Quantile regression using gradient boosted decision trees for daily residential energy load disaggregation

B Delcroix¹, S Sansregret¹, G Larochelle Martin¹ and A Daoud¹

¹ Hydro-Québec Research Institute, Laboratoire des Technologies de l’Énergie, Shawinigan, QC, Canada
delcroix.benoit@hydroquebec.com

Abstract. The building sector is responsible for approximately one-third of the total energy consumption, worldwide. This sector is undergoing a major digital transformation, buildings being more and more equipped with connected devices such as smart meters and IoT devices. This transformation offers the opportunity to better monitor and optimize building operations.

In the province of Quebec (Canada), most buildings are equipped with smart meters providing electricity usage data every 15 minutes. A current major challenge is to disaggregate the different energy use from smart meter data, a discipline called non-intrusive load monitoring in literature. In this work, the aim is to develop and validate a potentially generalizable model for all houses that identifies the daily share of each energy use based on building information, weather data and smart meter data. Input features are selected and ordered using an aggregated score composed of the correlation coefficient, the feature importance given by a decision tree, and the predictive power score. Two modelling methods based on quantile regression are tested: linear regression (LR) and gradient boosted decision trees (GBDT). Compared to ordinary least squares regression, quantile methods inherently provide more robustness and confidence intervals. Both models are trained and validated using separate datasets collected in 8 houses in Canada where metering and sub-metering were performed during a whole year.

Results on the test dataset indicate a better performance of the GBDT model, compared to the LR model, with a coefficient of determination of 0.88 (vs. 0.78), a mean absolute error of 6.34 % (vs. 8.89 %) and a maximum absolute error between the actual and predicted values in 95 % of the cases of 17.2 % (vs. 23.1 %).

1. Introduction

Worldwide, the building sector represents around one third of the final energy consumption [1] and approximately 50 % of the whole electricity use [2]. In the province of Quebec in Canada, this sector also represents around one third of the energy consumption, but it consumes around 55 % of the electricity, i.e. slightly more compared to the world [3]. Identifying electrical loads in buildings and implementing efficiency measures to make electricity use more grid-friendly are current challenging objectives for utilities, worldwide.

The identification of electrical loads in buildings based only on smart meter data is a task that is commonly called Non-Intrusive Load Monitoring (NILM) or load disaggregation in literature. The first studies on NILM were achieved in the 1980s at the MIT Energy Institute, e.g. by Hart [4]. Since then, the number of contributions in the NILM field has been increasing over time. Numerous review papers cover extensively this topic [5]–[7]. Ruano et al. [8] identified 4 main steps for the...
development of NILM solutions, i.e.: 1) data collection from smart meters and other potential devices (sub-metered energy uses); 2) event detection, i.e., identification of any change in the state of a device over time; 3) feature extraction, i.e., getting information about the load signature of a device; 4) load identification, i.e., based on the features previously defined, a classification process estimates the states of the devices and/or which devices are operating at a specified period or time.

As expected, datasets are critical to develop efficient NILM solutions. Previous references [5]-[8] provided a long list of available datasets used in the field. One of the key parameter influencing the choice of a NILM method is the sampling rate. High sampling rates (1 sample per second, or more) favors NILM methods that track the load signatures of devices. Lower sampling rates (1 sample per minute, hour or even day) make the load disaggregation task more and more complicated. In literature, the contrast between high and low frequency NILM method is common and well documented [8].

