ParDeeB: A Graph Framework for Load Forecasting Based on Parallel DeepNet Branches

Najmeh Neshat\textsuperscript{a,*}, Mohsen Sardari Zarchi\textsuperscript{b}, Hashem Mahlooji\textsuperscript{c}

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

Abstract

Recently, energy demand forecasting has emerged as a significant area of research because of its prominent impact on greenhouse gases (GHGs) emission and global warming. The problems of load forecasting are characterized by complex and nonlinear nature and also long-term historical dependency. Up to now, several approaches from statistical to computational intelligent have been applied in this research filed. The literature agrees with the fact that deep learning approach is more capable in dealing with these characteristics among existing approaches. However, the recent state-of-the-art deep network models are not robust against different historical dependency. 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 kind of networks can be incorporated. The parallel recurrent

\textsuperscript{*}Corresponding author. Tel.: +989133541567
Email addresses: neshat@meybod.ac.ir (Najmeh Neshat), sardari@meybod.ac.ir (Mohsen Sardari Zarchi), hmahlooji@sharif.edu (Hashem Mahlooji)
branches represent the historical dependency of determinants individually and this leads to better performance in case of different historical dependency in data. In this case study, the performance of the proposed model is examined through a comparison study with the state-of-the-art deep network models. The comparison resulted in that the proposed framework can improve the load forecasting by a significant margin on average.

**Keywords:** Load Forecasting, Deep Neural Networks, Parallel Deep Networks, Residential Load Demand

1. **Introduction**

The forecasting of energy demand is at the heart of energy management, which enables the suppliers to manage many frequent operation decisions. However, there are some difficulties with accurate forecasting due to its complicated and uncertain nature. In short-term energy demand forecasting, dependency on the time, consumer choice and the external weather make load forecasting a complex area of research with rare characteristics and important consequence.

Studies suggest that households are major contributors to total global energy demand [1, 2]. Hence, residual load forecasting is the center of attention in this area of research from the both academic and practical point of view. Different forecasting approaches, from statistical to computational intelligent have been applied in the past [3, 4]. Based on the number of techniques, the related literature is typically divided into two main categories: stand-alone and hybrid methods consisting of one and more than one technique respectively and based on the type of the underlying technique,
stand-alone methods are divided into three methods namely Statistical, Casual and Computational Intelligence methods.

Statistical methods model dynamic relationship between lagged values of determinants and forecast load demand based on historical data as follow: autoregressive (AR) and double seasonal Holt-Winter (DSHW) models [5], autoregressive with exogenous inputs (ARX) models [6], threshold ARX (TARX) models [7], generalized autoregressive conditional heteroscedasticity (GARCH) based models [8, 9, 10, 11], 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, 18, 19, 20]. The hybrid version of the mentioned methods also was suggested e.g., wavelet-based models [21, 12, 22].

Casual methods focus on formulating dynamics relationship between causal variables (determinants) and forecasted load demand. Causal models utilize least-square fitting method to extract forecasted load demand in terms of its determinants e.g., temperature, humidity and lagged data[23]. Different casual methods e.g., linear regression (LR) [24], non-linear regression (NLR) [25, 26, 27, 28, 29], Logistic or logit regression (LoR) [30] were widely utilized in the literature.

The real-world problems like load forecasting usually have nonlinear nature and a pitfall of them is that they are usually linear models as well; they might not perform well in data where there is high historical dependency pattern in data. Other words, they only can handle low data frequency e.g. weekly patterns, and the nonlinear behavior of load demand might become
too complicated to predict [31]. Besides, they are just utilized for short and medium forecasting term. However, their internal logic is very clear and they are called "White-box" methods.

Computational Intelligence methods such as Artificial Neural Network (ANN) and Support Vector Machine (SVM) are wildly used in this area of research considering their ability in handling hidden feature of data as well nonlinear modeling [32, 33, 34, 35, 36, 37, 38]. In addition, they are preferable for all temporal forecasting ranges. However, they also lack the ability to deal with high historical dependency pattern in data [39].

