2020 IEEE PES/IAS PowerAfrica

Short-Term Load Forecasting for Commercial Buildings Using 1D Convolutional Neural Networks

Abraham Kaligambe, Goro Fujita
Power System Laboratory
Shibaura Institute of Technology
Tokyo, Japan

***Abstract*—Many Commercial Buildings have employed smart meters to measure load consumption data at real-time intervals and then utilized by the Energy Management System (EMS). Load Forecasting based on historical load data is of key importance for effective operation, planning, and optimization of energy for Commercial Buildings. However, designing an accurate Load Forecasting Model is still an on-going challenge. Our methodology involved the usage of Deep Neural Networks (DNN) for Short-Term Load Forecasting. A special architecture of 1-Dimensional Convolutional Neural Networks (1D CNN) known as WaveNet was employed in our method because of its ability to extract rich features from historical load data sequences. A benchmark load consumption dataset of a Commercial Building for the fiscal year 2017 in Kyushu-Japan was used as a case study. Our model was evaluated and compared to other Machine Learning techniques for Forecasting. When tested on the same dataset, it outperformed them all.**

***Index Terms*—1D Convolutional Neural Networks, Commercial Buildings, Deep Neural Networks, Energy Management System, Short-Term Load Forecasting.**

I. Introduction

All over the world, many Commercial Buildings are integrating renewable energy, battery storage, and electric vehicles to their grid-connected Energy Management Systems as illustrated in Fig. 1. The economic viability and environmental benefits of this system are discussed in [1] and [2] respectively.

With the increasing improvements in technology, many electrical devices are becoming more affordable and smart. One such device is a Smart Meter. It can capture electrical load consumption data in timely intervals. Having access to plenty of historical load data in this Artificial Intelligence (AI) era can be very beneficial to Commercial Buildings. Load Forecasting based on historical load data is one of the benefits of utilizing the data from Smart Meters.

Short-Term Load and Solar PV Forecasting are of utmost importance for effective planning and energy optimization. Fig. 1 is a representation of the electrical system set up for the Commercial Building whose dataset we employed in our research. Accurate Load Forecasting is vital for their energy optimization and Electic Vehicle charging system.

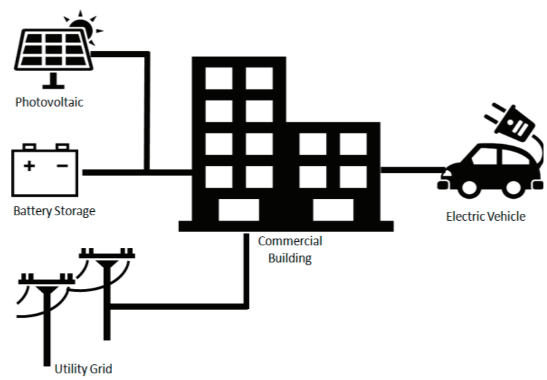

Figure 1. Commercial Building with EV and PV system

There are plenty of approaches and techniques for Load Forecasting. Some of the methods employ statistical methods. Reference [3] uses Autoregressive Integrated Moving Average (ARIMA) for a practical short term load forecasting in Spain. However, holidays are neglected. Reference [4] used a Random Forest Machine Learning model to combine many features for one day ahead load forecasting. However, only one day ahead forecasting is insufficient for some cases.

Nowadays, there is a lot of attention paid to Artificial Neural Networks (ANN) due to their success in many fields like speech and image recognition. Reference [5] showed the superiority of Long Short Term Memory(LSTM)-Recurrent Neural Networks(RNN) over other Machine learning techniques. Reference [6] and [7] used the combination of Convolutional Neural Networks (CNN) and Long Short Term Memory(LSTM) to extract rich features from historical load data. Reference [8] analyzes various methods to forecast energy consumption on national holidays.

