Deep Neural Networks for Energy Load Forecasting

Kasun Amarasinghe, Daniel L. Marino, Milos Manic
Department of Computer Science
Virginia Commonwealth University
Richmond, Virginia, USA
amarasinghek@vcu.edu, marinodl@vcu.edu, misko@ieee.org

***Abstract*—Smartgrids of the future promise unprecedented flexibility in energy management. Therefore, accurate predictions/forecasts of energy demands (loads) at individual site and aggregate level of the grid is crucial. Despite extensive research, load forecasting remains to be a difficult problem. This paper presents a load forecasting methodology based on deep learning. Specifically, the work presented in this paper investigates the effectiveness of using Convolutional Neural Networks (CNN) for performing energy load forecasting at individual building level. The presented methodology uses convolutions on historical loads. The output from the convolutional operation is fed to fully connected layers together with other pertinent information. The presented methodology was implemented on a benchmark data set of electricity consumption for a single residential customer. Results obtained from the CNN were compared against results obtained by Long Short Term Memories LSTM sequence-to-sequence (LSTM S2S), Factored Restricted Boltzmann Machines (FCRBM), "shallow" Artificial Neural Networks (ANN) and Support Vector Machines (SVM) for the same dataset. Experimental results showed that the CNN outperformed SVR while producing comparable results to the ANN and deep learning methodologies. Further testing is required to compare the performances of different deep learning architectures in load forecasting.**

Keywords— Deep Learning; Deep Neural Networks; Convolutional Neural Networks; Building Energy; Energy; Artificial Neural Networks

I. INTRODUCTION

Smart grids demand an unprecedented level of flexibility in the way energy is generated and distributed to minimize energy wastage and optimize usage [1], [2], [3]. Therefore, the power grid has to be able to dynamically adapt to changes in demand and efficiently distribute the generation. Further, the grid controller should have the capability of efficiently handing the distributed generation from various sources such as renewables [4]. Therefore, intelligent control decisions should be made in real-time at aggregate level as well as at individual component level. Therefore, the ability of forecasting the future demands accurately is imperative [2], [5].

Forecasting the demand of individual component is as important as forecasting the aggregate demand in several fronts. In terms of demand response, individual component/ building level demand forecasting enables responding to local demand with local generations [2]. In terms of future uncertainty mitigation, demand/load forecasting at aggregate and individual component level, plays a crucial role [2]. Further, apart from being a major energy consumer by accounting for 20%-40% of the total worldwide energy production [6], [7], [8], [9] buildings are shown to account for a significant portion of the total energy wastage [10]. Accurate forecasts or predictions of the energy demand at building level can help optimize energy usage and minimize wastage at building level. The advent of smart meters have made the energy consumption data available [11]. Thus, data driven and statistical modeling have been made possible [3].

Load forecasting is subcategorized in to three categories: 1) Short-term load forecasting (one hour to one week) 2) Medium-term load forecasting (week to a year) and 3) Long-term load forecasting (longer than a year) [2]. Regardless of the category, load forecasting has been shown to be a difficult problem. In that, individual building level load forecasting has been shown to be a harder task than aggregate load forecasting [2], [12]. Two major approaches exist in literature for performing energy load forecasting: 1) Physics principles based forecasting and 2) Statistical and machine learning based forecasting. This paper focuses only on the second approach. In literature, many machine learning approaches have been explored. In [3] , Artificial Neural Network (ANN) ensembles were used to perform the building level load forecasting. ANNs have been explored in detail for the purpose of all three categories of load forecasting [5] , [13], [14], [15], [16]. In [17], support vector machines coupled with empirical mode decomposition was used to perform long term load forecasting. Kernel based multi task learning methodologies were used for electricity demand prediction in [18]. In [12], individual household electricity loads are modeled using sparse coding to perform medium term load forecasting. Apart from the methods enumerated above, a multitude of different methods are proposed in literature for solving the load forecasting problem. In the interest of brevity, an exhaustive coverage of the presented methods is not provided in this paper. Readers are referred to [19], [20] and [4] for comprehensive surveys of different techniques used for load forecasting. However, despite the extensive research, individual site/building level load forecasting remains to be a difficult problem.