Ekonomou utilized ANN to forecast load for a country. Multi-Layer Perceptron (MLP) model is utilized to forecast load demand in terms of four available factors namely: weather conditions, historical demand data, GDP, and load capacity [40]. Comparison results obtained from the data of years 2012 and 2016 showed that MLP model outperforms the LR and SVM models. In the year 2014, Kialashaki et al. estimated energy demand for industrial sectors in terms of several determinants e.g., price of energy carriers and GDP using ANN and LR. Experimental results showed the superiority of the ANN model over LR based on accuracy and reliability indicators [41]. Abedinia et al. combined Radial neural network with stochastic search to forecast short-term load demand. The prediction results were validated by comparing with the results of the MLP network, echo-state network, and wavelet transform [42]. The similar results were obtained by Gajowniczek and Zabkowski [43].

In the context of load forecasting, He et al. proposed a DeepNet model to forecast short-term load demand. Convolution Neural Network (CNN) was
utilized to extract the features from historical patterns, which further forms the basis for load forecasting [44].

Shi et al. proposed a deep learning approach to estimate short-term electricity load. Deep RNNs were implemented to estimate demand at two different levels i.e. regional aggregate level and household dis-aggregate level. Based on the experimental results, the authors stated that Deep RNN performed better than the shallow neural network [45]. Further, 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 considering the appliance consumption sequences. The obtained results imply that load demand is highly dependent on residential behavior [47].

Recently, Bedi and Toshniwal developed the DeepNet, the RNN and the SVM models using load demand data of Union Territory Chandigarh, India [48]. To consider the dynamics of the load demand determinants, cluster analysis was performed on the data. The comparison resulted in the superiority of DeepNet framework for short-term load forecasting over RNN and SVM. In this study, investigating the various nonlinear exogenous determinants such as climate conditions and economic variables in order to analyze the trend of the load demand patterns was suggested as a future study. It should be noted that clustering the data based on season, day and interval data led to several DeepNet models. In fact, different periodic histories in data required clustering and led to several models for load forecasting.

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

- It is capable of capturing nonlinearity of data and modeling the complex nonlinear forecasting problems because of high generalization capability of computational intelligence models.

- In contrast to traditional ANN models, it is able to handle long term data dependency pattern because of various deep hidden layers.

- It is robust against different exogenous variables with different historical data in forecasting problems by incorporating dummy variables. To avoid the model becoming over parameterized, prior studies needed to cluster the data in terms of determinants e.g., the hour 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 one is that it is robust against different exogenous variables with different historical dependency using parallel multi-input branches.

To add up, we propose a graph framework based on parallel DeepNet branches to address the aforementioned characteristics. The paper is organized as follows: Section 2 introduces the preliminaries and state-of-the-art methods that are used in the research. Next, Section 3 describes the proposed DeepNet Modeling framework. Section 4 evaluates the base forecasters (that are extracted from the literature and considered in the benchmark) and the proposed model by considering the case study. The Numerical results are provided in Section 5. Finally, Section 6 concludes the paper and outlines the main results.
2. Preliminary

2.1. Load Forecasting

Short-term load demand depends on weather, customers daily and monthly demand patterns, and effects of varying weather conditions on these patterns. Previous studies have addressed to the weather elements i.e., temperature, solar insolation, humidity, and wind speed as exogenous variables [49]. Forecasting load demand typically consists of examining historical load data along with information about past, current, and predicted future of exogenous variables.