In this paper, we investigate the usage of a Deep Neural Network technique using 1D CNNs with multiple layers stacked together to form a WaveNet structure. WaveNet is mostly popular for its application on processing raw audio as described in [9] and [10]. Our model is carefully modeled to

978-1-7281-6746-6/20/$31.00 ©2020 IEEE

suit Commercial Buildings. We, therefore, put emphasis not only on forecasting accuracy but also on the ability to predict load consumption on weekends, national and company holidays. We also considered usage of minimal historical load consumption data for model training to forecast hourly energy consumption up to a week ahead.

The rest of the paper is organized as follows; Section 2 provides a background overview of the under workings of 1D CNNs for time series. Section 3 explains the methodology used. Section 4 demonstrates our findings and discussion.

II. 1D CONVOLUTIONAL NEURAL NETWORKS FOR TIME SERIES.

This section elaborates on the operation of Convolutional Neural Networks (CNNs) on time series data such as historical Load consumption data. Generally speaking, CNNs attempt to learn the relationship between the input and the output and store the learned experience in their filter weights.

CNNs are a special type of Artificial Neural Networks that are very popular in fields such as image and video recognition, image classification, natural language processing, and many more. Depending on the shape of data, CNNs use filters of similar shape to the data being processed. Time series data [11] is one-dimensional data hence the application of 1D CNNs [12],[7].

CNNs use a specialized linear operation known as a convolution in at least one of the layers in the network. Convolution [13] is a mathematical operation on two functions and is denoted with an asterisk (*).

Fig. 2(a) depicts the operation of a conv1D layer with a single filter and kernel size of 3 across 3-time steps of a sample of time series data. The layer computes a weighted sum of 3 input time steps $\{x_0,x_1,x_2\}$, adds a bias (b), and then applies an activation function to the result as seen in (1).

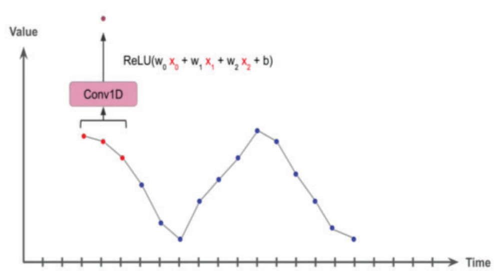

Figure 2(a). 1D Conv. Layer operation

$$Y_0 = ReLU(w_0x_0+w_1x_1+w_2x_2+b) \quad (1)$$

Where Y_0 is the first computed output by Conv1D layer, $\{x_0,x_1,x_2\}$ are input values, (w_0,w_1,w_2) are weights of the filter, b is the bias and $ReLU$ is the activation function known as rectified linear unit described in [14]

The conv1D layer then slides the filter by one time step as shown in Fig. 2(b). A weighted sum is computed on the next 3 input time steps $\{x_1,x_2,x_3\}$, a bias (b) is added and a $ReLU$ [14] activation function is applied as observed in (2). This operation is repeated up to the last input sequence.

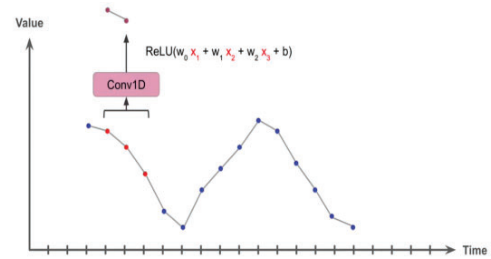

Figure 2(b). 1D Conv. Layer operation

$$Y_1 = ReLU(w_0x_1+w_1x_2+w_2x_3+b) \quad (2)$$

Where Y_1 is the second computed output by the Conv1D layer, $\{x_1,x_2,x_3\}$ are input values, (w_0,w_1,w_2) are weights of the filter, b is the bias and $ReLU$ [14] is the activation function.

Conv1D layers have no memory because each output is computed based on only a small window of input time steps which is proportional to the kernel size. This issue is overcome by stacking multiple Conv1D layers together. It's preferable to stack multiple conv1D layers each with a small kernel size. This lowers the number of parameters and hence the computation time is lower than using large kernel size.

