari Zarchi); hmahlooji@sharif.edu (H. Mahlooji)

doi: 10.24200/sci.2021.56343.4673

dynamic relationship between lagged determinant values and the forecast load demand based on historical data: Autoregressive (AR) and Double Seasonal Holt-Winter (DSHW) models [5], Autoregressive models with exogenous inputs (ARX) [6], ARX threshold models (TRX) [7], models based on Generalized Autoregressive Conditional Heteroscedasticity (GARCH) [8–10] Autoregressive Integrated Moving Average (ARIMA) models [8,12], semi/nonparametric models [13,6] or Dynamic Regression (DR), Seasonal Autoregressive Integrated Moving Average (SARIMA) [14,15], Transfer Function (TF) models [16] and Grey models [17–20]. The hybrid version of the mentioned methods has also been proposed, e.g., Wavelet-based models [12,21,22].

Casual methods focus on formulating dynamic relationships between causal variables (determinants) and forecasted load demand. Causal models utilize the least-square fitting method to extract forecasted load demand in terms of its determinants, such as temperature, humidity, and lagged data [23]. Different casual methods such as Linear Regression (LR) [24], Nonlinear Regression (NLR) [25–29], and logistic or logit regression (LoR) [30] have been widely used in the literature.

Real-world problems, like load forecasting, are typically nonlinear in nature, but they also tend to be linear models, which means they may not work well with data that exhibits strong historical dependency patterns. In other words, they only can only handle a low data rate such as weekly patterns, and the nonlinear behavior of load demand can become too complex to predict [31]. Also, they are only used for short to medium-term forecasts. However, their internal logic is obvious, and they are called “White-box” methods.

Computational intelligence methods such as Artificial Neural Network (ANN) and Support Vector Machine (SVM) are widely used in this research field given their ability to process hidden data features and nonlinear modeling [32–38]. In addition, they are preferred for all time forecast intervals. However, they are unable to process data that shows a strong pattern of historical dependence [39].

Ekonomou used ANN to forecast the load for a country. Weather conditions, historical demand data, Gross Domestic Production (GDP), and load capacity are the four factors that can be used to forecast load demand using the Multi-Layer Perceptron (MLP) model [40]. The comparison results obtained from the data of the years 2012 and 2016 showed that the MLP model outperforms the LR and SVM models. In the year 2014, Kialashaki et al. estimated energy demand for industrial sectors in terms of various determinants, such as the price of energy carriers and GDP using ANN and LR. The experimental results showed the superiority of the ANN model over LR based on the accuracy and reliability indicators [41]. Abedinia and Amjadi combined a radial neural network with a stochastic search to forecast short-term load demand. The prediction results were validated by comparing them to the results of the MLP network, the echo state network, and the wavelet transform [42]. Gajowniczek and Zabkowski [43] came to similar conclusions.

In the context of load forecasting, He proposed a DeepNet model to forecast short-term load demand. The Convolutional Neural Network (CNN) was used to extract the features from historical patterns, which continues to form the basis of load forecasting [44].

Shi et al. have proposed a deep-learning approach to estimate short-term electrical load. To estimate demand at two different levels, deep Recurrent Neural Networks (RNNs) were used. both at the regional aggregate level and at the household disaggregate level. Based on the experimental results, the authors found that the deep RNN performs better than the shallow neural network [45]. Additionally, Rahman et al. presented an RNN-based approach to predict electricity demand for residential and commercial buildings [46]. Kong et al. developed a DeepNet-based framework to forecast short-term load demand by considering the device consumption patterns. The results obtained imply that load demand is strongly dependent on residential behavior [47].

Recently, Bedi and Toshniwal developed the DeepNet, the RNN and the SVM models using load demand data from the Union Territory of Chandigarh, India [48]. Cluster analysis was carried out on the data to take into account the dynamics of the load demand determinants. The comparison revealed the superiority of the DeepNet framework for short-term load forecasting over RNN and SVM. Future research was recommended in this study to investigate trends in load demand patterns by examining various nonlinear exogenous determinants, such as climate conditions and economic variables. It should be noted that clustering the data based on seasonal, daily, and interval data has resulted in several DeepNet models. In fact, different periodic data histories required clustering and led to multiple load forecasting models.

According to the suggestions of the state-of-the-art studies [48] and to solve the mentioned problem, a graph framework based on parallel DeepNet branches is proposed, which has the following characteristics:

- Due to the high generalization ability of computational intelligence models, it is able to capture nonlinearities in data and model complex nonlinear forecasting problems;
- Due to its many deep hidden layers, it can handle long-term data dependency models, unlike conventional ANN models;

- It is robust to different exogenous variables with different historical data in the case of forecasting problems thanks to the inclusion of dummy variables. Previous studies required the data to be clustered by determinant to prevent the model from being over-parameterized, such as time of day. Dummy variables have been utilized in the proposed modeling framework to incorporate the daily, monthly, and yearly seasonality into the model and;
- The most important feature is its robustness to different exogenous variables with different historical dependencies when using multi-input parallel branches.

To address the above features, we propose a graphical framework built on parallel DeepNet branches. The paper is organized as follows: Section 2 introduces the research preliminary findings and cutting-edge methodology. The proposed DeepNet modeling framework is then described in Section 3. By taking the case study into account, Section 4 assesses the proposed model as well as the base forecasters (which are taken from the literature and used as the benchmark). Section 5 includes the numerical results. The paper is finally concluded in Section 6, which also summarizes the key findings.