This paper investigates the possibility of using deep learning for performing individual building level load forecasting. Multi layered architecture of deep learning allows the learning process to be carried out with multiple layers of abstraction. Thus, deep learning architectures allow the learning structure to learn complex relationships between inputs and outputs and complex patterns in data. Thus, deep learning has revolutionized a range of fields such as speech recognition and computer vision. A comprehensive overview and a review of deep learning methodologies can be found in [21].

In previous work for load forecasting using deep learning, authors of [2] performed building level load forecasting using

978-1-5090-1412-5/17/$31.00 ©2017 IEEE

Conditional Restricted Boltzmann Machines (CRBM) [22] and Factored Conditional Restricted Boltzmann Machines (FCBRM) [23]. Authors compared the two methods to several traditional methods including Support Vector Machines and traditional or "shallow" Artificial Neural Networks (ANN). Their experimental results showed that the FCRBM method outperformed the other tested methodologies. In another deep learning approach, the authors use Deep Belief Networks for performing short-term load forecasting on a Macedonian hourly electricity consumption dataset [24]. They compare the results to an ANN and conclude that DBN significantly outperforms the ANN on the tested dataset. In [25], the authors present an ensemble method combining Deep Belief Networks and Support Vector Machines and present load forecasting as a test case of the presented deep learning based regression methodology.

In our previous work, we investigated the effectiveness of using Long Short Term Memory (LSTM) based algorithms for building level load forecasting [26]. Two variants of the LSTM architecture were investigated: 1) standard LSTM architecture and 2) LSTM based sequence-to-sequence architecture. The same dataset which was used in [2] with the same train and test split was used for comparing results. It was shown that the sequence to sequence architecture performed well and produced comparable results with the FCRBM algorithm in [2].

This paper presents a continuation of the work of load forecasting using deep learning by exploring a different deep learning technique for performing building level energy load forecasting. The presented methodology uses Convolutional Neural Networks (CNNs) to perform the forecasting. In the presented work, multiple convolutional layers are used on historical load data before performing the final regression task. The presented CNN based load forecasting methodology was tested on a benchmark dataset which contained electricity consumption data for a single residential customer with time resolution of one hour. In order to compare results with the earlier work in deep learning, the same dataset used in [2] and [26] was used. To benchmark the performance, the same forecasting process was carried out using standard "shallow" Artificial Neural Networks (ANNs) and Support Vector Machines (SVM). Therefore, performance of CNN was compared against ANN, SVM, LSTM, LSTM-S2S and FCRBM. In addition to comparing with other algorithms, several architectures of CNNs were tested.

The rest of the paper is organized as follows. Section II provides background on Convolutional Neural Networks. Section III elaborates the presented energy load forecasting methodology using CNNs. Section IV elaborates the experimental setup. Section V presents the experimental results. Finally, section VI concludes the paper.

II. Convolutional Neural Networks

This section provides a brief overview on Convolutional Neural Networks (CNNs).

CNNs are a special type of neural network, which is mainly used for processing data with a grid topology [27]. For instance, images can be viewed as 2D grids and time series data such as energy consumption data, can be viewed as 1D grids. CNNs have been used successfully literature mainly for computer vision tasks such as image classification [21], [28]. CNNs use a specialized linear operation named convolution in at least one of the layers in the network.

Convolution is defined as an operation on two functions on real valued arguments [28]. The convolution operation is denoted with an asterisk.

$$s = (x * w) \tag{1}$$

where x denotes the input function, w denotes the weighting function. In the context of CNNs, the weighting function is called a "kernel". The output of the convolution operation is often called the "feature map" (denoted by s).