The output sequence of a conv1D layer is shorter than the input sequence. This is rectified by padding[12] the input sequence with Zeros at both ends of the input sequence known as Same Padding. Causal padding is also another form of padding that requires all the zeros to be added at the start of the input sequence. A Conv1D layer shifts a filter across the input sequence. This is known as a stride. One stride implies the filter is shifted one-time step at a time across the input sequence.

Each Conv1D layer is specified by its filter weights which are determined in the training stage by an iterative update process. That is, they are first randomly initialized and then adjusted by backpropagation [12],[13] to minimize a cost function (3). All weights are then fixed in the testing stage. These weights play the role of system memory.

$$MSE= \sum_{i=1}^{N} \frac{(y_i-\hat{y}_i)^2}{N} \quad (3)$$

Where $\hat{y}_i$ is the forecasted value, y_i is the actual value, and N is the number of testing samples.

III. DATA ANALYSIS AND PROPOSED METHODOLOGY

This section presents the data analysis we carried out to identify the best input features for our model. We also dive into our proposed methodology in this section.

A. Feature analysis

Electrical load consumption data are time series i.e. They vary with time. Many factors affect the consumption of energy in a Commercial Building. To extract useful features from historical load data, we have to start with data analysis.

We used historical load data from April 2017 to March 2018 for a Commercial Building located in Kyushu, Japan. The data were recorded using Smart Meters at one-minute intervals. The data was unstructured and we, therefore, started by doing data cleaning and also converted the data to hourly intervals. After cleaning, we plotted various graphs of the historical load data for visualization. Fig. 3 is a sample of the time series historical data for April 2017. The trend of the time series can be observed clearly. High load consumption during working hours and low consumption at night, holidays, and on weekends.

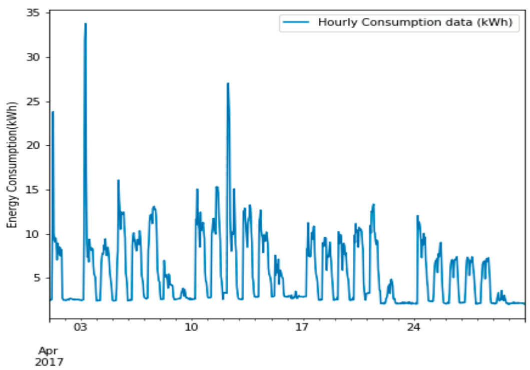

Figure 3. Sample of Historical Load Data

Neural Networks require input features for training. The main input feature is the historical load data. However electrical load consumption of most Commercial Buildings is dependent on many other factors too. Fig. 4 is a box plot that elaborates on how the load consumption is higher on weekdays than on weekends. The median for weekday data is 8.2kWh and 3.1kWh for weekend load data. Weekend consumption data has many outliers as seen in Fig. 4.

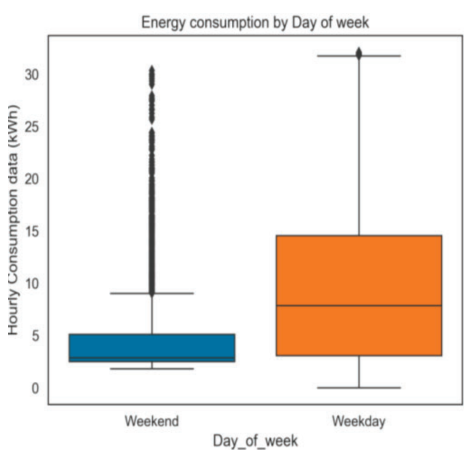

Figure 4. Weekend Vs Weekday Consumption

Weather greatly influences energy consumption in Japan e.g. Temperature, Humidity, and Wind speed. However, for this research, we only used temperature data because the effect of humidity and wind speed was very minuscule. Fig. 5 is a scatter plot that depicts how electrical load consumption varies with temperature. The temperature range of 15 degrees to 25 degrees Celsius is ideal for low energy consumption.