2. Preliminary

2.1. Load forecasting

Short-term load demand depends on weather conditions, customers daily and monthly demand patterns, and the effects of changing conditions on these patterns. Previous studies have focused on meteorological elements, i.e., temperature, solar radiation, humidity, and wind speed as exogenous variables [49]. Load demand forecasting typically consists of examining historical load data along with information about the past, current, and predicted futures of exogenous variables.

2.2. Deep Neural Networks (DNN)

The term deep learning generally refers to the Deep Neural Network (DNN). A DNN is a complex neural network with more than two hidden layers. DNN can have different structures and topologies from which Dense Neural Network, RNN, and CNN can be used as prediction tools [50]. Dense or fully connected networks have a feed-forward structure in which each neuron is connected to all neurons in the next layer. Although dense networks can be used to build a nonlinear model, they are not suitable for models that rely on historical data [44]. To build a historical-based model, RNNs are widely used. The RNN takes into account the dependencies between the data nodes leading to the recording of the historical dependency pattern in the data. Dependencies are supported by maintaining accumulated knowledge of subsequent timestamps using network or feedback loops. A time-domain unrolled RNN is shown in Figure 1. In this figure, I_t denotes the input value at timestamp t, S_t denotes the state at timestamp t, and O_t denotes the output at timestamp t. The current state S_t is calculated in terms of the current input I_t and the previous hidden state S_{t-1}. Mathematically it can be given as:

$$S_t = f_\theta(UI_t + WS_{t-1}) \qquad O_t = f_\alpha(VS_t). \tag{1}$$

RNNs are capable of dealing with short-term dependencies, but they fail to handle long-term dependencies due to the vanishing gradient event (He, 2017 [44]). To solve the above problem, a deep learning framework has been developed. The two state-of-the-art deep learning models are Long-Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks. CNNs with 2D filters were first introduced in computer vision applications. However, CNNs with 1D filter are more recently used for time series or history-based model. Figure 2 schematically shows how the 1D filter can be used. Compared to RNNs, CNNs are faster and require fewer computational resources, but RNNs such as LSTMs have the ability to build a model based on history.

2.3. Performance metrics

Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE) are the measures used to evaluate model fit. These performance metrics were chosen because they are commonly used in the relevant literature and because they are also considered the primary measures of fitness strength. The MAPE approach is often used in the forecasting literature. Though mathematically straightforward, it has some

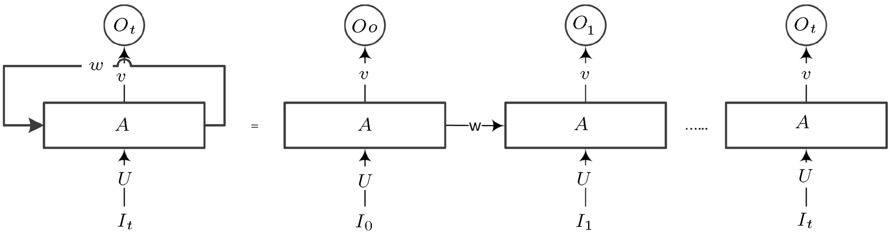

Figure 1. Architecture of an RNN network [51].

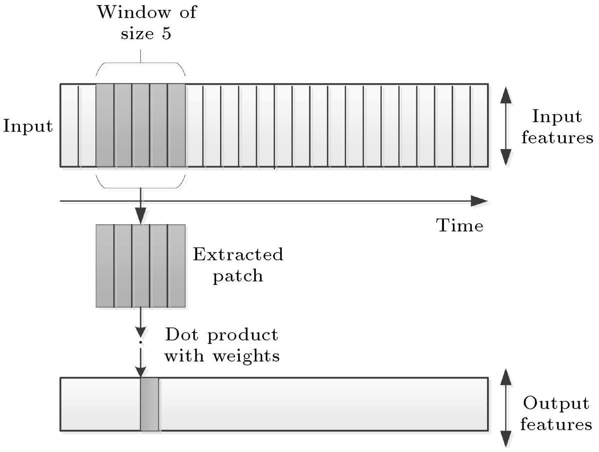

Figure 2. Performing 1D convolution filter [52].

drawbacks. The MAPE approach cannot handle actual data with a zero value. The impact of this limitation depends on the percentage of the actual dataset that is zero. It can be expressed mathematically as Eq. (2):

$$MAPE = \frac{100}{N} \sum \frac{|y_i - \acute{y}_i|}{y_i}, \tag{2}$$

where y_i represents the actual load demand and y'_i represents the predicted load demand and n represents the number of values.

RMSE is another common metric used in evaluating the accuracy of the fitness. RMSE provides a way to evaluate the mean error taking into account the load demand. The accuracy of the output load demand, which is the primary goal of fitness, can be captured by methods like MAPE, so this metric is crucial because it can offer good accuracy. Mathematically, it can be presented as Eq. (3):

$$RMSE = \sqrt{\frac{1}{N} \sum (y_i - \acute{y}_i)^2}. \tag{3}$$

3. DNN framework

We proposed a novel DNN framework based on Parallel DeepNet branches to forecast day-ahead load demand. Therefore, we called our model ParDeeB. Each branch of ParDeeB is a subnet where different types of deep networks like RNN, CNN, or Dense networks can be used. These branches are concatenated in an acyclic graph to form the final structure. The ParDeeB framework consists of three main phases: (i) data preparation, (ii) multi-input RNN branches, and (iii) non-history-based determinants.