2.2. Deep Neural Networks

The term Deep Learning is usually used to refer 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 of which Dense Neural Network, Recurrent Neural Networks (RNNs), and Convolutional Neural Network (CNN) can be used as a tool for forecasting [50]. Dense or fully connected networks have a feed-forward structure in which each neuron is connected to all neurons of the next layer. Although dense networks can be used to build a nonlinear model, they are not suitable for models which depend on historical data [44]. In order to build a historical based model, RNNs are widely used. The RNN considers dependencies among data nodes that lead to capturing the historical dependency pattern in data. The dependencies are incorporated by persisting the knowledge accumulated from subsequent timestamps using network loops or feedback. An RNN unrolled in the time domain is shown in Figure 1. In this figure, \( I_t \) denotes input value.
at timestamp \( t \), \( S_t \) stands for the state at timestamp \( t \), \( O_t \) states output at timestamp \( t \). The current state \( S_t \) is computed in terms of current input \( I_t \) and previous hidden state \( S_{t-1} \). Mathematically it can be given as:

\[
S_t = f_\theta(UI_t + WS_{t-1}) \quad O_t = f_\alpha(VS_t)
\] (1)

RNNs are capable of dealing with short-term dependencies, but they fail to handle long-term dependencies because of vanishing gradient event (He, 2017). In order to tackle the aforementioned problem, Deep learning structure was developed. The two state-of-the-art deep learning models are Long Short-Term Memory (LSTM) and Gated recurrent units (GRUs) networks. Convolutional neural networks with 2D filters were first introduced for computer vision application. However, CNNs with 1D filter are used recently for time series or historical based model. Figure 2 shows a schematic of how the 1D filter can be used. In comparison with RNNs, CNNs are faster and needs less computational resources, however, RNNs such as LSTMs have the ability to build a historical based model.

2.3. Performance Metrics

The metrics used in evaluating the model fitness consist of the root mean square error (RMSE) and mean absolute percentage error (MAPE). These performance metrics are selected due to the wide use in the related literature and are also considered as the main metrics to establish the strength of the fitness. *Mean Absolute Percentage Error* The MAPE approach is common in the forecasting literature. However it is mathematically simple, but it has some limitations. The MAPE approach is not capable of dealing with the actual data with zero value. The effect of this limitation depends on the
percentage of actual data set being zero. Mathematically it can be presented as Eq (2).

$$MAPE = \frac{100}{N} \sum \frac{|y_i - \hat{y}_i|}{y_i}$$

(2)

Where $y_i$ represents the actual load demand and $\hat{y}_i$ represents the predicted load demand and $n$ indicates the number of values.

*Root Mean Square Error* RMSE is another common metric utilized in the evaluation of the accuracy of the fitness. RMSE provides a means of evaluating mean error considering load demand. This metric is important since methods such as MAPE do capture the accuracy in terms of the value of output load demand which is the main objective of the fitness and therefore can provide good accuracy. Mathematically it can be presented as Eq (3).

$$RMSE = \sqrt{\frac{1}{N} \sum (y_i - \hat{y}_i)^2}$$

(3)

3. The Proposed DNN Framework

To forecast day-ahead load demand, we proposed a novel Deep Neural Network framework based on Parallel DeepNet Branches. Therefore, we called our model ParDeeB. Each branch of ParDeeB is a subnetwork in which different kind of deep network such as RNNs, CNNs or Denses network can be used. These branches are concatenated as the acyclic graph to form the final structure. ParDeeB framework contains three major phases: i) Data preparation, ii) Multi-Input RNN branches, and iii) Non-historical based determinants.
3.1. Data preparation

In order to forecast load demand robustly, several aspects of input data should be considered. Let $D = \{X = [X_1, X_2, \ldots, X_n], Y = [y_1, y_2, \ldots, y_n]\}$ be the $n$ samples of input data where $X_t = x^1_t, x^2_t, \ldots, x^m_t$ includes $m$ determinants and $y_t$ is target value (actual load value). With the aim of presenting a periodical pattern appropriately, both historical and non-historical determinants should be considered in load forecasting. The historical determinants can be incorporated by using deep RNN. Therefore, in the first step of the proposed framework, $k$ previous timestamps of determinant $x^j_t$ are formed as sequential vector $S^j_t = x^j_{t-k}, x^j_{t-1}, \ldots, x^j_t$, where $k$ is the size of lookback (lagged data). If the sizes of lookbacks for all determinants are considered the same, the sequential input variables $X_{seq} = [S_1, S_2, \ldots, S_n]$ obtained from $X$ is a 3D tensor (matrix) where $S_t = [S^1_t, S^2_t, \ldots, S^m_t]$. Figure 3 depicts $X_{seq}$ when sizes of lookbacks of all determinants are the same.