A wide diversity of approaches is used to achieve energy load disaggregation. Traditional disaggregation algorithms commonly used for high-frequency applications and as benchmarks are based on combinatorial optimisation and factorial hidden Markov model [9]. Approaches combining classical machine learning (ML) techniques such as support vector machines and k-nearest neighbor methods are also popular in the field [10]. More recently, deep learning (DL) methods such as recurrent and convolutional neural networks have been introduced [11]. In general, ML and DL techniques turn out to be efficient and commonly used for energy use prediction [12]. Birt et al. [13] developed a methodology to disaggregate hourly electricity consumption of Canadian dwellings into base and activity loads. The model is based on hourly and minutely metered electricity consumption from meters and heating/cooling systems. The modelling approach is a piecewise regression-based model that captures the strong correlation observed between external temperatures and electrical consumptions. Kipping and Trømborg [14] developed a load disaggregation methodology based on hourly electricity consumption coming from smart meters and applied to Norwegian dwellings. The load disaggregation model is based on ordinary least square linear regression using several explanatory variables as inputs, including weather data (heating degree days, especially) and house properties (dwelling type, occupancy, etc.). This model aims at identifying two kind of loads from the whole electricity consumption: temperature-independent (related to the baseload) and temperature-dependent (related to space heating). Deb et al. [15] developed a load disaggregation algorithm based on a clustering approach, i.e., an unsupervised classification methodology, enabling to identify space heating and domestic hot water loads from the total electricity consumption.

In this work, two traditional ML techniques are used to train a daily load disaggregation model in a supervised manner using smart meter and sub-metered data collected during a year from 8 different houses. The main aim is to develop a model that is general to all buildings, and not specific to each of them. In section 2, the data and methods used in this study are presented, including data preprocessing and selection (correlation statistical tests and ranking), modelling approaches for load disaggregation, and error indicators. In section 3, are presented the results about the importance of each input variable, the performance of both load disaggregation modeling approaches, and some considerations about the computing times during training and inference.

2. Data and methods

Developing load disaggregation models requires insightful data, appropriate algorithms and performance indicators. This section provides details about these topics.

2.1. Data

Data were collected during a year (1\textsuperscript{st} of May 2017 \rightarrow 30\textsuperscript{th} of April 2018) from 8 individual houses located in the region of Trois-Rivières, Québec, Canada. Table 1 presents general properties of each house. For each of them, a complete dataset is created, including all the variables categorized and presented in Table 2. These variables can also be classified into two groups: first, time series including weather, calendar and electricity use data; second, constant variables providing time-invariant
properties related to the occupancy, the house characteristics and the presence of equipment (generally binary variables).

| House ID | Location         | # of occupants | Construction year | Area [m²] | # of rooms |
|----------|------------------|----------------|-------------------|-----------|------------|
| 68       | Trois-Rivières   | 4              | 1960              | 191       | 12         |
| 69       | Trois-Rivières   | 5              | 2001              | 245       | 11         |
| 73       | Trois-Rivières   | 4              | 2000              | 219       | 13         |
| 82       | Trois-Rivières   | 3              | 1987              | 183       | 10         |
| 85       | Trois-Rivières   | 4              | 1976              | 222       | 12         |
| 91       | Saint-Étienne-des-Grès | 4 | 2011 | 295 | 10 |
| 92       | Trois-Rivières   | 4              | 2015              | 259       | 10         |
| 93       | Shawinigan       | 2              | 1988              | 170       | 10         |

| Data type            | List of variables                                                                 | Ref. |
|----------------------|----------------------------------------------------------------------------------|------|
| Calendar (time series) | day of week [category]                                                        | n.a. |
| Weather (hourly time series) | global horizontal irradiation (GHI) [W/m²]; direct horizontal irradiation (DHI) [W/m²]; wind speed [m/s]; temperature [°C]; dew point [°C]; relative humidity [%]; pressure [atm] | [16] |
| Occupancy (constant)  | # of children [-]; # of adults [-]; # of animals [-]                              |      |
| House properties (constant) | heated garage [binary]; house type [category]; # of floors [-]; construction year [YYYY]; area [m²]; # of rooms (excluding garage) [-] |      |
| Equipment (constant)  | heat pump [binary]; # of thermostats [-]; dehumidifier [binary]; ventilation [binary]; window air conditioner [binary]; swimming pool [binary]; spa [binary]; Additional heating system such as baseboard [binary], gas-fired heater [binary], fireplace [binary] or other [binary] | In-house |
| Electricity use (minutely time series) | meter [kWh]; heating [kWh]; domestic hot water [kWh]; other [kWh] |      |

### 2.2. Data pre-processing

Before any developments related to data selection and ranking, and model development, data pre-processing is needed to obtain a clean dataset in line with the main aim to obtain a model generalizable to all buildings. Several data pre-processing techniques are used, i.e.:

- Daily resampling of all time series: for weather data, daily averages of each variable are computed; for electricity use, the total daily consumptions are computed for each use (meter [total], heating, domestic hot water [DHW], and other).
- One-hot encoding of the day of week, which is the only categorical variable.
- Definition of input variables: all variables except the 3 energy uses (heating, DHW and other) are potential input variables. Only the ones that are significant (see section 2.3.) will be retained as input variables to train the load disaggregation models.
- Definition of output variables: the output variables of interest are the share (in percentage) of each electricity use (heating, DHW and other). The daily share of each use is calculated based on the data given in kWh.
- Data partitioning (training, validation and test datasets): both load disaggregation models are based on traditional ML techniques (linear regression [LR] and gradient boosted decision trees [GBDT]). To avoid overfitting, a common issue with ML techniques, data partitioning is often
achieved. In this study 80 % of the data are used for model training and the remaining 20 % are put aside for testing. In the GBDT case, among the 80 % of the training data, 25 % (i.e., 20 % of the whole dataset) is used for validation during training.

2.3. Correlation statistical tests
As Table 2 suggests, there are a lot of potential input variables that could explain the three targeted energy uses, i.e., heating, DHW and other. To avoid high dimensional input matrices and reduce computing time, only the input variables that influence the chosen energy uses are retained. To do so, two correlation statistical tests are conducted to quantify the influence of each input variable on each output variable (pair-wise analysis): the Pearson and Spearman tests. The Pearson correlation test measures the strength of a linear association between two variables. The Spearman test is the non-parametric version of the Pearson test and it measures the strength and direction of monotonic association between 2 variables. For details about both tests, see [17], [18]. Two criteria are used to decide if an input variable is retained: first, the absolute value of the correlation coefficient \( r \) should be higher than 0.1 for at least one output variable; second, the p-value of the statistical test should be lower than 0.05 for at least one output variable.

2.4. Data ranking methodology
Once the correlation statistical tests conducted, some input variables explaining the output variables are retained. Among these selected input variables, some are more insightful than others. To rank these variables, a method in 3 steps has been developed: 1) for each pair of input and output variables, correlation coefficients are computed (Pearson coefficient, power predictive score [19], and feature variables, a method in 3 steps has been developed: 1) for each input variable and for each correlation coefficient correlation coefficients are computed (Pearson coefficient, power predictive score [19], and feature variables, a method in 3 steps has been developed: 1) for each pair of input and output variables, 3 normalized coefficients are calculated based on Table 3; 3) for each input variable, the sum of the 3 normalized coefficients is computed to obtain an aggregated score.

Table 3. Methodology to calculate the normalized coefficients.

| Inputs \( C_{n,i} \) | \( \ldots \) | Outputs \( O_1 \) \( \ldots \) \( O_m \) | Average of each row | Normalized coefficient |
|------------------|------------------|------------------|------------------|------------------|
| 1 \( \| C_{1,1} \| \ldots \| C_{1,m} \| \) | \( M_1 \) | \( M_1 - \min(M_1, \ldots, M_n) \) | \( \max(M_1, \ldots, M_n) - \min(M_1, \ldots, M_n) \) |
| \( \ldots \) | \( \ldots \) | \( \ldots \) | \( \ldots \) | \( \ldots \) |
| 1 \( \| C_{n,1} \| \ldots \| C_{n,m} \| \) | \( M_n \) | \( M_n - \min(M_1, \ldots, M_n) \) | \( \max(M_1, \ldots, M_n) - \min(M_1, \ldots, M_n) \) |

2.5. Proposed load disaggregation methods
Both proposed models are based on quantile regression methods [21]. Compared to traditional ordinary least squares regression, quantile methods inherently provide more robustness and confidence intervals. The loss function \( \mathcal{L} \) for quantile regression can be formulated as follows:

\[
\mathcal{L}_\alpha(y, \hat{y}) = \sum_{i=1}^{n} 1_{y_i < \hat{y}_i} (1 - \alpha) |y_i - \hat{y}_i| + 1_{y_i \geq \hat{y}_i} \alpha |y_i - \hat{y}_i| \quad (1)
\]

Where \( \alpha \) is a value between 0 and 1 defining the quantile; \( y \) is the vector (size \( n \)) of output values; \( \hat{y} \) is the vector (size \( n \)) of predicted output values; 1 is a binary function having a value of 1 if the subcripted condition is met, 0 otherwise. This loss function is strictly similar to the mean absolute error loss function if the quantile \( \alpha \) is equal to 0.5, i.e., for median values.