3.1. Data preparation

To robustly forecast load demand, several aspects of the input data must be considered. Let $D = X = [X_1, X_2, \cdots, X_n]$, $Y = [y_1, y_2, \cdots, y_n]$ be the n input data samples, where $X_t = x_t^1, x_t^2, \cdots, x_t^m$ includes m determinants and y_t is the target value (actual load value). To adequately represent a periodic pattern, historical and non-historical determinants must be considered in the load forecast. Deep RNN can be used to incorporate the historical determinants. Therefore, in the first step of the proposed framework, k previous timestamps of the determinant x_t^j are formed as the sequential vector $S_t^j = x_{t-k}^j; x_{t-1}^j; x_t^j$, where k is the size of the lookback (lagged data). If the size of the lookbacks is assumed to be the same for all determinants, the sequential input variable $X_{seq} = [S_1, S_2, \cdots, S_n]$ obtained from X is a 3D tensor (matrix) where $S_t = [S_t^1, S_t^2, \cdots, S_t^m]$. Figure 3 shows X_{seq} when the lookback size of all determinants is the same.

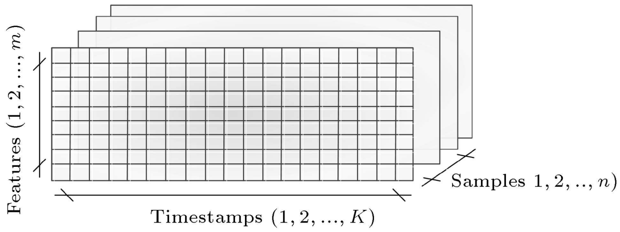

Figure 3. An overview of X_{seq}, a 3D timestamp data tensor.

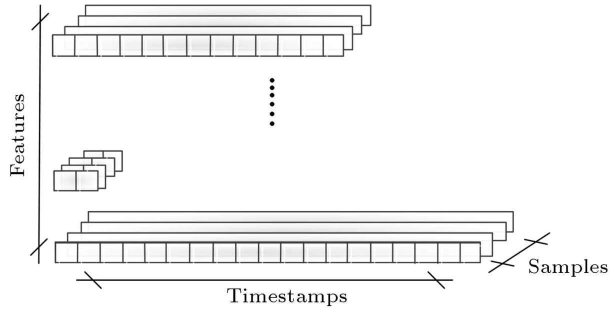

Figure 4. Overview of X_{seq} with different timestamp lengths.

Depending on the importance of the history of the variable and its periodic type, previous timestamps can be sampled at time step d. So the sequential vector is defined as $S_t^j = x_{t-kd}^j, x_{t-d}^j, , x_t^j$. For example, instead of considering the previous timestamp for each hour, we can only sample one value every 6 hours, which means that d is set to 6. Since each variable has a different periodic history, we have prepared a sequence vector of each variable with its individual time step (d) and lookback (k). So the sequence lengths of S_{seq}^j are different and therefore X_{seq} is not a 3D tensor. Figure 4 shows the overview of X_{seq} when we have different steps and lookback for the determinants. Note that the d and k are hyperparameters and are set experimentally.

3.2. Multi-input RNN branches

When the input to the neural network is in the form of a tensor, the design of the structure of the DNN is

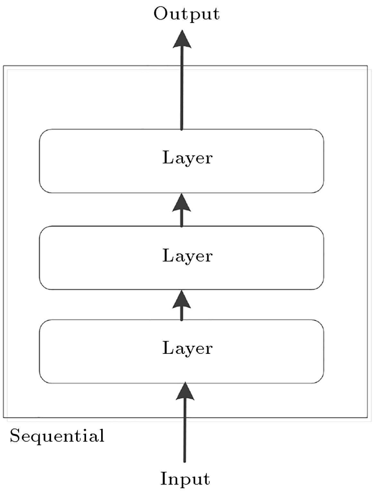

Figure 5. A sequential model: A linear stacking of layers.

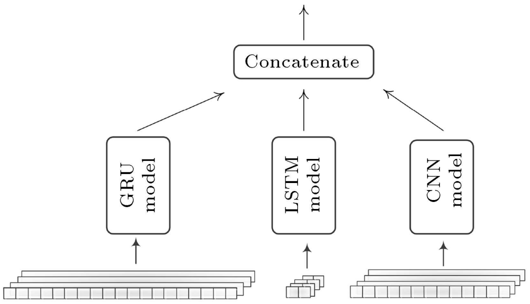

Figure 6. A simple model with three parallel branches consisting of an individual determinant timestamp vector.

simple. In this case, the network can be implemented sequentially with a linear stacking of layers, as shown in Figure 5. Although the sequential model is so common, it is inflexible when the input data does not come under the form of a tensor, e.g., Figure 4. Additionally in a forecasting model, each determinant could require a different type of neural networks such as CNNs, LSTMs, or GRUs. To overcome this challenge, we innovatively proposed parallel deep branching in our framework to represent the history of each determinant individually. Parallel branches process their determinants using RNNs and dense networks. Then the branches are merged together through concatenated and dense layers. Concatenated layer brings all the output from the previous branches and integrates them into the next layer. In fact, the concatenated layer only linked the output of different branches into one layer. For example, Figure 6 shows a simple model with three branches that are merged with a concatenated layer.

3.3. Non-historical determinants

Some input determinants are not historical-based data. However, these determinants can have a causal effect on load demand. These casual determinants are usually modeled by fully connected or dense layers. In this research, we add a dense DNN branch to our framework to benefit from the casual determinants. In order to ensemble all RNN and dense branches, a directed acyclic graph topology is employed. In this framework called ParDeeB, the input is processed by several parallel branches. Then the branch outputs are concatenated into a connected layer. Figure 7 shows the overview of the ParDeeB framework. Note that each branch consists of a dense layer and an RNN layer.

4. Experiment

In this section, we conduct the empirical study to evaluate the ParDeeB framework and to analyze the forecasting accuracy of the different basic models, i.e., GRU, CNN, Dense, LSTM, and the proposed model.