According to the importance of variable history and its periodic type, the previous timestamps may be sampled with time step $d$. So the sequential vector is defined as $S^j_t = x^j_{t-kd}, x^j_{t-d}, \ldots, x^j_t$. For example, instead of considering the previous timestamp for each hour, we can sample only one value per 6 hours that means $d$ is set to 6. Since each variable has different periodic history, we prepared a sequence vector of each variable with its individual time step ($d$) and lookback ($k$). So, the sequence lengths of $S^j_{seq}$ are different and consequently, $X_{seq}$ is not a 3D tensor. Figure 4 depicts the overview of $X_{seq}$ when we have different step and lookback for determinants. Note that the $d$ and $k$ are hyper-parameters, which are set experimentally.
3.2. Multi-Input RNN branches

When the input of the neural network is in the form of tensor, designing the structure of the deep neural network is straightforward. In this case, the network can be implemented with a linear stack of layer sequentially as shown in Figure 5. Although the sequential model is so common, it is inflexible when the input data is not in the form of tensor e.g. Figure 4. Besides, in a forecasting model, each determinant might need a different type of neural networks such as CNNs, LSTMs or GRUs. To overcome this challenge, we innovatively proposed parallel deep branches in our framework to represent the history of each determinant individually. The parallel branches process their determinants by using RNN and Dense networks. Then the branches are merged together through concatenated and dense layer. Concatenated layer brings all the output from the previous branches and integrates them into the next layer. In fact, the concatenated layer, only connected different branches output into one layer. For example, Figure 6 shows a simple model with three branches which are merged with a concatenated layer.

3.3. Non-historical Based Determinants

Some input determinants are not historical based data. However, these determinants can have causal effect on the load demand. These casual determinants are usually model 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 the all RNN and dense branches, a directed acyclic graph topology is employed. In this framework which is called ParDeeB, the input is processed by several parallel branches. Then the branches outputs are concatenated in a jointed layer. Figure 7 shows the
overview of ParDeeB framework. Note that each branch consists of a dense layer and an RNN layer.

4. Experiment

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

4.1. Case Study

In order to evaluate the proposed DL framework and to evaluate the forecasting accuracy of the various base models, we use the data of peak load and weather condition (e.g., temperature and wind speed) of Shahrekord, Iran, in the period of 03/03/2015 to 03/03/2018 which are from [51]. The detailed statistical information of the exogenous variables is provided as Table I. According to the data and the literature, customers hourly, daily and monthly demand patterns vary because load value behaves differently on different holiday status; hour of night/day; weekday; and season. Figure 8 demonstrates the periodic pattern of load demand over the yearly time horizon in terms of daily intervals. Hence, dummy variables were defined to model the daily, weekly and monthly periodic effects. The 24 hours of the day were modeled by a dummy variable with three classes named low, moderate and high load type and in the same way seven days of the week modeled with seven classes. The season information was considered through incorporating two dummy variables representing the month number and the day of month. The overall goal of interval/daily characterization is that the proposed model to be capable of forecasting load for the all user-specified season, day and
time intervals of the day. Similarly, the intraday patterns of holidays are different from those of typical days; so, the binary dummy variable of Holiday was considered in the model. To add up, 6 dummy variables were added to model their effects on load demand pattern. Table I provides the statistical information of the used data (such as maximum, average, minimum and the peak load demand of days. In the models development phase, we divided the data into three sets: Training, Validation and Test sets. The first 75 percentage of samples (which are the samples of 1st, 2nd and 3rd years) were used for training and validating the different models and the rest (the samples of 4th year) were utilized to test the models performance. This study has 30768 samples with 23 determinants and one target which are recorded hourly. Considering that there are 24 electricity consumption values per day, the training and validation dataset comprises 23076 data points. Likewise, the test dataset comprises 7692 data points.

