Influential Factors for Accurate Load Prediction in a Demand Response Context

Morten Gill Wollsen
Center for Energy Informatics
Maersk McKinney-Moller Institute
University of Southern Denmark
Email: mgw@mmmi.sdu.dk

Mikkel Baun Kjærgaard
Center for Energy Informatics
Maersk McKinney-Moller Institute
University of Southern Denmark
Email: mbkj@mmmi.sdu.dk

Bo Nørregaard Jørgensen
Center for Energy Informatics
Maersk McKinney-Moller Institute
University of Southern Denmark
Email: bnj@mmmi.sdu.dk

***Abstract*—Accurate prediction of a buildings electricity load is crucial to respond to Demand Response events with an assessable load change. However, previous work on load prediction lacks to consider a wider set of possible data sources. In this paper we study different data scenarios to map the influence of the different data parameters. We also look at the temporal aspect of predicting by looking at the predicted seasons. By predicting with a MultiLayer Perceptron, which is a universal approximator, it is possible to focus solely on the influence of the parameters instead of the prediction algorithm itself. Finally, multiple prediction algorithms are compared. The influential factor analysis is based on data from an entire year from a office building in Denmark. The results show that weather data is the most crucial data parameter. A slight improvement from load data was however seen using only occupancy data. Next, the time of day that is being predicted greatly influence the prediction which is related to the weather pattern. By presenting these results we hope to improve the modeling of building loads and algorithms for Demand Response planning.**

I. Introduction

Electricity grids are facing challenges due to peak consumption and renewable electricity generation. In this context, demand response offers a solution to many of the challenges, by enabling the integration of consumer side flexibility in grid management. Commercial buildings are good candidates for providing flexible demand due to their volume and the stability of their loads. In order to successfully plan a Demand Response (DR) event, it is necessary to have accurate predictions of the loads in a building. An important aspect when improving the accuracy of load prediction is the data used to perform the prediction with. Excluding important data parameters result in that the underlying variations in load will not be accurately modeled. On the other hand, including unnecessary data parameters complicates the modeling.

From a black-box system modeling perspective, we are interested in knowing which factors influence the prediction performance. Our case is the Green Tech House in Vejle, Denmark where we focus on predicting the power consumption of the ventilation system. The Green Tech Center is a shared office building which holds a total of 38 offices and conference rooms. The ventilation system is relevant as it is well positioned to accommodate DR events, and therefore accurate predictions are crucial. Available data of the Green Tech House are ventilation power consumption, occupancy prediction [1], weather station data and weather forecasts. The occupancy prediction is based on PIR sensors placed in every room. The data is from April 2015 to April 2016. Since this is an office building, all weekends has been removed from the data as there is no activity at these times.

Kwok et al. [2] shows a cooling load prediction that has been improved by the addition of mimicked occupancy data as an additional input. They test three different scenarios with the addition of extra data in each scenario. The third scenario with all input parameters shows the best result. The work by Leung et al. [3] is similar in many ways. They also mimic the occupancy data by using primary air-handling units as well as office electricity usage. From this they test 4 different scenarios and just like Kwok et al. they show an improvement in prediction accuracy with the additional input. Newsham and Birt [4] uses PIR sensors, but also mimic the occupancy through electricity usage in offices. Their prediction accuracy only saw a slight improvement with the additional data.

The three mentioned papers and this paper all mimic the occupancy data. It is debatable which approach is most correct, but we all agree that the actual or correct occupancy data is practically impossible to get. This paper stands out as it tests the influence of weather station data and occupancy data independently, and will thereby give a richer knowledge about the influences of the individual available data parameters.