4.1. Case study

To evaluate the proposed DL framework and to evaluate the forecasting accuracy of the different baseline models, we use the data of peak load and meteorological conditions (e.g., temperature and wind speed) from Shahrekord, Iran during the period from 03/03/2015 to 03/03/2018 which belong to [51]. Detailed statistical information on the exogenous variables is provided in Table 1. According to the data and the literature, customer demand patterns vary by hour, day, and month because the load value behaves differently under different holiday states; times of night/day; day of the week; and season. Figure 8 demonstrates the periodic pattern of load demand over the yearly time horizon in terms of daily intervals. Therefore, dummy variables were defined to model the daily, weekly, and monthly periodic effects. A dummy variable with three classes-low, medium, and high load type-was used to model the 24 hours of the day and seven days of the week, each of which seven classes was assigned. Similar to how the seven days of the week were modeled with seven classes, the 24 hours of the day were modeled by a dummy variable with three classes labeled as low, medium, and high load types. Seasonal information was accounted for by incorporating two dummy variables representing the number of months and the day of the month. The main purpose of the interval/daily characterization is to allow the proposed model to forecast the load for all seasons, days, and user-specified time intervals of the day. Similarly, the intraday patterns of public holidays differ from those of typical days; Therefore, the binary dummy variable of Holiday was considered in the model. In summary, 6 dummy variables were added to model their impact on the load demand pattern. Table 1 contains the statistical information of the data used (such as the maximum, average, minimum, and peak load demand of the days). In the model development phase, we divided the data into training, validation, and testing

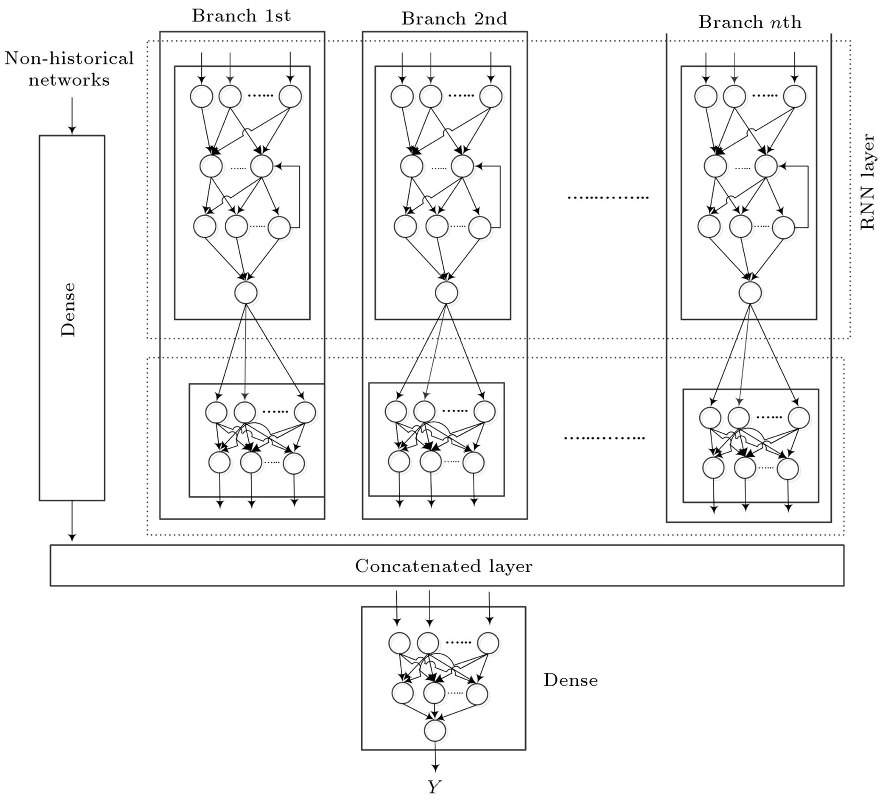

Figure 7. The overview of the proposed framework (ParDeeB).

Table 1. Daily statistics of the model inputs (exogenous and lagged data) and the related outputs.

Input	Min value	Max value	Mean value
Min temp	–18.40	21.80	3.6
Max temp	–2.00	37.40	22.27
Min humidity	–1.70	99.00	19.23
Max humidity	4.00	100.00	64.19
Mean humidity	5.00	95.00	41.47
Rainfall	0.00	42.60	0.62
Snow	0.00	34.80	0.13
Total rainfall	0.00	42.60	0.75
Snow height	0.00	11.30	0.05
Wind direction (03)	0.00	360.00	206.20
Wind speed (03)	0.00	13.00	1.81
Wind direction (09)	0.00	360.00	184.43
Wind speed (09)	0.00	16.00	4.02
Wind direction (15)	0.00	3210.00	207.14
Wind speed (15)	0.00	18.00	4.00
Max wind direction	4.00	390.00	205.86
Max wind speed	1.00	35.00	10.40
Load demand	84.00	355.89	212.81
Peak load	161.64	355.89	262.60

sets. The first percent of the samples (i.e., the 1st, 2nd, and 3rd year samples) were used to train and validate the various models, and the remainder (the 4th year samples) were used to test the performance of the model used. This study includes 30768 samples with 23 determinants and one target which are recorded every hour. With 24 energy consumption values per day, the training and validation dataset consists of 23076 data points. The test dataset also includes 7692 data points.