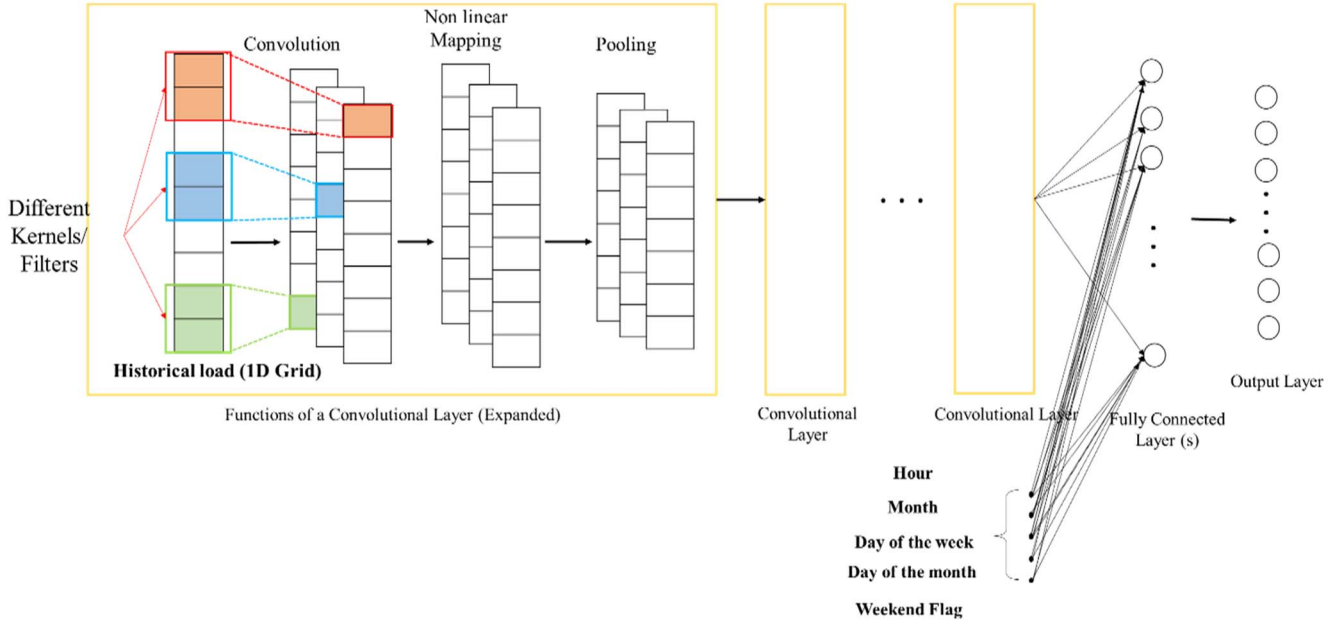

Fig. 1: Convolutional Neural Network architecture employed for energy load forecasting

The convolution operation is usually applied to inputs represented in multidimensional arrays. Further, the kernel is also a multidimensional array of weights that is changed as the algorithm learns through the iterations. Therefore, in a general case, since the inputs and kernels are multidimensional, the convolution operation is applied to more than one dimension. Thus, the convolution operation for a two dimensional input can be expressed as:

$$s(i,j) = (I * K)(i,j) = \sum_{l}\sum_{m} I(l,m)K(i+l, j+m) \quad (2)$$

where I represents a two dimensional input, K represents a two dimensional kernel. S is the resulting feature map after the convolution.

Each convolutional layer is comprised of three phases. The first phase performs the above described convolution operation and produces a feature map. Then, the elements of the feature map is run through a nonlinear activation function. The rectified linear activation function [29] is usually used in this stage [28]. Finally, a pooling function is used in the third stage to further modify and smoothen the feature map. The pooling operation makes the representation less susceptible to small variations in the input. Out of various pooling techniques, in the presented work, max pooling method is used. In max pooling, the operation returns the maximum value of a predefined rectangular neighborhood. Other pooling techniques such as average pooling, min pooling and weighted average pooling have been used in literature [28].

As mentioned, the designed network can consist of one or more convolutional layers. Once the convolutional layer(s) produce their outputs, the output is sent to one or more fully connected layers. Fully connected layers can be thought of as hidden layers in a standard neural network. The output layer is placed after the fully connected layers. The output layer performs a similar function to an output layer in a standard ANN. Learning process of the CNN is carried out using back propagation.