4.2. Framework implementation

In order to evaluate the ParDeeB framework, this framework should be implemented and evaluated in our case study. In ParDeeB implementation, Python 3.7 is used. This implementation is evaluated on a desktop PC with Geforce 1060 GTX GPU, i3−6100 CPU, 12GB RAM and Ubuntu Linux OS. The model use Keras 2.2 with TensorFlow 1.10 as backend for deep learning algorithms. In addition, others Python packages such as Scikit-lean, NumPy, Openpyxl, and Matplotlib are used for preprocessing and post-processing algorithm on the input data. The determinants values of the dataset have different scale variations, which can make training algorithm less efficient. Therefore, the dataset is normalized statistically. The data normalization on
each sample $X$ is performed as Eq (4):

$$X_{\text{new}} = \frac{X - \mu}{\sigma}$$

(4)

where $\mu$ and $\sigma$ are mean and standard deviation respectively.

5. Results and discussion

As we mentioned in section 3, ParDeeB framework can have parallel branches as many as are needed. Based on the nature of our case study, we considered three parallel branches. First and second branches are responsible for historical data, while the third branch presents non-historical data. The input of the first branch is a sequence of previous load demand values. The second branch gets 19 determinant sequences. For example the sequences of previous temperature, wind, etc. Finally, the third branch accepts 6 dummy determinants such as the value of the year, the month, etc. In order to find the best model configuration for our case study, different network types, time-steps and lookbacks are examined. Table 2 shows the five top configurations among experimented configurations. According to the experimental result, the best configuration can be obtained by using LSTM for the first branch, GRU for second branch and Dense for the third Branch. From this configuration, we can conclude that LSTM can model long-term historical data perfectly rather than GRU and CNN. So for the first branch in which previous load demand should be modeled, using LSTM with long sequence (lookback=168) and sampling all data ($d=1$) is the best configuration. Furthermore, we can see that other historical determinants need simpler models. Hence for the second branch GRU can perform well which
determinant sequences have time-step $d=24$ hours (means daily sampling) and lookback=7. The optimal selection for the third branch is a Dense model without any recurrent layers. This branch accepts only dummy determinants without considering previous 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. According to the experimental result, the best configuration can be obtained by using LSTM for the first branch, GRU for second branch and Dense for the third Branch. From this configuration, we can conclude that LSTM can model long-term historical data perfectly rather than GRU and CNN. So for the first branch in which previous load demand should be modeled, using LSTM with long sequence (lookback=168) and sampling all data ($d=1$) is the best configuration. Furthermore, we can see that other historical determinants need simpler models. Hence for the second branch GRU can perform well which determinant sequences have time-step $d=24$ hours (means daily sampling) and lookback=7. The optimal selection for the third branch is a Dense model without any recurrent layers. This branch accepts only dummy determinants without considering previous 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 hyper-parameters that their optimal values are obtained experimentally. The main hyper-parameters are Activation function, Regularization function, Dropout value, and Recurrent dropout value. The optimal valued of these hyper-parameters for the ParDeeB model which
implement according to Figure 9 is shown in Table 3. The dropout is used to overcome overfitting problem in training phase of the Neural Network. In the dropout procedure, some weights between layers connections are selected randomly and set to zero temporarily. However, in recurrent neurons like LSTM, it is also recommended to ignore randomly some recurrent values in each epoch. In Table 3, The dropout value refers to dropout between layers and, the recurrent dropout refers to dropout value of the LSTM and GRU neuron used in the model.

After explaining the experimental configuration selection and obtaining the optimal model topology, the accuracy of the models should be compared. In order to validate our forecasting model, ParDeeB was compared with four well-known types of the deep neural network 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 train and test data. From the accuracy results listed in Table 4, it is evident that ParDeeB does more reliable and accurate forecasting than other developed models.