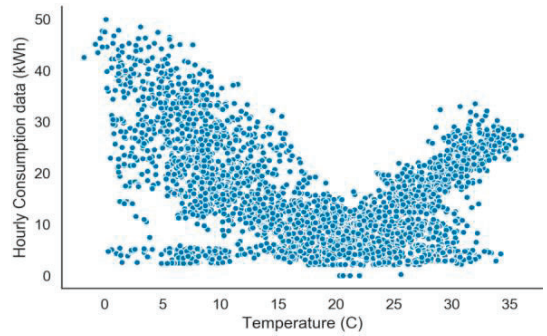

Figure 5. Scatter Plot of Load Vs Temperature

The other useful input features considered in our model were Seasons, Holidays, and working time. We used a correlation matrix to select 6 most useful input features for our Short-term Load Forecasting Model.

B. Data Preprocessing

Before feeding the input features to a Neural Network model, data preprocessing is necessary. The data types of some of the input features are categorical and others are numerical. Our model can only accept numeric data as input. Therefore, we applied one hot encoding method [6] on all the categorical data. We also normalized [5] historical load and temperature data using the Min-Max scaling function to a range of 0 to 1.

The final dataset contained 8229 data points with a mean of 9.5kWh. These were randomly split into training (92%), validation (6%), and testing (2%) samples.

C. WaveNet Architecture

Our technique involves stacking 6 1D Convolutional layers together. Each layer uses multiple filters with similar kernel size. Stride size, dilation rate and the type of padding were also set. This is summarized in Table I.

TABLE I. 1D CNN MODEL DESIGN

Model Design	**Quantity**
Number of filters	32
Kernel Size	2
Strides	1
Padding	Causal
Dilation_rate	{1,2,4,8,16,32}

The historical load consumption data sequence is fed into the input layer. The filters in the input layer compute the weighted sums and biases and pass their output as an input to the dilated [9] hidden layer and the process is repeated. The output of the output layer is then used as the next input for the input layer. This operation is repeated and forms our WaveNet [10] structure as seen in Fig. 6. All the layers are dilated as in [9] and [10] with a dilation rate of {1,2,4,8,16,32}.

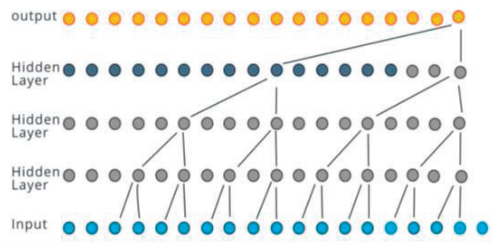

Figure 6. 1D CNN WaveNet Architecture

Lower layers learn short term patterns while higher layers learn long term patterns. Each added convolution layer doubles the receptive field [15].

The final output of the WaveNet is fed into a fully connected (FC) layer. At this stage, all the other input features are also fed into the fully connected layer. The output of the fully connected layer is then passed to the final output layer with only one neuron. This is the layer responsible for the Load Forecasting.

The training was done on the training dataset using gradient descent. Adam [16] optimizer was used to achieve this process. To increase non-linearity, we applied an activation function, ReLU[14] on all layers. The model was compiled using MSE (3) as the loss function.

IV. Experimental Setup and Results

This section is dedicated to our findings from the research. We describe the architecture and hyperparameters of our 1D CNN model. We evaluate the performance of our 1D CNN model in comparison with other popular Machine Learning models. Finally, we discuss the challenges encountered in using our model for Load Forecasting.

A. Experimental Setup

At first, we applied popular regression models on the train data and evaluated them on the validation data as summarized in Table II. MLPRegressor model performed best.

TABLE II. PERFORMANCE OF POPULAR REGRESSOR MODELS

Regression Models	MAE	RMSE
Linear Regression (Benchmark)	2.4152	3.9121
KNeighborsRegressor	2.3090	3.7458
RandomForestRegressor	2.1945	3.5490
GradientBoostingRegressor	2.4988	3.9605
MLPRegressor	2.1752	3.4136
ExtraTreesRegressor	2.2518	3.6198

We then applied different Neural Network models to the same dataset. The WaveNet-1D CNN model architecture was trained and tuned. The best performance was attained using 50 epochs, ReLU activation function, ADAM optimizer, MSE as a cost function, and a learning rate of 0.0003. Our model was designed using Keras Deep Learning library with TensorFlow as its back end. Scikit-learn and Pandas library were also very beneficial for our research.