III. Convolutional Neural Networks for Energy Load Forecasting

This section elaborates the presented methodology for energy load forecasting using Convolutional Neural Networks

The objective of the presented load forecasting methodology is to estimate the energy load for a time step or multiple time steps in the future, given historical electricity load data. For purpose of simplicity, in the presented work, the time step between two measurements are assumed constant.

Let historical energy load data for M time steps be available. Then, the vector of available historical energy load data can be expressed as:

$$\vec{y} = \{y_{[0]}, y_{[1]}, y_{[2]}, y_{[3]}, \ldots, y_{[M-1]}\} \quad (3)$$

where $y_{[t]}$ represents the actual energy load measurement for the time step t. The forecasting algorithm should predict the energy load for the next N time steps. The vector of predicted N time steps can be expressed as:

$$\vec{\hat{y}} = \{\hat{y}_{[M]}, \hat{y}_{[M+1]}, \hat{y}_{[M+2]}, \hat{y}_{[M+3]}, \ldots, \hat{y}_{[M+N-1]}\} \quad (4)$$

where $\hat{y}_{[t]}$ represents the forecasted/ predicted energy load for the time step t.

In performing the energy load prediction, a pre-defined number of historic energy load measurements are fed into the convolutional layers of the CNN. Since energy load measurements are time series data, 1-D convolutions are performed on the data. As mentioned in the previous section, three stages of the convolution operation are applied to the historical data. Therefore, the input vector to the convolutional layers takes the same form as Eq (3).

Once the convolutional operations are performed, they are forwarded to a set of fully connected layers. In this stage, in addition to the historic energy load measurement data, information about the date and time of the first prediction is fed into the CNN as inputs. The input vector for the fully connected layers of the CNN can be expressed as:

$$\vec{i}_{fc} = \{\vec{o}_c, h_{[M]}, m_{[M]}, dw_{[M]}, dm_{[M]}, wf_{[M]}\} \quad (5)$$

where $\vec{i}_{fc}$ is the input vector for the fully connected layer, $\vec{o}_c$ is the output from the convolutional layers, $h_{[M]}$, $m_{[M]}$, $dw_{[M]}$, $dm_{[M]}$ and $wf_{[M]}$ represent the hour, month, day of the week, day of the month and a flag which is set if it's a weekend for the time stamp of the first prediction. I.e. if historical energy load data for 10 previous time steps are used as inputs to make the prediction of next time stamps, time stamp data for the 11th time step is used as the input for the hidden layer. Fig. 1 shows the proposed CNN architecture for energy load forecasting.

Standard back propagation is used for training of the model. A gradient descent model such as Stochastic Gradient Decent (SGD) is used to perform the optimization.

The loss function for the optimization can be expressed as:

$$L = \sum_{t=1}^{N}\left(y_{[t]} - \hat{y}_{[t]}\right) \quad (6)$$

where L is expressed as the total prediction error for N future time steps.

IV. Experimental Setup

This section elaborates the experimentation process of the presented methodology. First, the used dataset is introduced. Then, the specifics of the implementation are discussed.

A. Dataset

The presented CNN based energy load forecasting methodology was implemented on a benchmark dataset of

TABLE I: TESTED CNN ARCHITECTURES

Test case	Filters	Kernel Sizes	Pooling filters	Hidden layers
1	[4]	[[10]]	[[2]]	[20,20]
2	[4]	[[10]]	[[5]]	[20,20]
3	[4]	[[20]]	[[2]]	[10,10]
4	[10]	[[20]]	[[2]]	[10,10]
5	[5, 5]	[[10], [5]]	[[3], [3]]	[20,20]
6	[4, 8, 8]	[[10],[10],[6]]	[[2], [2], [2]]	[20,20]
7	[4, 4, 8]	[[10], [5],[2]]	[[1], [1], [2]]	[10]
8	[4, 8, 16, 16, 32]	[[3], [3], [3], [3], [3]]	[[2], [2], [2], [2], [2]]	[20,20]

electrical consumption. The used dataset contains electrical consumption data for a single residential customer named "Individual household electric power consumption dataset" [30]. The used dataset contains power consumption data for the period of four years (December 2006 – November 2010) with one-minute resolution. The dataset contains aggregate active power and three sub-metering measurements for three separate sections of the house. In the presented work, only the aggregate active power data are used.