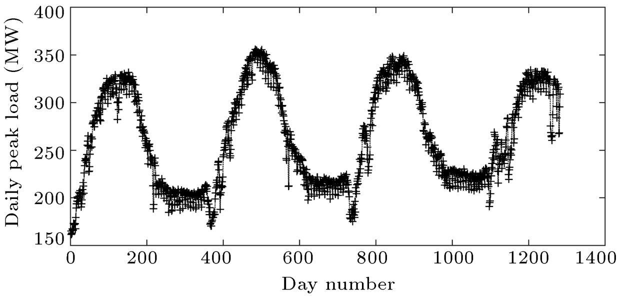

Figure 8. Periodicity of peak load demand versus daily intervals.

4.2. Framework implementation

To evaluate the ParDeeB framework, we implemented and evaluated it in a case study. Python 3.7 is used in the ParDeeB implementation. This implementation was tested on a desktop computer with a Geforce 1060 GTX GPU, an i3–6100 CPU, 12 GB RAM, and a Ubuntu Linux operating system. The model uses Keras 2.2 with TensorFlow 1.10 as the backend for deep learning algorithms. Additionally, other Python packages such as Scikit-lean, NumPy, Openpyxl, and Matplotlib are used for pre-processing and post-processing algorithms on the input data. The determinant values of the dataset have different scale variations, which can make the training algorithm less efficient. The data set is therefore statistically standardized. Data normalization for each sample X is performed as Eq. (4):

$$X_{new} = \frac{X-\mu}{\sigma}, \tag{4}$$

where μ and σ are the mean and the standard deviation, respectively.

5. Results and discussion

As we mentioned in Section 3, the ParDeeB framework can have as many parallel branches as needed. Based on the nature of our case study, we considered three parallel branches. The first and second branches are responsible for historical data, while the third branch presents non-historical data. The input to the first branch is a sequence of previous load demand values. The second branch contains 19 determinant sequences. For example, the sequences of previous temperature, wind, etc. The third branch accepts six dummy determinants, including the value of the year, the month, etc. Different network types, time steps, and lookbacks are investigated to determine the best model configuration for our case study. Table 2 shows the top five configurations of the tested configurations. Based on the experimental result, the best configuration can be obtained using LSTM for the first branch, GRU for the second branch, and Dense for the third branch. From this configuration, we can conclude that LSTM can perfectly model long-term historical data instead of GRU and CNN. Therefore, for the first branch where the previous load demand is to be modeled, using LSTM with a long period (*lookback* = 168) and sampling all data ($d = 1$) is the best configuration. Furthermore, we can see that other historical determinants need simpler models. Therefore, for the second branch, GRU can work correctly when the determinant sequences have a time-step $d = 24$ hours (means daily sampling) and *lookback* = 7. The optimal choice for the third branch is a dense pattern without recurring layers. This branch accepts only dummy determinants without considering previous values. These three layers are merged into one by a concatenated layer and finally

Table 2. The results of different configurations, where the symbols n, d, and l denote the number of neurons, time step, and lookback values, respectively.

No. of Conf.	Branch 1	Branch 2	Branch 3	MAPE test error
1	LSTM with $n = 64$, $d = 1$, $l = 168$	GRU with $n = 128$, $d = 24$, $l = 7$	Dense with $n = 64$	6.54
2	LSTM with $n = 64$, $d = 1$, $l = 168$	LSTM with $n = 64$, $d = 24$, $l = 7$	Dense with $n = 128$	6.71
3	LSTM with $n = 128$, $d = 1$, $l = 168$	CNN with $n = 32$, $d = 24$, $l = 14$	Dense with $n = 256$	7.02
4	GRU with $n = 256$, $d = 1$, $l = 84$	LSTM with $n = 64$, $d = 24$, $l = 7$	Dense with $n = 128$	7.12
5	LSTM with $n = 64$, $d = 2$, $l = 336$	GRU with $n = 64$, $d = 24$, $l = 14$	Dense with $n = 64$	7.14

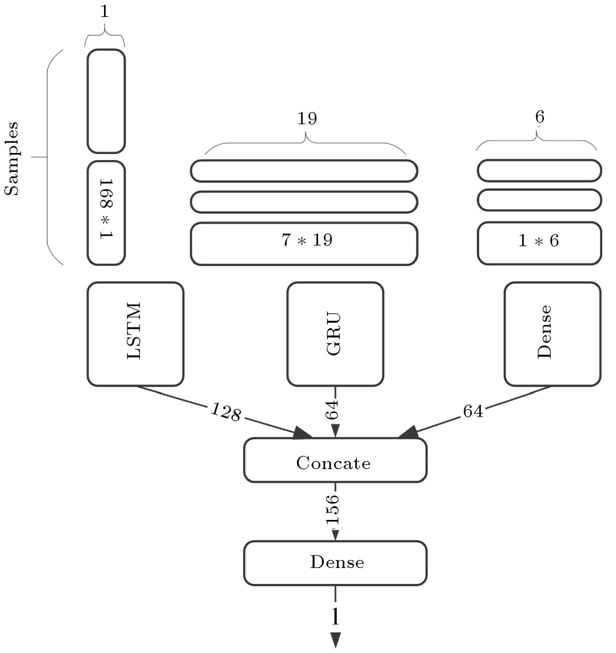

Figure 9. The overview of the optimal configuration of the ParDeeB model.

result in the last dense layer of 256 neurons. Figure 9 shows an overview of the optimal configuration of the ParDeeB model for our case study. According to the experimental result, the best configuration can be obtained by using LSTM for the first branch, GRU for the second branch, and Dense for the third branch. From this configuration, we can conclude that LSTM is better at modeling long-term historical data than GRU and CNN. For the first branch, where the previous load demand needs to be modeled, using LSTM with a long sequence ($lookback = 168$) and sampling all data ($d = 1$) is the best configuration. Furthermore, we can see that other historical determinants require simpler models. Therefore, GRU can perform well for the second branch which has determinant sequences time step $d = 24$ hours (means daily sampling) and $lookback = 7$. The optimal choice for the third branch is a Dense model with no recurrent layers. This branch only accepts dummy determinants without taking into account the preceding values. These three layers are merged by a concatenated layer and finally, are led to last dense layer with 256 neurons. Figure 9 shows an overview of the optimal configuration of the ParDeeB model for our case study.