II. Method

We have derived three experiments to review the influential factors for accurate load prediction. The three experiments look at the following dimensions:

- Data addition
- Temporal dimension
- Technical dimension

The predictions will be performed by a MultiLayer Perceptron (MLP) unless otherwise stated. A Multilayer Perceptron is a feed-forward artificial neural network that is shown to be a universal approximator, robust, fault tolerant and noise immune [5]. Inside the MLP, the number of nodes in the hidden layer is the mean of the number of inputs and outputs [6]. The training is performed with the resilient backpropagation algorithm [7]. The network is trained until an error of 0.001 % with a maximum of 1000 iterations and the network uses

978-1-5090-4158-9/16/$31.00 ©2016 IEEE

the tanh activation function. To overcome the influence by the random generator, an ensemble approach of 100 parallel networks is applied. The operator in the ensemble is the median. Using the median will make the result more resistant to outliers. Additional information on ensembles and their operators can be found in [8]. The error measurement used is the root mean square error (RMSE).

Additionally, the regression is performed through a 10-fold cross-validation. The average from the 10 bins is used as the final regression error. Using cross-validation will decrease the influence of the different seasons of the single year data set, as well as give a more accurate error estimate.

A. Experiment 1: Data addition

This experiment will investigate the influence of adding additional data to the prediction. With the data parameters available a total of 6 different scenarios are derived. The input data scenarios are:

A: Load
B: Load and weather station
C: Load, weather station and occupancy prediction
D: Load, weather station and weather forecast
E: Load and occupancy prediction
F: All possible input parameters

Load data (A) is the historical ventilation power consumption, as well as the time of day and day of the week. Weather station data include relative humidity (%), temperature (°C), rain indicator, illuminance (lx), wind direction and wind speed $(\mathrm{m/s})$. Occupancy predictio is the amount of rooms predicted to be occupied [1]. Finally, weather forecast is forecasted temperature $(\mathrm{m/s})$ and forecasted solar irradiance $(\mathrm{W/m^2})$. In order to make the regression stateless in a temporal context, all data is delayed before the regression is applied. The data is in hourly resolution and the prediction window is 3 hours. The applied delay is also 3 hours. We performed a test on different delays, but a delay of 3 hours gave the best results. The results from this experiment will uncover which data parameters are necessary for a successful prediction accuracy.

B. Experiment 2: Temporal dimension

This experiment will investigate the influence of temporal patterns in the prediction accuracy. The data will be split into 4 bins that is divided by seasons. This will overrule the 10-fold cross validation. The seasons are: Spring (April to June), summer (July to September), fall (October to December) and winter (January to March). This separation came natural as the data starts in April. The experiment will reveal any influences from the seasons of the year.

C. Experiment 3: Technical dimension

The final experiment investigates the influence of the prediction algorithm. Researchers choice of modeling technique comes down to personal preference, the type of problem, but also what is *hip* at the time. By testing a variety of different algorithms the influence can be revealed. The tested algorithms are: MultiLayer Perceptron (MLP), Adaline Network, Echo State Network (ESN), Extreme Learning Machine (ELM), General Regression Neural Network (GRNN), Linear regression and Support Vector Machine (SVM). The setup for MLP has already been reviewed. All algorithms are used in an ensemble with 100 parallel runs and the median operator as mentioned earlier.

1) Adaline Network: An ADALINE artificial neural network is a simple feed-forward neural network model with no hidden layers. The weights of the neurons are updated as follows

$$w \leftarrow w + \eta(o - y)x \tag{1}$$

where w is the weight vector, η is the learning rate, o is the output of the model, y is the desired output and x is the input vector. The learning rate is 0.01. Additional information about ADALINE networks can be found in [9].

2) Echo State Network: An Echo State Network is a recurrent neural network that is initialized with a random "reservoir". Recurrent neural networks has a temporal memory making it suitable for time series predictions. Only the weights between the reservoir and the output nodes are trained. The output weights are calculated by using the Moore-Penrose pseudoinverse:

$$\mathbf{W^{out}} = \left(\mathbf{S}^{+}\mathbf{D}\right)^{T} \tag{2}$$

where $\mathbf{S}^{+}$ is the pseudoinverse of the state matrix $\mathbf{S}$ and $\mathbf{D}$ is the teacher collection matrix. $\mathbf{X^T}$ denotes the transpose of the matrix $\mathbf{X}$. The ESN uses a reservoir size of 100, a noise level of 0.0001, a spectral radius of 1.0, the tanh activation function and the leaky node with a leakage rate of 0.5. The Moore-Penrose pseudoinverse is calculated using the SVD decomposition. More information on Echo State Networks and their parameters can be found in [10].