The dataset contains 2075259 active power measurements in the aforementioned range of dates. Even though the dataset contained one-minute resolution data, the study was conducted in one-hour resolution data. One-hour resolution data were obtained by averaging the one-minute resolution data. Therefore, the dataset was reduced to 34608 records. The dataset was split in to training and testing sets. In [2] and [26], first three years were used for training and the last year data were used for testing. For comparison, the same train test split was used. In addition, to get generalized performance, training was carried out using k-fold cross validation as well.

B. Implementation Details

As mentioned, the CNN based load forecasting algorithm was implemented on one-hour resolution data. The presented methodology was implemented to perform a forecast for the next 60 hours. I.e. N in Eq. (4) was set to 60. In order to perform the said prediction, historic power consumption data for the 60 immediate previous hours were fed in the CNN based methodology. I.e. M was set to 60 in Eq. (3).

Therefore, 60 inputs were fed into the convolutional layers of the CNN. Since the inputs were time series data, it was considered to be data with the form of a 1D grid. Therefore, the Eq. (2) presented in Section II has to be modified for a 1D input. Further the kernel used in the convolution layers are defined as 1 dimensional kernels. Fig. 1 shows the architecture of the designed CNN. In the implemented CNN, each of the convolutional layers were designed to have the above mentioned three phases; 1) convolution phase, 2) non-linear transformation and 3) pooling (subsampling) phase. As mentioned, the convolution phase for the three layers were carried out with 1D kernels. Non-linear transformation for all convolutional layers was carried out using the linear rectifier unit (ReLU) activation function. The pooling phase for all convolutional layers was performed using max-pooling. Once the convolutional layers produced their output, it was sent to a fully connected layer(s) (hidden layer (s)). In this experiment, two hidden layers with 20 neurons each were used for all the test cases. The hidden layer used the ReLU function as its activation function. Since there were 60 outputs, the output layer contained 60 neurons. The output layer contained a linear activation function to produce outputs. Different CNN architectures with different convolutional layers, kernel sizes and pooling filter sizes were tested. Table I summarizes the different CNN architectures for each test case. For all architectures, for training, ADAM [31] algorithm was used as the gradient based optimizer. For all the test cases the same training and testing data were used.

In order to provide a benchmark, the load forecasting process was carried out using a standard feed forward "shallow" ANN and Support Vector Machines (SVM). Historical energy load data for 60 previous time steps together with the same time step

TABLE II: TESTING ERROR COMPARISON WITH OTHER ALGORITHMS

Algorithm	RMSE (Specific Training/Testing for Comparison)	RMSE With Cross Validation
CNN	0.677	0.732
LSTM – S2S [24]	0.625	N/A
FCRBM [2]	0.663	N/A
SVM	0.814	0.895
ANN	0.691	0.736

TABLE III: ERRORS SUMMARY FOR DIFFERENT CNN TEST CASES (CROSS VALIDATION)

CNN Test Case	RMSE (Testing)
1	**0.732**
2	0.734
3	0.746
4	0.766
5	0.744
6	0.737
7	0.757
8	0.756

data sent to the CNN were used as inputs to ANN and SVM. All three algorithms were implemented with K-fold cross validation

V. Experimental Results

This section elaborates results obtained from the conducted experiments. As mentioned, the dataset was split in to a training set and a testing set. The training set contained data for the first three years and the testing set contained the data for the final year. This specific train/testing split was used to follow the same procedure used in [2] and [26] so that results can be compared. It was noticed that this specific split resulted in testing errors, which were less than the training errors. Even though this was an unusual observation, the specific split had to be used to be able to compare with the previous work done with DNN. In addition to the specific split, K-fold cross validation was performed for CNNN, ANN and SVM.