In the following sub-sections, both chosen quantile regression methods are briefly described: first, the quantile linear regression method; second, the gradient boosted decision trees method.
2.5.1. Quantile linear regression. Quantile linear regression (LR) models present the same general form as ordinary least squares linear regression models, which can be formulated as follows:

\[ \hat{y}_i = a x_i + b \]  

(2)

Where \( \hat{y}_i \) is the estimate of the output value for the \( i \)th sample; \( a \) is the row vector of weights; \( x_i \) is the column vector of input values for the \( i \)th sample (same size as vector \( a \)); \( b \) is the intercept. The parameters included in \( a \) and \( b \) are identified by minimizing the loss function (equation (1)). In this work, the quantile linear regression model is developed in Python using the library mlinsights [22].

2.5.2. Quantile gradient boosted decision trees. This kind of model can be formulated as follows:

\[ \hat{y}_i = \bar{y} + f_1(x_i) + \cdots + f_m(x_i) \]  

(3)

Where \( \bar{y} \) is the average of the output values observed in the training dataset; \( f \) is a decision tree function; \( m \) is the number of decision trees included in the model. During optimization (minimization of the loss function defined in equation (1)), additional decision trees are progressively added until the maximum number is reached or until a new decision tree does not improve the loss function. The principle of GBDT model is to add progressively new estimators (in this case, decision trees) during optimization to correct previous errors. In this work, the quantile gradient boosted decision trees model is developed in Python using the library Scikit-Learn [23].

2.6. Error indicators

In this study, 3 error indicators are used: the first two are well-known, i.e., the coefficient of determination \( R^2 \) and the mean absolute error \( MAE \); the last chosen indicator is the 95th percentile of the absolute error \( AE \).

3. Results and discussion

This section focuses on presenting the main results of this work, especially on the input feature importance and the performance of both models (Quantile-LR and Quantile-GBDT). A brief discussion on the computing times concludes this section.

3.1. Feature importance

From all the variables listed in Table 2, only 9 are retained based on the correlation statistical tests achieved according to the methodology presented in section 2.3. Table 4 presents the kind of correlation between the 9 retained input variables and the 3 output variables (i.e., the share of each electricity use [heating, DHW and other]). Two variables are considered linearly or monotonically correlated if the absolute value of the Pearson or Spearman correlation coefficient is higher than 0.1 and if the p-value is lower than 0.05, respectively. In summary, Table 4 highlights the importance of weather variables for all output variables, and the occupancy and the equipment for DHW.

Figure 1 confirms the importance of weather data based on the aggregated score calculated for each input variable and based on the methodology presented in section 2.4.

To conclude this section, we should highlight that the statistical analysis is based on only 8 houses. Results should then be put into perspective.

3.2. Load disaggregation performance

Two models for daily electrical load disaggregation have been developed based on the methods presented in section 2.5., i.e., a first based on quantile linear regression (LR) and a second based on quantile gradient boosted decision trees (GBDT). Both have been trained on the training dataset (80 % of the whole dataset). In this section, their performances are assessed on the test dataset, i.e., the remaining 20 % (unknown during training). Table 5 presents the errors observed in the whole test dataset and for each electricity use. This table clearly highlights that the GBDT model is always more efficient than the LR model.
Table 4. Correlation between input and output variables.