3) Extreme Learning Machine: An Extreme Learning Machine has many similarities to ESNs, because it is also randomly initialized and only the output weights are trained. ELM is a feed-forward neural network with a single hidden layer, and the output weights are the weights between the hidden layer and the output layer. Similarly to ESN the output weights are also calculated by using the Moore-Penrose pseudoinverse:

$$\mathbf{W^{out}} = \left(\mathbf{S}^{+T}\mathbf{D}\right)^{T} \tag{3}$$

where once again $\mathbf{S}^{+}$ is the pseudoinverse of the state matrix $\mathbf{S}$ and $\mathbf{D}$ is the desired output matrix. The ELM also uses 100 nodes in its hidden layer. More information about Extreme Learning Machines can be found in [11].

4) General Regression Neural Network: A General Regression Neural Network is a feed-forward network with two hidden layers; a pattern layer and a summation layer. The pattern layer applies a Gaussian kernel and the tanh activation function. The summation layer contains two nodes. The first node is the numerator which is the summation of the multiplication of the desired output and the activation function. The second node is the denominator which is the summation of all activation functions. Both nodes are forwarded to the output layer where a fraction between the nodes from the summation layer is performed. More information about General Regression Neural Networks can be found in [12].

5) Linear regression: Although artificial neural networks are able to approximate a linear function, a successful linear regression gives an insight to the nature of the data. The linear regression performed is an Ordinary Least Squares Regression performed via a QR decomposition. In case of a singular matrix, a tiny amount of noise is added. If the matrix is singular, the addition of noise can give very wrong results, but ensures that an output can always be calculated. The linear regression is performed with the Apache Commons Math library.

6) Support Vector Machine: Modeling using Support Vector Machine, or Support Vector Regression, is a linear mapping in a high dimensional space induced by a kernel. The kernel used is a Radial Basis Function kernel and the support vector uses the ϵ-band with $\epsilon = 1 * 10^{-3}$. Additional information on Support Vector Regression can be found in [13].

D. Setting for the Data Collection

The results of the experiments will directly be connected to the building in which they are performed and the characteristics of said building. The Green Tech Center is a building with a glass facade which can mimick a greenhouse. The building is heated by district heating radiators as well as heated air ventilation. The building is cooled only by the air ventilation, or if the building's occupants open windows. The ventilation system is a combined air-refreshing and temperature regulation system. As such, the ventilation system is controlled by the amount of CO_2 in a room and the temperature.

III. Results and Discussion

The results will be presented and discussed by their individual experiments.

A. Experiment 1: Data addition

Figure 1 shows the prediction error for all 6 data scenarios. The prediction in this case is 2 hours ahead, but we also tested 1 hour and 3 hours ahead, which is our time of interest in this building. The other predictions showed similar tendencies. The predictions has been grouped by the hour of the day of the prediction. This gives insight to the nature of the ventilation load and connects the results to a daily pattern.

All data scenarios are very similar in the hours 0-6 and 13-24. The remaining hours, 6-13, is where people arrive at work, and therefore the time with the biggest variation in occupancy. It is also in these hours the solar radiation increases in power. From the data scenarios we see that scenario E with the occupancy information can boost the performance of only using the load data in scenario A, just as seen in the work by Newsham and Birt [4]. However, using only the weather station data alone in scenario B is just as good as supplying weather forecast (D) or using all available inputs (F). This indicates that the current weather is the largest influence on the ventilation requirement as mentioned earlier. This time of variation in occupancy is also the time of variation in the power of the solar radiation, as the sun really gains power in the time before midday. With the glass facade of the building, the building will be heated up by the sun in these hours. The reason that scenario E has a lower error is because of the variation in occupancy and solar power occurring at the same time of the day.