Besides, the models were analyzed schematically for forecasting accuracy according to Figure 10. This figure depicts the diagram of the actual value and the predicted values for the first 100 test data. The x-axis and y-axis in Figure 10 represent the test data number and peak load demand in terms of Megawatt respectively. The red line remarked with 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

By emerging computational intelligent approach, significant advancements have been done for developing efficient and accurate load forecasting models. In this study, a novel graph framework for load forecasting based on parallel DeepNet branches was proposed and compared to the state-of-art models in this filed. This framework characterizes by handling long term data dependency pattern, capturing nonlinearity of data, and robustness against different exogenous variables with different historical data dependency using parallel multi-input branches. From the results and observations that are illustrated 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 the load forecasting by 3.09 percent on average. In details, the proposed model outperforms CNN model with a 10.84 percent improvement because it is able to handle long-term data dependency pattern with more hidden layers. It can be explained as while the proposed model is deep structure, the CNN model is less deep than the proposed one. An improvement in accuracy by 0.72 percent can be resulted from the proposed model over the Dense model by incorporating the recurrent mechanism. The recurrent parallel structure of the proposed model results in an improvement of 0.46 percent over the LSTM and GRU models. These parallel recurrent branches represent the history of each determinant individually when there are different periodic histories in data. To add up, in general, deep learning models are well suited
to the problem of load forecasting. However, some contributions in their configurations like parallel branches make them more accurate and robust.

References

[1] F. Amara, K. Agbossou, Y. Dubé, S. Kelouwani, A. Cardenas, S. S. Hosseini, A residual load modeling approach for household short-term load forecasting application, Energy and Buildings 187 (2019) 132–143.

[2] N. Neshat, An approach of artificial neural networks modeling based on fuzzy regression for forecasting purposes, International Journal of Engineering 28 (11) (2015) 1651–1655.

[3] A. Azadeh, Z. Faiz, A meta-heuristic framework for forecasting household electricity consumption, Applied Soft Computing 11 (1) (2011) 614–620.

[4] N. Neshat, M. R. Amin-Naseri, F. Danesh, Energy models: Methods and characteristics, Journal of Energy in Southern Africa 25 (4) (2014) 101–111.

[5] R. Kamranrad, A. Amiri, Robust holt-winter based control chart for monitoring autocorrelated simple linear profiles with contaminated data, Scientia Iranica 23 (3) (2016) 1345–1354.

[6] R. Weron, A. Misiorek, Forecasting spot electricity prices: A comparison of parametric and semiparametric time series models, International journal of forecasting 24 (4) (2008) 744–763.
[7] A. Misiorek, S. Trueck, R. Weron, Point and interval forecasting of spot electricity prices: Linear vs. non-linear time series models, Studies in Nonlinear Dynamics & Econometrics 10 (3).

[8] A. K. Diongue, D. Guegan, B. Vignal, Forecasting electricity spot market prices with a k-factor gigarch process, Applied energy 86 (4) (2009) 505–510.

[9] R. C. Garcia, J. Contreras, M. Van Akkeren, J. B. C. Garcia, A garch forecasting model to predict day-ahead electricity prices, IEEE transactions on power systems 20 (2) (2005) 867–874.

[10] C. R. Knittel, M. R. Roberts, An empirical examination of restructured electricity prices, Energy Economics 27 (5) (2005) 791–817.

[11] C. Wang, C. Rao, Y. Meng, X. Sun, Research on impact of shenzhen-hong kong stock connect mechanism on the volatility of stock markets, Scientia Iranica.

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

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

[14] K. Jeong, C. Koo, T. Hong, An estimation model for determining the annual energy cost budget in educational facilities using sarima (seasonal
autoregressive integrated moving average) and ann (artificial neural network), Energy 71 (2014) 71–79.

[15] Y. Wang, J. Wang, G. Zhao, Y. Dong, Application of residual modification approach in seasonal arima for electricity demand forecasting: A case study of china, Energy Policy 48 (2012) 284–294.