Note that each branch has hyperparameters whose optimal values are determined experimentally. The main hyperparameters are the Activation function, Regularization function, Dropout value, and Recurrent dropout value. The optimal value of these hyperparameters for the ParDeeB model implemented according to Figure 9 is presented in Table 3. Dropout is used to overcome the overfitting problem in the training phase of the neural network. In the dropout procedure, some weights between layer connections are randomly selected and temporarily set to zero. However, for recurrent neurons like LSTM, it is also recommended to randomly ignore some recurrent values at each epoch. In Table 3, the dropout value refers to the dropout between layers and recurrent dropout refers to the dropout value of the LSTM and GRU neurons used in the model.

After explaining the selection of experimental configuration and obtaining the topology of the optimal model, it is necessary to compare the accuracy of the models. To validate our forecasting model, Pardeeb has been compared with four well-known species of the DNN, including: (i) Gru, (ii) LSTM, (iii) Dense, and CNN. These four neural network models were developed in standard configuration. Table 4 shows the performance of the four deep networks and the ParDeeB model based on MAPE and RMSE for the training and test data. The accuracy results in Table 4 show that ParDeeB makes more reliable and accurate forecasting than other developed models.

Table 3. The optimal value of these hyperparameters for the ParDeeB model.

Branch	**Activation function**	**Regularization**	**Dropout**	**Recurrent dropout**
LSTM	Tanh	L2	0.1	0.4
GRU	Tanh	L2	0.05	0.05
Dense	ReLU	No	0.2	0

Table 4. Comparison of accuracy of developed models based on performance metrics.

Model	**MAPE train error**	**MAPE test error**	**RMSE train error**	**RMSE test error**
CNN	16.78	16.99	21.90	21.72
GRU	6.55	7.42	9.68	11.34
Dense	6.31	7.31	9.51	11.59
LSTM	6.15	7.22	9.24	11.26
ParDeeB	5.52	6.54	8.95	10.88

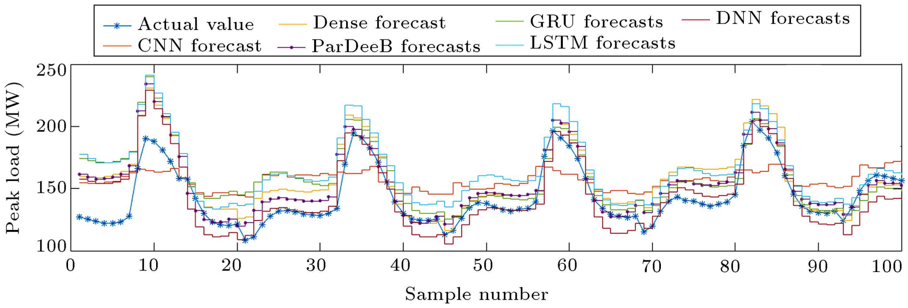

Figure 10. The result of the Pardeeb and other models for 100 samples compared to the actual values.

In addition, the models were schematically analyzed for forecasting accuracy according to Figure 10. This figure shows the diagram of the actual value and the predicted values for the first 100 test data. The number of test data and peak load demand, both expressed in megawatts, are represented by the x and y axes in Figure 10. The red line marked by the star represents the actual values, while the other colors represent the forecasts of the developed models. As these results show, the ParDeeB model can outperform other developed networks.

6. Conclusion

Significant progress has been made in the development of efficient and accurate load forecasting models through the emerging intelligent computing approach. In this study, a new graphical framework for load forecasting based on DeepNet parallel branches was proposed and compared to the state-of-art models in the field. This framework excels in handling long-term data dependency patterns, capturing data nonlinearity, and robustness to various exogenous variables with different historical data dependencies using multiple-input parallel branches. From the results and observations presented in the previous section, we can conclude that the proposed model is the best alternative for forecasting day-ahead load demand among the developed models. It can significantly improve the accuracy of load forecasting by an average of 3.09%. In detail, the proposed model outperforms the Convolutional Neural Network (CNN) model with an improvement of 10.84% because it is able to handle long-term data dependency patterns with more hidden layers. The CNN model has a shallower structure than the one proposed, which can be explained by the fact that the proposed model has a deep structure. A 0.72% improvement in accuracy can result from the proposed model over the Dense model by incorporating the recursive mechanism. The recurrent parallel structure of the proposed model leads to an improvement of 0.46% compared to the Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) models. These parallel recurrent branches represent the history of each determinant individually when there are different periodic histories in the data. In summary, deep learning models are generally well suited to the load forecasting problem. However, certain contributions in their configurations such as parallel branches make them more accurate and robust.