There is an interesting elevated prediction error during the beginning of the day (0-5) and in the end of the day (20-24). Any connection to the solar radiation or outside temperature would mean the error of scenario A would not be elevated, but that is not the case. By having added an input that is the time of the day, the error for all scenarios has been decreased. We believe the error originates from an inconsistent base ventilation load to keep the building at a fixed temperature. This base load is based on a temperature setpoint with both a day setting and a night setting. A possible way to lower this error would be to include an input that indicates the switch from day mode to night mode. Experiments with this additional input will be performed in future work.

The results do not match those in related work [2]–[4]. We believe there are a few reasons for this. Firstly, the very different ways the occupancy is mimicked or calculated, may influence the results, and the approach may not be independent of the ventilation load as it needs to be for the results to be comparable. Secondly, the buildings differ greatly in size and location.

B. Experiment 2: Temporal dimension

Figure 2 shows the prediction error for the four seasons of the year with the B data scenario. Similarly to experiment 1, the shown prediction is 2 hours ahead, with 1 and 3 hours ahead showing similar results. We see a distinction between the cold months (fall and winter) and the hot months (spring and summer), keeping in mind the separation of the seasons. The results confirm those of experiment 1. In the morning hours when the building's occupants go to work, we see an increased error in the hot months, which is due to the increased solar power in these months. Additionally we see a difference in the errors in the off-work hours that confirm the changing temperature setpoint mentioned in experiment 1.

The shift in the relation between input and output in the seasons can be dealt with in the modeling. By including an additional input that is the day of the year, the learning algorithms might learn to differentiate the seasons. This additional input parameter will also be experimented with in future work. Ideally the algorithm would also be trained with data from an entire year to create a good generalization. Not like in this experiment where the predicted periods were not included in the training data. However, with data from a single year, there would be no pattern from this new input that the learning algorithm could learn from. Every combination of this new input and the input that states the hour of the day would be a unique combination for every single data sample. This means a good generalized model would require data from at least two years.

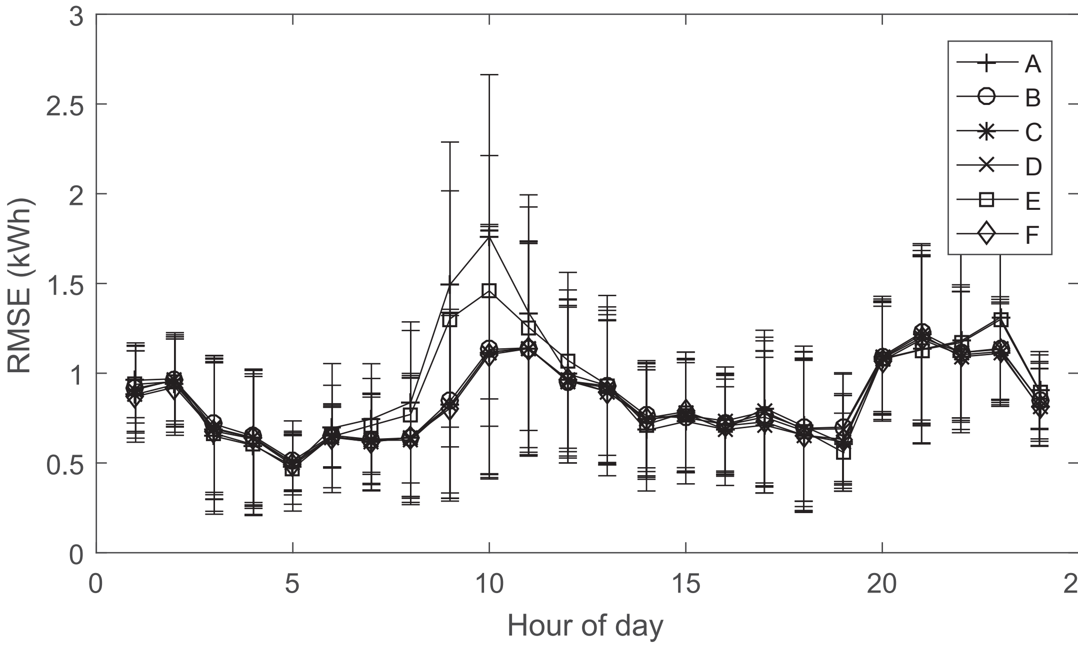

Figure 1. The error from predicting 2 hours ahead on all hours of the day with the defined data scenarios.