B. Results

Fig.7 depicts the comparison of actual load consumption and our 1D CNN model predictions for one week on the test dataset. It shows a good fit and a stable forecast for Short-Term Load Forecasting.

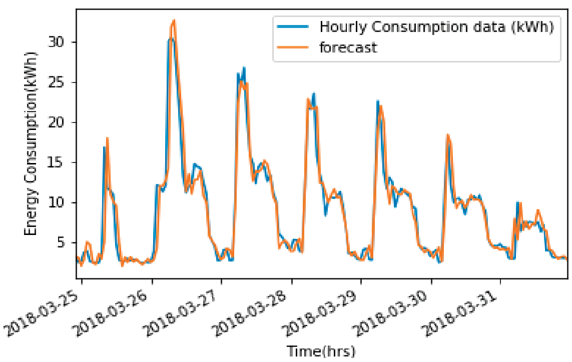

Figure 7. One week forecast Vs Actual Consumption

The performance evaluation of our model was computed using RMSE and MAE[5][6] functions on the test dataset. The result was first compared to MLPRegressor model (baseline model) whose performance was the best amongst the popular Machine Learning regression models. We then compared the evaluation results with other popular Deep Learning Models as shown in Table III. Our Model outperformed them in both Testing MAE and Testing RMSE[5][6].

TABLE II. PERFORMANCE COMPARISONS

Algorithm	Test MAE	Test RMSE
MLPRegressor (Baseline)	2.1885	3.5161
DNN	2.6631	3.8791
LSTM	1.1120	2.5143
1D CNN & LSTM	0.8244	1.9821
WaveNet-1D CNN	0.6377	1.3863

This is an indication of the power of 1D CNN WaveNet architecture to extract useful features from a raw dataset. A Test RMSE of 1.3863 implies that the model's prediction is off by + or - 1.3863kWh on the test dataset. This could vary when tested on more data from the Building used as a case study.

C. Discussion

We close off this section by stating some of the challenges Neural Networks experience in load forecasting and what we did to overcome them. The main threat to using Neural Networks in forecasting is overfitting. To avoid overfitting in our model, we shuffled the dataset during training, applied a drop out method, and also used early stopping function to avoid overtraining. Fig.8 is a snippet of the learning curve plot that shows a good fit. It indicates the comparison between train loss and test loss during training over 50 epochs.

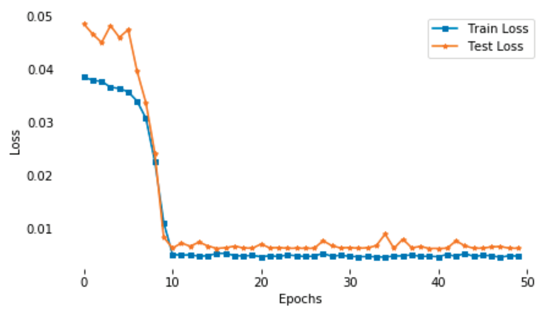

Figure 8. Learning Curve

Both LSTM and CNN models have been proved to be impressive on forecasting electrical load consumption. 1D CNN was superior to LSTM on our specific dataset but LSTM could outperform CNN on different datasets. We recommend that both LSTM and 1D CNN models are thoroughly trained and tested on a wide range of datasets to provide better comparisons.

CONCLUSION

This paper presented a unique method of short term load forecasting for commercial buildings. Notably, it proposed the use of a Deep Learning technique that utilized 1D Convolutional Neural Networks with a WaveNet architecture. Our Model was able to outperform other state-of-the-art prediction methods when applied to our benchmark dataset.1D CNN with a WaveNet architecture was able to make a week ahead forecast and showed the lowest value of average MAE and RMSE compared to other Models.