Table II summarizes the results obtained from all the algorithms for testing sets in terms of RMSEs. It can be seen that all the DNN architectures perform better than the SVM but produced similar results to ANN. It was seen that the CNN slightly outperformed the FCRBM but not the LSTM-S2S presented in [26]. ANN performed better than the SVM but couldn't outperform any of the deep architectures.

As mentioned, it was noticed that when the specific training testing split was used, the testing error was lower than the training error. Therefore, it was important to get a more general idea of the algorithms performance. Table II presents the results obtained by 4-fold cross validation. It can be seen that when cross validation was performed, the results are not as good as the specific split. Cross-validated results were not available for the LSTM and the FCRBM.

Fig.2 (a) and Fig.2 (b) depicts two samples from CNN predictions in the training and testing sets respectively. Figures show samples where the actual measurement is compared with the CNN estimation. In both training and testing it can be been that the CNN is able to follow the general trend of data showing capabilities of generalization. The superior generalization capability of CNN is shown in the cross validated results as well.

In addition to comparing the presented method with other algorithms, several architectures of the CNN was tested as mentioned in Section IV (see Table I). As mentioned, eight different test cases were tried and it was seen that the much variation of the performance wasn't shown, in training or testing, regardless of the architecture. Table III presents the results obtained over all the test cases. It was seen that Test Case 6 (3 convolutional layers) managed to produce the lowest training error while Test Case 1 (1 convolutional layer) managed to produce the lowest testing error.

However, the method needs to be tested on several different datasets and many architectures that are more different to accurately analyze effect of having multiple convolutional layers and different kernel sizes on the forecasting/ prediction accuracy. Further, all these methods need to be extensively tested on several datasets to be able to provide accurate comparisons on their performance.

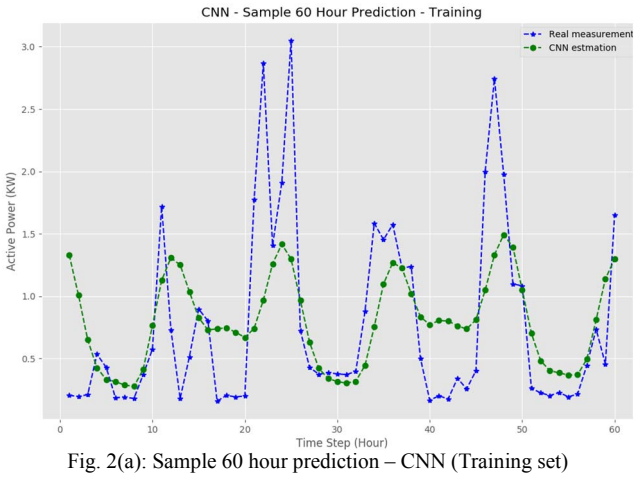

Fig. 2(a): Sample 60 hour prediction – CNN (Training set)

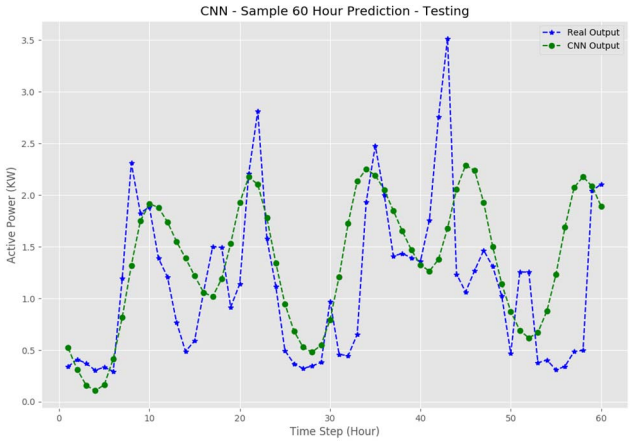

Fig. 2(b): Sample 60 hour prediction – CNN (Testing Set)