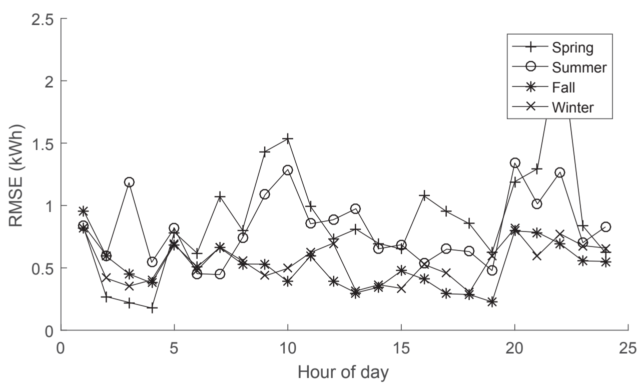

Figure 2. The prediction error of the seasons of the year from a 2 hours ahead prediction.

C. *Experiment 3: Technical dimension*

Figure 3 shows the prediction error from the different types of prediction algorithms from 1 hour ahead with the B data scenario. The predictions from 2 hours and 3 hours ahead show an overall increased error, but otherwise a similar pattern. GRNN and SVM are essentially identical with the highest error. ELM lies between the GRN and SVM pair and the rest of the algorithms. All the other algorithms (MLP, Adaline, ESN and Linear) are almost identical which is very interesting. Especially the fact that a simple Linear regression can produce an equally low error as an ANN indicates a simple relation between the input data and output data.

If the relation can be linearly modeled, we would also expect the non-linear algorithms to be able to model the system. The algorithms are configured for a general solution, and it is a possibility that a more specific configuration is required for SVM, GRNN and ELM to reach the performance of the other algorithms. It is also a possibility that the dimensionality of the data is too big and that feature selection or feature reduction is a necessary step for those algorithms.

Because the historic power consumption is one of the input parameters, a primitive model would be a simple delay of the historic input. However, a check shows that the linear model includes every input in its model, indicating that every input parameter, including the weather, is crucial to the model's accuracy. A linear relation also allows for more simple modeling algorithms besides artificial neural networks for future modeling.

IV. Conclusion

In this paper we have tested a variety of different data scenarios, temporal scenarios and technical scenarios through various tests to map the factors that influence the accuracy of ventilation power consumption predictions. In two of the experiments we see that the most influential data parameter is weather data. The current weather is the main thing the ventilation system has to account for. Adding additional data such as weather forecasts or occupancy will not further increase the accuracy. We also see that the time of the day that is predicted has a big influence. This is again directly related to the sun's radiation, which is the main heat source of the case building. A slight increase in prediction accuracy was possible with occupancy data alone because of the relation between occupants' work schedule and the solar radiation.

The results of this paper can be used when modeling building characteristics such as ventilation load or to inform the data selection process for modeling.