As Renewable Energy, Electric Vehicles, and Battery Storage become more affordable, effective planning and energy optimization is of key importance to Commercial Buildings. This can be facilitated by Solar PV and Load Forecasting. We hope to apply a similar technique on Solar PV Forecasting in our future works.

ACKNOWLEDGMENT

Special gratitude goes to DAI-DAN CO., LTD for the provision of the dataset for their Near-Zero Energy Building in Kyushu, Japan.

REFERENCES

[1] G. Barone, A. Buonomano, F. Calise, C. Forzano, and A. Palombo, "Building to vehicle to building concept toward a novel zero energy paradigm: Modelling and case studies," *Renew. Sustain. Energy Rev.*, vol. 101, no. May 2018, pp. 625–648, 2019, doi: 10.1016/j.rser.2018.11.003.

[2] F. Meggers *et al.*, "Reduce CO 2 from buildings with technology to zero emissions," *Sustain. Cities Soc.*, vol. 2, no. 1, pp. 29–36, 2012, doi: 10.1016/j.scs.2011.10.001.

[3] G. Juberias, R. Yunta, J. Garcia Moreno, and C. Mendivil, "New arima model for hourly load forecasting," *Proc. IEEE Power Eng. Soc. Transm. Distrib. Conf.*, vol. 1, pp. 314–319, 1999.

[4] A. Lahouar and J. Ben Hadj Slama, "Day-ahead load forecast using random forest and expert input selection," *Energy Convers. Manag.*, vol. 103, pp. 1040–1051, 2015, doi: 10.1016/j.enconman.2015.07.041.

[5] S. Bouktif, A. Fiaz, A. Ouni, and M. A. Serhani, "Optimal deep learning LSTM model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches," *Energies*, vol. 11, no. 7, 2018, doi: 10.3390/en11071636.

[6] W. He, "Load Forecasting via Deep Neural Networks," *Procedia Comput. Sci.*, vol. 122, pp. 308–314, 2017, doi: 10.1016/j.procs.2017.11.374.

[7] K. Amarasinghe, D. L. Marino, and M. Manic, "Deep neural networks for energy load forecasting," *IEEE Int. Symp. Ind. Electron.*, pp. 1483–1488, 2017, doi: 10.1109/ISIE.2017.8001465.

[8] F. Ziel, "Modeling public holidays in load forecasting: a German case study," *J. Mod. Power Syst. Clean Energy*, vol. 6, no. 2, pp. 191–207, 2018, doi: 10.1007/s40565-018-0385-5.

[9] A. van den Oord *et al.*, "WaveNet: A Generative Model for Raw Audio," pp. 1–15, 2016, [Online]. Available: http://arxiv.org/abs/1609.03499.

[10] T. Le Paine *et al.*, "Fast Wavenet Generation Algorithm," pp. 1–6, 2016, [Online]. Available: http://arxiv.org/abs/1611.09482.

[11] J. C. B. Gamboa, "Deep Learning for Time-Series Analysis," 2017, [Online]. Available: http://arxiv.org/abs/1701.01887.

[12] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, "1D Convolutional Neural Networks and Applications: A Survey," pp. 1–20, 2019, [Online]. Available: http://arxiv.org/abs/1905.03554.

[13] C. C. J. Kuo, "Understanding convolutional neural networks with a mathematical model," *J. Vis. Commun. Image Represent.*, vol. 41, pp. 406–413, 2016, doi: 10.1016/j.jvcir.2016.11.003.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2015 Inter, pp. 1026–1034, 2015, doi: 10.1109/ICCV.2015.123.

[15] W. Luo, Y. Li, R. Urtasun, and R. Zemel, "Understanding the effective receptive field in deep convolutional neural networks," *Adv. Neural Inf. Process. Syst.*, no. Nips, pp. 4905–4913, 2016.

[16] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–15, 2015.