VI. Conclusion and Discussion

This paper investigated the effectiveness of using Convolutional Neural Networks for individual building level energy load forecasting. The work was an extension of our work on deep learning based energy load forecasting in [26]. The presented method was implemented on a benchmark dataset of electricity consumption data for one residential customer. For comparison, load forecasting was carried out using ANN and SVM. The results obtained by CNN was compared to ANN, SVM and other deep architectures LSTM and FCRBM. In addition to the algorithm comparisons, testing was carried out with different CNN architectures that consisted of different convolutional layers. It was noticed that results didn't vary much across the different architectures. The presented CNN managed to produce comparable results to the FCRBM presented in [2] and Sequence to Sequence LSTM presented by us in [26] on the same dataset. Therefore, it can be concluded that CNN remains to be a viable candidate for producing accurate load forecasts. However, it needs to be further tested with different datasets to validate the performance. In addition to these methods, authors in [24] and in [25] presented Deep Belief Networks for load forecasting. In order to compare the effectiveness of all these algorithms they need to be tested on different real world datasets. Further, majority of the work (except [24]) do not use weather data as an input to their work. It is important to add the

weather data as inputs to the learning algorithms as it has a direct relationship with the energy consumption. As future work, we plan to compare the results of the above algorithms with weather data to find the effectiveness of deep learning algorithms in load forecasting. In addition, short term, medium term and long term forecasting will be carried out with different methods to check their effectiveness.

REFERENCES

[1] C. Clastres, "Smart grids: Another step towards competition, energy security and climate change objectives," *Energy Policy*, vol. 39, no. 9, pp. 5399–5408, Sep. 2011.

[2] E. Mocanu, P. H. Nguyen, M. Gibescu, and W. L. Kling, "Deep learning for estimating building energy consumption," *Sustainable Energy, Grids and Networks*, vol. 6, pp. 91–99, Jun. 2016.

[3] J. G. Jetcheva, M. Majidpour, and W.-P. Chen, "Neural network model ensembles for building-level electricity load forecasts," *Energy and Buildings*, vol. 84, pp. 214–223, Dec. 2014.

[4] P. Siano, "Demand response and smart grids—A survey," *Renewable and Sustainable Energy Reviews*, vol. 30, pp. 461–478, Feb. 2014.

[5] C. Roldán-Blay, G. Escrivá-Escrivá, C. Álvarez-Bel, C. Roldán-Porta, and J. Rodríguez-García, "Upgrade of an artificial neural network prediction method for electrical consumption forecasting using an hourly temperature curve model," *Energy and Buildings*, vol. 60, pp. 38–46, May 2013.

[6] L. Pérez-Lombard, J. Ortiz, and C. Pout, "A review on buildings energy consumption information," *Energy and Buildings*, vol. 40, no. 3, pp. 394–398, 2008.

[7] K. Amarasinghe, D. Wijayasekara, H. Carey, M. Manic, D. He, and W. P. Chen, "Artificial neural networks based thermal energy storage control for buildings," in *IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society*, 2015, pp. 005421–005426.

[8] D. Wijayasekara and M. Manic, "Data-fusion for increasing temporal resolution of building energy management system data," in *IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society*, 2015, pp. 004550–004555.

[9] M. Manic, D. Wijayasekara, K. Amarasinghe, and J. J. Rodriguez-Andina, "Building Energy Management Systems: The Age of Intelligent and Adaptive Buildings," *IEEE Industrial Electronics Magazine*, vol. 10, no. 1, pp. 25–39, Mar. 2016.

[10] S. Naji *et al.*, "Estimating building energy consumption using extreme learning machine method," *Energy*, vol. 97, pp. 506–516, Feb. 2016.

[11] M. Manic, K. Amarasinghe, J. J. Rodriguez-Andina, and C. Rieger, "Intelligent Buildings of the Future: Cyberaware, Deep Learning Powered, and Human Interacting," *IEEE Industrial Electronics Magazine*, vol. 10, no. 4, pp. 32–49, Dec. 2016.