| Input variables | Heating | Domestic hot water | Other |
|-----------------|---------|--------------------|-------|
| DHI             | L.C. / M.C. | N.C.              | L.C. / M.C. |
| Dew point       | L.C. / M.C. | L.C. / M.C.       | L.C. / M.C. |
| GHI             | L.C. / M.C. | N.C.              | L.C. / M.C. |
| Main            | L.C. / M.C. | L.C. / M.C.       | L.C. / M.C. |
| Temperature     | L.C. / M.C. | L.C. / M.C.       | L.C. / M.C. |
| Wind speed      | L.C. / M.C. | M.C.              | L.C. / M.C. |
| Dehumidifier    | N.C.      | L.C. / M.C.       | N.C.   |
| # of children   | N.C.      | L.C. / M.C.       | N.C.   |
| Spa             | N.C.      | L.C. / M.C.       | N.C.   |

L.C.: linearly correlated; M.C.: monotonically correlated; N.C.: not correlated

Figure 1. Aggregated score of input variables.

Table 5. Errors observed in the whole test dataset.

| Output variable | Quantile-LR (model 1) | Quantile-GBDT (model 2) |
|-----------------|-----------------------|-------------------------|
|                 | $R^2$ [-] | MAE [%] | $AE$ – Percentile 95 [%] | $R^2$ [-] | MAE [%] | $AE$ – Percentile 95 [%] |
| Heating         | 0.74      | 10.20   | 24.52          | 0.87      | 6.39    | 19.29          |
| DHW             | 0.34      | 5.60    | 15.24          | 0.52      | 4.80    | 13.02          |
| Other           | 0.68      | 10.87   | 24.44          | 0.82      | 7.84    | 19.55          |
| All             | 0.78      | 8.89    | 23.07          | 0.88      | 6.34    | 17.18          |

The results are less unequivocal if each house and use are analyzed separately. Figure 2 compares the true and predicted distributions of output values for each house and use. It shows that the GBDT model is more efficient in most of the cases (around two thirds), reproducing with more fidelity the distributions of observed output values, especially the median values (black lines in the boxplot). However, the remaining one third of the cases (highlighted by the rectangles with grey edges in Figure 2) are more adequately estimated with the LR model, predicted median values being closer to the observed ones. More globally, results presented in Table 5 and Figure 2 show that the GBDT model is more capable to estimate the daily share of each electricity use, compared to LR model.

3.3. Computing time

Table 6 provides hints about the equipment used for this study and the computing times observed during training and inference, showing that both models are low resource intensive.
Figure 2. Comparison between observed and simulated (GBDT in orange and LR in green) value distributions (displayed percentiles: 5, 25, 50, 75 and 95) for each house and use – test dataset.

Table 6. Computing times for model training and inference.

| Task       | Quantile-LR      | Quantile-GBDT     |
|------------|------------------|-------------------|
| Training   | 0.44 second/model| 0.57 second/model |
| Inference  | $4.06 \times 10^{-6}$ second/sample | $5.22 \times 10^{-6}$ second/sample |
| Configuration | Laptop – Intel Core i7 2.2 GHz – 16 Go RAM | |

4. Conclusion

Compared to sub-metering, load disaggregation techniques are cost-effective solutions to identify energy uses in buildings. They also turn out to be useful for key end applications like fault detection and diagnosis, and load control. In this work, two models based on quantile regression are tested: linear regression and gradient boosted decision trees. Previously, data were collected, preprocessed and selected based on a defined methodology. Results show that the most important input variables are related to weather, especially outside temperature. The comparison between both models shows a clear advantage of gradient boosted decision trees, yielding lower errors and better replications of output value distributions.