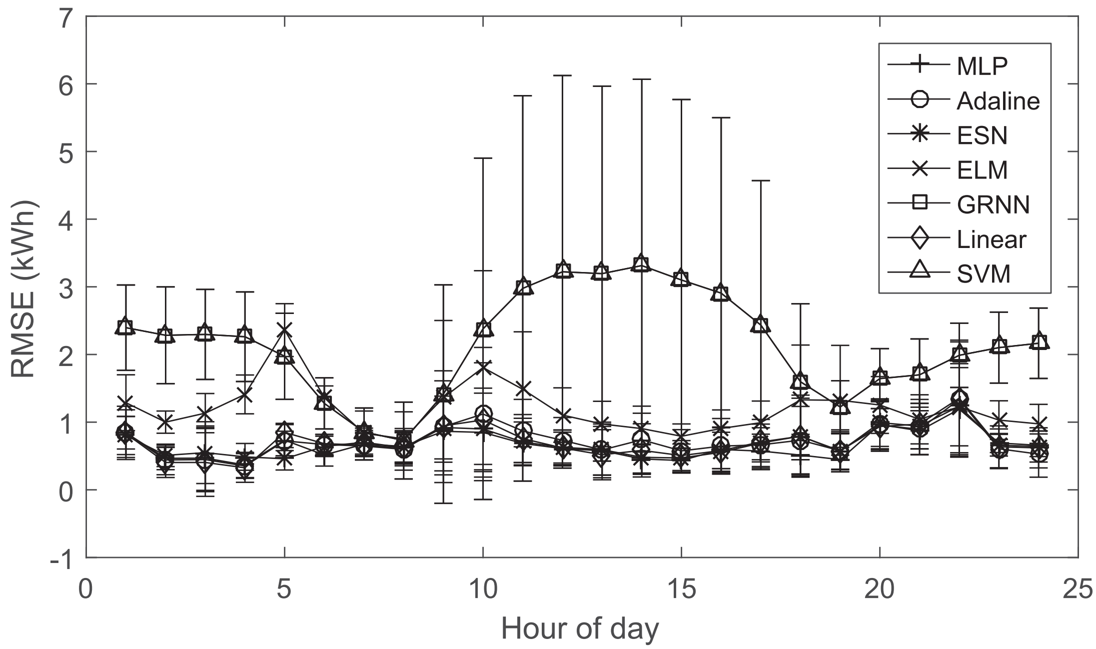

Figure 3. The error from predicting 1 hour ahead with the different prediction algorithms.

ACKNOWLEDGEMENT

This work is supported by EUDP for the project Demand-Response Capacity Management in Commercial Buildings (64014-0512).

REFERENCES

[1] F. C. Sangogboye, K. Imamovic, and M. B. Kjærgaard, "Improving occupancy presence prediction via multi-label classification," in *2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, 2016.

[2] S. S. Kwok, R. K. Yuen, and E. W. Lee, "An intelligent approach to assessing the effect of building occupancy on building cooling load prediction," *Building and Environment*, vol. 46, 2011.

[3] M. Leung, C. Norman, L. L. Lai, and T. T. Chow, "The use of occupancy space electrical power demand in building cooling load prediction," *Energy and Buildings*, vol. 55, 2012.

[4] G. R. Newsham and B. J. Birt, "Building-level occupancy data to improve arima-based electricity use forecasts," in *Proceedings of the 2nd ACM workshop on embedded sensing systems for energy-efficiency in building*, 2010.

[5] G. Zhang, B. E. Patuwo, and M. Y. Hu, "Forecasting with artificial neural networks:: The state of the art," *International journal of forecasting*, vol. 14, 1998.

[6] J. Heaton, *Introduction to neural networks with Java*. Heaton Research, Inc., 2008.

[7] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: The rprop algorithm," in *Neural Networks, 1993., IEEE International Conference on*, 1993.

[8] N. Kourentzes, D. K. Barrow, and S. F. Crone, "Neural network ensemble operators for time series forecasting," *Expert Systems with Applications*, vol. 41, 2014.

[9] B. Widrow and M. A. Lehr, "30 years of adaptive neural networks: perceptron, madaline, and backpropagation," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1415–1442, 1990.

[10] M. Lukoševičius and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," *Computer Science Review*, vol. 3, no. 3, pp. 127–149, 2009.

[11] G.-B. Huang, D. H. Wang, and Y. Lan, "Extreme learning machines: a survey," *International Journal of Machine Learning and Cybernetics*, vol. 2, no. 2, pp. 107–122, 2011.

[12] D. F. Specht, "A general regression neural network," *IEEE transactions on neural networks*, vol. 2, no. 6, pp. 568–576, 1991.

[13] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and computing*, vol. 14, no. 3, pp. 199–222, 2004.