[12] C. N. Yu, P. Mirowski, and T. K. Ho, "A Sparse Coding Approach to Household Electricity Demand Forecasting in Smart Grids," *IEEE Transactions on Smart Grid*, vol. 8, no. 2, pp. 738–748, Mar. 2017.

[13] M. Q. Raza and Z. Baharudin, "A review on short term load forecasting using hybrid neural network techniques," in *2012 IEEE International Conference on Power and Energy (PECon)*, 2012, pp. 846–851.

[14] M. D. Felice and X. Yao, "Short-Term Load Forecasting with Neural Network Ensembles: A Comparative Study [Application Notes]," *IEEE Computational Intelligence Magazine*, vol. 6, no. 3, pp. 47–56, Aug. 2011.

[15] R. Kumar, R. K. Aggarwal, and J. D. Sharma, "Energy analysis of a building using artificial neural network: A review," *Energy and Buildings*, vol. 65, pp. 352–358, Oct. 2013.

[16] S. M. Sulaiman, P. A. Jeyanthy, and D. Devaraj, "Artificial neural network based day ahead load forecasting using Smart Meter data," in *2016 Biennial International Conference on Power and Energy Systems: Towards Sustainable Energy (PESTSE)*, 2016, pp. 1–6.

[17] L. Ghelardoni, A. Ghio, and D. Anguita, "Energy Load Forecasting Using Empirical Mode Decomposition and Support Vector Regression," *IEEE Transactions on Smart Grid*, vol. 4, no. 1, pp. 549–556, Mar. 2013.

[18] J. B. Fiot and F. Dinuzzo, "Electricity Demand Forecasting by Multi-Task Learning," *IEEE Transactions on Smart Grid*, vol. PP, no. 99, pp. 1–1, 2016.

[19] A. K. Singh, Ibraheem, S. Khatoon, M. Muazzam, and D. K. Chaturvedi, "Load forecasting techniques and methodologies: A review," in *2012 2nd International Conference on Power, Control and Embedded Systems*, 2012, pp. 1–10.

[20] L. Hernandez *et al.*, "A Survey on Electric Power Demand Forecasting: Future Trends in Smart Grids, Microgrids and Smart Buildings," *IEEE Communications Surveys Tutorials*, vol. 16, no. 3, pp. 1460–1495, Third 2014.

[21] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[22] V. Mnih, H. Larochelle, and G. E. Hinton, "Conditional Restricted Boltzmann Machines for Structured Output Prediction," *arXiv:1202.3748 [cs, stat]*, Feb. 2012.

[23] G. W. Taylor, G. E. Hinton, and S. T. Roweis, "Two Distributed-State Models For Generating High-Dimensional Time Series," *Journal of Machine Learning Research*, vol. 12, no. Mar, pp. 1025–1068, 2011.

[24] A. Dedinec, S. Filiposka, A. Dedinec, and L. Kocarev, "Deep belief network based electricity load forecasting: An analysis of Macedonian case," *Energy*, vol. 115, Part 3, pp. 1688–1700, Nov. 2016.

[25] X. Qiu, L. Zhang, Y. Ren, P. N. Suganthan, and G. Amaratunga, "Ensemble deep learning for regression and time series forecasting," in *2014 IEEE Symposium on Computational Intelligence in Ensemble Learning (CIEL)*, 2014, pp. 1–6.

[26] D. L. Marino, K. Amarasinghe, and M. Manic, "Building Energy Load Forecasting using Deep Neural Networks," *arXiv:1610.09460 [cs]*, Oct. 2016.

[27] Y. Bengio, Y. Lecun, and Y. Lecun, *Convolutional Networks for Images, Speech, and Time-Series*. 1995.

[28] I. J. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[29] X. Glorot, A. Bordes, and Y. Bengio, "Deep Sparse Rectifier Neural Networks," in *PMLR*, 2011, pp. 315–323.

[30] M. Lichman, "UCI Machine Learning Repository." 2013.

[31] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv:1412.6980 [cs]*, Dec. 2014.