References
[1] International Energy Agency, “Buildings - A source of enormous untapped efficiency potential,” 2020. https://www.iea.org/topics/buildings (accessed Nov. 29, 2020).
[2] International Energy Agency, “Electricity Information: Overview,” 2020. https://www.iea.org/reports/electricity-information-overview (accessed Feb. 24, 2021).
[3] J. Whitmore and P.-O. Pineau, “État de l’énergie au Québec,” HEC Montréal, 2021.
[4] G. W. Hart, “Nonintrusive appliance load monitoring,” Proceedings of the IEEE, vol. 80, no. 12, 1870–1891, 1992.
[5] S. S. Hosseini, K. Agbossou, S. Kelouwani, and A. Cardenas, “Non-intrusive load monitoring through home energy management systems: A comprehensive review,” Renewable and Sustainable Energy Reviews, vol. 79, 1266–1274, 2017.
[6] A. Zoha, A. Gluhak, M. A. Imran, and S. Rajasegarar, “Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey,” Sensors, vol. 12, no. 12, 16838–16866, 2012.
[7] M. Zeitman and K. Roth, “Nonintrusive appliance load monitoring: Review and outlook,” IEEE transactions on Consumer Electronics, vol. 57, no. 1, 76–84, 2011.
[8] A. Ruano, A. Hernandez, J. Ureña, M. Ruano, and J. Garcia, “NILM techniques for intelligent home energy management and ambient assisted living: A review,” Energies, vol. 12, no. 11, 2203, 2019.
[9] N. Batra et al., “NILMTK: An open source toolkit for non-intrusive load monitoring,” 2014, 265–276.
[10] S. S. Hosseini, S. Kelouwani, K. Agbossou, A. Cardenas, and N. Henao, “Adaptive on-line unsupervised appliance modeling for autonomous household database construction,” International Journal of Electrical Power & Energy Systems, vol. 112, 156–168, 2019.
[11] P. Huber, “Deep Neural Networks for Non-Intrusive Load Monitoring on Low Frequency Data,” presented at the Conference EU NILM 2019, Thessaloniki, Greece, Oct. 02, 2019. Accessed: Feb. 25, 2021. [Online]. Available: https://www.youtube.com/watch?v=010fawyCOCs
[12] J. Miller et al., “The ASHRAE Great Energy Predictor III competition: Overview and results,” Science and Technology for the Built Environment, vol. 26, no. 10, 1427–1447, Nov. 2020, doi: 10.1080/23744731.2020.1795514.
[13] J. B. Birt, G. R. Newsham, I. Beausoleil-Morrison, M. M. Armstrong, N. Saldanha, and I. H. Rowlands, “Disaggregating categories of electrical energy end-use from whole-house hourly data,” Energy and Buildings, vol. 50, 93–102, 2012.
[14] A. Kipping and E. Tromborg, “Modeling and disaggregating hourly electricity consumption in Norwegian dwellings based on smart meter data,” Energy and Buildings, vol. 118, 350–369, 2016.
[15] C. Deb, M. Frei, J. Hofer, and A. Schlueter, “Automated load disaggregation for residences with electrical resistance heating,” Energy and Buildings, vol. 182, 61–74, 2019.
[16] National Renewable Energy Laboratory, “National Solar Radiation Database,” 2020. https://nsrdb.nrel.gov/ (accessed May 15, 2020).
[17] Laerd statistics, “Pearson Product-Moment Correlation,” 2018. https://statistics.laerd.com/statistical-guides/pearson-correlation-coefficient-statistical-guide.php (accessed Feb. 26, 2021).
[18] Laerd statistics, “Spearman’s Rank-Order Correlation,” 2018. https://statistics.laerd.com/statistical-guides/spearmans-rank-order-correlation-statistical-guide.php (accessed Feb. 26, 2021).
[19] F. Wetschoreck, “RIP correlation. Introducing the Predictive Power Score,” 2020. https://towardsdatascience.com/rip-correlation-introducing-the-predictive-power-score-3d90808b9598 (accessed Feb. 26, 2021).
[20] Scikit-Learn, “Feature selection,” 2020. https://scikit-learn.org/stable/modules/feature_selection.html (accessed Feb. 26, 2021).
[21] Y. Li and J. Zhu, “L1-norm quantile regression,” Journal of Computational and Graphical Statistics, vol. 17, no. 1, 163–185, 2008.
[22] X. Dupré, “Quantile regression with scikit-learn,” 2018. http://www.xavierdupre.fr/app/mlinsights/helpsphinx/blog/2018/2018-05-07_quantile_regression.html (accessed Feb. 27, 2021).
[23] F. Pedregosa et al., “Scikit-learn: Machine learning in Python,” the Journal of machine Learning research, vol. 12, 2825–2830, 2011.