References

1. Amara, F., Agbossou, K., Dubé, Y., et al. “A residual load modeling approach for household short-term load forecasting application”, *Energy and Buildings*, **187**, pp. 132–143 (2019).
2. Neshat, N. “An approach of artificial neural networks modeling based on fuzzy regression for forecasting purposes”, *International Journal of Engineering*, **28**(11), pp. 1651–1655 (2015).
3. Azadeh, A. and Faiz, Z. “A meta-heuristic framework for forecasting house- hold electricity consumption”, *Applied Soft Computing*, **11**(1), pp. 614–620 (2011).
4. Neshat, N., Amin-Naseri, M.R., and Danesh, F. “Energy models: Methods and characteristics”, *Journal of Energy in Southern Africa*, **25**(4), pp. 101–111 (2014).
5. Kamranrad, R. and Amiri, A. “Robust holt-winter based control chart for monitoring autocorrelated simple linear profiles with contaminated data”, *Scientia Iranica*, **23**(3), pp. 1345–1354 (2016).
6. Weron, R. and Misiorek, A. “Forecasting spot electricity prices: A compari- son of parametric and semiparametric time series models”, *International Journal of Forecasting*, **24**(4), pp. 744–763 (2008).
7. Misiorek, A., Trueck, S., and Weron, R. “Point and interval forecasting of spot electricity prices: Linear vs. non-linear time series models”, *Studies in Nonlinear Dynamics and Econometrics*, **10**(3) (2006)
8. Diongue, A.K., Guegan, D., and Vignal, B. “Forecasting electricity spot mar- ket prices with a k-factor gigarch process”, *Applied Energy*, **86**(4), pp. 505–510 (2009).
9. Garcia, R.C. Contreras, J. Van Akkeren, M., et al. “A garch forecasting model to predict day-ahead electricity prices”, *IEEE Transac-Tions on Power Systems*, **20**(2), pp. 867–874 (2005).
10. Knittel, C.R. and Roberts, M.R. “An empirical examination of restructured electricity prices”, *Energy Economics*, **27**(5), pp. 791–817 (2005).

11. Wang, C., Rao, C., Meng, Y., et al. "Research on impact of shenzhen-hong kong stock connect mechanism on the volatility of stock markets", *Scientia Iranica*, **29**(1), pp. 372–386 (2022).

12. Yang, Z., Ce, L., and Lian, L. "Electricity price forecasting by a hybrid model, combining wavelet transform, arma and kernel-based extreme learning machine methods", *Applied Energy*, **190**, pp. 291–305 (2017).

13. Vilar, J.M., Cao, R., and Aneiros, G. "Forecasting next-day electricity de- mand and price using nonparametric functional methods", *International Journal of Electrical Power and Energy Systems*, **39**(1), pp. 48–55 (2012).

14. Jeong, K., Koo, C., and Hong, T. "An estimation model for determining the an- nual energy cost budget in educational facilities using sarima (seasonal autoregressive integrated moving average) and ann (artificial neural net- work)", *Energy*, **71**, pp. 71–79 (2014).

15. Wang, Y., Wang, J., Zhao, G., et al. "Application of residual modifi- cation approach in seasonal arima for electricity demand forecasting: A case study of china", *Energy Policy*, **48**, pp. 284–294 (2012).

16. Nogales, F.J., Contreras, J., and Conejo, A.J. "Forecasting next- day electricity prices by time series models", *IEEE Transactions on Power Systems*, **17**(2), pp. 342–348 (2002).

17. Mbae, A.M. and Nwulu, N.I. "Day-ahead load forecasting using improved grey verhulst model", *Journal of Engineering, Design and Technology*, **18**(5), pp. 1335–1348 (2020).

18. Kazemi, A., Modarres, M., Mehregan, M., et al. "A markov chain grey forecasting model: A case study of energy demand of industry sector in iran", In *3rd International Conference on Information and Financial Engineering IPEDR*, **12** (2011).

19. Hashem-Nazari, M., Esfahanipour, A., and Fatemi Ghomi, S. "A basic form- focused modeling and a modified parameter estimation technique for grey prediction models", *Scientia Iranica*, **25**(5), pp. 2867–2880 (2018).

20. Ma, X., Wu, W., Wang, Y., et al. "Predicting primary energy consumption using ndgm (1, 1, k, c) model with simpson formula", *Scientia Iranica*, **6**(6), pp. 3379–3395 (2021).

21. Tan, Z., Zhang, J., Wang, J., et al. "Day-ahead electricity price forecasting using wavelet transform combined with arima and garch models", *Applied Energy*, **87**(11), pp. 3606–3610 (2010).

22. Shahriari-kahkeshi, M. "Robust fault detection and isolation scheme using fuzzy wavelet network with a hybrid design algorithm", S*cientia Iranica*, **24**(3), pp. 1467–1481 (2017).

23. Lee, C.-P., Lin, W.-C., and Yang, C.-C. "A strategy for forecasting option price using fuzzy time series and least square support vector regression with a bootstrap model", *Scientia Iranica*, **21**(3), pp. 815–825 (2014).

24. Song, K.-B., Baek, Y.-S., Hong, D.H., et al. "Short-term load forecast- ing for the holidays using fuzzy linear regression method", *IEEE Transactions on Power Systems*, **20**(1), pp. 96–101 (2005).

25. Bilgili, M., Sahin, B., Yasar, A., et al. "Electric energy demands of turkey in residential and industrial sectors", *Renewable and Sustainable Energy Reviews*, **16**(1), pp. 404–414 (2012).

26. Ghiassi, M., and Nangoy, S. "A dynamic artificial neural network model for forecasting nonlinear processes", *Computers and Industrial Engineering*, **57**(1), pp. 287–297 (2009).

27. Kazemi, A., Shakouri, H.G., Menhaj, M.B., et al. "A hierarchical artificial neural network for transport energy demand forecast: Iran case study", *Neural Network World*, **20**(6), p. 761 (2010).

28. Neshat, N., Hadian, H., and Alangi, S.R. "Technological learning modelling towards sustainable energy planning", *Journal of Engineering, Design and Technology*, **18**(1), pp. 84–101 (2020).

29. Neshat, N., Hadian, H., and Behzad, M."Nonlinear arimax model for long-term sectoral demand forecasting", *Management Science Letters*, **8**(6), pp. 581–592 (2018).