[16] F. J. Nogales, J. Contreras, A. J. Conejo, R. Espínola, Forecasting next-day electricity prices by time series models, IEEE Transactions on power systems 17 (2) (2002) 342–348.

[17] A. M. Mbae, N. I. Nwulu, Day-ahead load forecasting using improved grey verhulst model, Journal of Engineering, Design and Technology.

[18] A. Kazemi, M. Modarres, M. Mehregan, N. Neshat, A. Foroughi, 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, Vol. 12, 2011.

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

[20] X. Ma, W. Wu, Y. Wang, B. Zeng, W. Cai, Predicting primary energy consumption using ndgm (1, 1, k, c) model with simpson formula, Scientia Iranica.

[21] Z. Tan, J. Zhang, J. Wang, J. Xu, Day-ahead electricity price forecasting using wavelet transform combined with arima and garch models, Applied Energy 87 (11) (2010) 3606–3610.
[22] M. Shahriari-kahkeshi, Robust fault detection and isolation scheme using fuzzy wavelet network with a hybrid design algorithm, scientiairanica 24 (3) (2017) 1467–1481.

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

[24] K.-B. Song, Y.-S. Baek, D. H. Hong, G. Jang, Short-term load forecasting for the holidays using fuzzy linear regression method, IEEE transactions on power systems 20 (1) (2005) 96–101.

[25] M. Bilgili, B. Sahin, A. Yasar, E. Simsek, Electric energy demands of turkey in residential and industrial sectors, Renewable and Sustainable Energy Reviews 16 (1) (2012) 404–414.

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

[27] A. Kazemi, H. G. Shakouri, M. B. Menhaj, M. R. Mehregan, N. Neshat, A hierarchical artificial neural network for transport energy demand forecast: Iran case study, Neural Network World 20 (6) (2010) 761.

[28] N. Neshat, H. Hadian, S. R. Alangi, Technological learning modelling towards sustainable energy planning, Journal of Engineering, Design and Technology.
[29] N. Neshat, H. Hadian, M. Behzad, Nonlinear arimax model for long-term sectoral demand forecasting, Management Science Letters 8 (6) (2018) 581–592.

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

[31] G. Tsekouras, E. Dialynas, N. Hatziargyriou, S. Kavatza, A non-linear multivariable regression model for midterm energy forecasting of power systems, Electric Power Systems Research 77 (12) (2007) 1560–1568.

[32] A. Azadeh, N. Neshat, K. Rafiee, A. Zohrevand, 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) (2012) 221–240.

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

[34] A. Kazemi, G. Shakouri, M. Menhaj, M. Mehregan, N. Neshat, A multi-level artificial neural network for residential and commercial energy demand forecast: Iran case study, in: International Conference on Management Technology and Applications (ICMTA), 2010, pp. 24–28.

[35] A. Khwaja, X. Zhang, A. Anpalagan, B. Venkatesh, Boosted neural networks for improved short-term electric load forecasting, Electric Power Systems Research 143 (2017) 431–437.
[36] D. Niu, D. Liu, D. D. Wu, A soft computing system for day-ahead electricity price forecasting, Applied Soft Computing 10 (3) (2010) 868–875.

[37] B. Yao, J. Yao, M. Zhang, L. Yu, Improved support vector machine regression in multi-step-ahead prediction for rock displacement surrounding a tunnel, Scientia Iranica. Transaction A, Civil Engineering 21 (4) (2014) 1309.

[38] X. Qiu, Y. Ren, P. N. Suganathan, G. A. Amaratunga, Empirical mode decomposition based ensemble deep learning for load demand time series forecasting, Applied Soft Computing 54 (2017) 246–255.

[39] N. Maleki, A. Nikoubin, M. Rabbani, Y. Zeinali, Bitcoin price prediction based on other cryptocurrencies using machine learning and time series analysis, Scientia Iranica.

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

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

[42] O. Abedinia, N. Amjady, 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) (2016) 1511–1525.
[43] K. Gajowniczek, T. Bkowski, Electricity forecasting on the individual household level enhanced based on activity patterns, PLoS one 12 (4) (2017) e0174098.

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

[45] H. Shi, M. Xu, Q. Ma, C. Zhang, R. Li, F. Li, A whole system assessment of novel deep learning approach on short-term load forecasting, Energy Procedia 142 (2017) 2791–2796.

[46] A. Rahman, V. Srikumar, A. D. Smith, Predicting electricity consumption for commercial and residential buildings using deep recurrent neural networks, Applied energy 212 (2018) 372–385.

[47] W. Kong, Z. Y. Dong, D. J. Hill, F. Luo, Y. Xu, Short-term residential load forecasting based on resident behaviour learning, IEEE Transactions on Power Systems 33 (1) (2017) 1087–1088.

[48] J. Bedi, D. Toshniwal, Deep learning framework to forecast electricity demand, Applied energy 238 (2019) 1312–1326.

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

[50] J. Amezquita-Sancheza, M. Valtierra-Rodriguez, H. Adeli, Machine learning in structural engineering, Scientia Iranica.

[51] N. Neshat, M. Sardarizarchi, H. Mahlooji, Application of deep learning models based on fully-connected and recurrent neural networks to
residual peak load forecasting, Sharif Journal of Industrial Engineering & Management 36 (1.2) (2020) 103–111.

[52] C. Francois, Deep learning with python (2017).

Biographies

Najmeh Neshat, ASSISTANT PROFESSOR of University of Meybod (Industrial Engineering Group); INTERESTED IN Energy System Modeling and Optimization; REVIEWER of Elsevier journals e.g., Applied Energy and Computational Methods in Sciences and Engineering; AUTHOR of more than 15 related articles in peer-reviewed journals indexed in Thomson and JCR.

Mohsen Sardari Zarchi got his PhD in computer engineering, Artificial Intelligence from University of Isfahan in 2015. He is now working as an assistant professor at the Faculty of Computer Engineering, Meybod University. His research interests include Deep learning, Image processing, Artificial intelligence and Bioinformatics.

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

Figures and Tables
Figure 1: Architecture of an RNN network [51]

Figure 2: Performing 1D convolution filter [52]

Figure 3: An overview of $X_{seq}$ which is a 3D timestamps data tensor
Figure 4: An overview of $X_{seq}$ with different timestamp lengths

Figure 5: A sequential model: a linear stack of layer

Figure 6: A simple model with three parallel branches consisting of individual determinant timestamp vector
Figure 7: The overview of the proposed framework (ParDeeB)

Figure 8: Peak Load Demand Periodicity in terms of daily intervals
Figure 9: The overview of the optimal configuration of the ParDeeB model

Figure 10: The Result of the ParDeeB and other models for 100 samples compared to actual values

29
|                      | 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     |

Table I: The daily statistical data of the model inputs (exogenous and lagged data) and the related outputs.
| No. of Conf. | Branch 1               | Branch 2               | Branch 3               | MAPE Test Error |
|-------------|------------------------|------------------------|------------------------|-----------------|
| 1           | LSTM with $n = 64$     | GRU with $n = 128$    | Dense with $d = 1, l = 168$ | 6.54            |
| 2           | LSTM with $n = 64$     | LSTM with $n = 64$    | Dense with $d = 24, l = 7$ | 6.71            |
| 3           | LSTM with $n = 128$    | CNN with $n = 32$     | Dense with $d = 24, l = 14$ | 7.02            |
| 4           | GRU with $n = 256$     | LSTM with $n = 64$    | Dense with $d = 24, l = 7$ | 7.12            |
| 5           | LSTM with $n = 64$     | GRU with $n = 64$     | Dense with $d = 24, l = 14$ | 7.14            |

Table II: The results of different configuration where the symbols $n$, $d$, and $l$ refer to the number of neurons, time-step and lookback values respectively.

| 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 III: The optimal valued of these hyper-parameters for the ParDeeB model.
| 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           |

Table IV: The accuracy comparison of developed models based on performance metrics