30. Dudek, G. "Pattern-based local linear regression models for short-term load forecasting", *Electric Power Systems Research*, **130**, pp. 139–147 (2016).

31. Tsekouras, G., Dialynas, E., Hatziargyriou, N., et al. "A non-linear multivariable regression model for midterm energy forecasting of power systems", *Electric Power Systems Research*, **77**(12), pp. 1560–1568 (2007).

32. Azadeh, A., Neshat, N., Rafiee, K., et al. "An adaptive neural network-fuzzy linear regression approach for improved car ownership estimation and forecasting in complex and uncertain environments: the case of Iran", *Transportation Planning and Technology*, **35**(2), pp. 221–240 (2012).

33. Huang, C.-H., Yang, F.-H., and Lee, C.-P. "The strategy of investment in the stock market using modified support vector regression model", *Scientia Iranica*, **25**(3), pp. 1629–1640 (2018).

34. Kazemi, A., Shakouri, G., Menhaj, M., et al. "A multilevel artificial neural network for residential and commercial energy de- mand forecast: Iran case study", In *International Conference on Man- agement Technology and Applications (ICMTA)*, pp. 24–28 (2010).

35. Khwaja, A., Zhang, X., Anpalagan, A., et al. "Boosted neural networks for improved short-term electric load forecasting", *Electric Power Systems Research*, **1**, pp. 431–437 (2017).

36. Niu, D., Liu, D., and Wu, D.D. "A soft computing system for day-ahead electricity price forecasting", *Applied Soft Computing*, **10**(3), pp. 868–875 (2010).

37. Yao, B., Yao, J., Zhang, M., et al. "Improved support vector machine re- gression in multi-step-ahead prediction for rock displacement surround- ing a tunnel", *Scientia Iranica*, Transaction A, Civil Engineering, **21**(4), p. 1309 (2014).

38. Qiu, X., Ren, Y., Suganthan, P.N.G., et al. "Empirical mode decomposition based ensemble deep learning for load demand time series forecasting", *Applied Soft Computing*, **54**, pp. 246–255 (2017).

39. Maleki, N., Nikoubin, A., Rabbani, M., et al. "Bitcoin price prediction based on other cryptocurrencies using machine learning and time series analysis, *Scientia Iranica*, **30**(1), pp. 285–301 (2020).

40. Ekonomou, L. "Greek long-term energy consumption prediction using artificial neural networks", *Energy*, **35**(2), pp. 512–517 (2010).

41. Kialashaki, A. and Reisel, J.R. "Development and validation of artificial neural network models of the energy demand in the industrial sector of the united states", *Energy*, **76**, pp. 749–760 (2014).

42. Abedinia, O. and Amjady, N. "Short-term load forecast of electrical power system by radial basis function neural network and new stochastic search algorithm", *International Transactions on Electrical Energy Systems*, **26**(7), pp. 1511–1525 (2016).

43. Gajowniczek, K. and Zabkowsk, T. "Electricity forecasting on the individual household level enhanced based on activity patterns", *Plos One*, **12**(4), e0174098 (2017).

44. He, W. "Load forecasting via deep neural networks, Procedia Computer Science", **122**, pp. 308–314 (2017).

45. Shi, H., Xu, M., Ma, Q., et al. "A whole system assessment of novel deep learning approach on short-term load forecasting", *Energy Procedia*, **142**, pp. 2791–2796 (2017).

46. Rahman, A., Srikumar, V., and Smith, A.D. "Predicting electricity consump- tion for commercial and residential buildings using deep recurrent neural networks", *Applied Energy*, **212**, pp. 372–385 (2018).

47. Kong, W., Dong, Z.Y., Hill, D.J., et al. "Short-term residential load forecasting based on resident behaviour learning", *IEEE Transactions on Power Systems*, **33**(1), pp. 1087–1088 (2017).

48. Bedi, J. and Toshniwal, D. "Deep learning framework to forecast electricity demand", *Applied Energy*, **238**, pp. 1312–1326 (2019).

49. Elamin, N. and Fukushige, M. "Modeling and forecasting hourly electricity demand by sarimax with interactions", *Energy*, **165**, pp. 257–268 (2018).

50. Amezquita-Sancheza, J., Valtierra-Rodriguez, M., and Adeli, H. "Machine learning in structural engineering", *Scientia Iranica*, **27**(6), pp. 2645–2656 (2020).

51. Neshat, N., Sardarizarchi, M., and Mahlooji, H. "Application of deep learn- ing models based on fully-connected and recurrent neural networks tresidual peak load forecasting", *Sharif Journal of Industrial Engineering and Management*, **36**(1.2), pp. 103–111 (2020).

52. Francois Chollet, F., *Deep Learning with Python, Simon and Schuster*, ISSN: 1638352046 (2021).

Biographies

Najmeh Neshat, Assistant Professor at Meybod University (Industrial Engineering Group); interested in modeling and optimizing energy systems; reviewer of Elsevier Journals e.g., Applied Energy and Computational Methods in Science and Engineering; author of more than 15 related articles in peer-reviewed journals indexed in Thomson and JCR.

Mohsen Sardari Zarchi received his PhD in Computer Engineering, Artificial Intelligence from Isfahan University in 2015. He is currently working as an Assistant Professor at the Faculty of Computer Engineering, at Meybod University. His research interests include deep learning, image processing, artificial intelligence, and bioinformatics.

Hashem Mahlooji is a Professor of Industrial Engineering at the Sharif University of Technology. His research interests include the simulation of discrete event systems, Monte Carlo, and applied statistics.