Auto-tuning method for data-driven models in building energy consumption prediction: a case of cooling load prediction

Xuyuan Kang¹, Da Yan¹, Yuan Jin¹, Hongsan Sun¹
¹Building Energy Research Center, School of Architecture, Tsinghua University
* yanda@tsinghua.edu.cn

Abstract. Building consumes a significant portion of energy in the world. Improving energy use efficiency in building sector is a key approach of achieving sustainable and environmental friendly development targets. Building energy consumption prediction is essential for energy planning, equipment control and system management. Traditional physics-based model is widely used but requires sophisticated input parameter. Data-driven models, on the other perspective, utilize historical data to make predictions for future scenarios. However, the accuracy of this model greatly relies on the parameters of the machine learning algorithms within. This research proposes an auto-tuning method for machine learning algorithms in building energy prediction models, and discusses the difference between global optimum and local optimum. Based on this research, the accuracy and efficiency of the data-driven models are significantly improved.

1. Introduction
Building represents a large portion of total energy consumption in the world. Building-related sector takes up approximately 30% of total energy consumption in China. It is very essential to evaluate and analyse the building energy consumption from technical perspective. Building energy consumption prediction is considered a crucial approach to improve the efficiency in design phase and operation control, thus providing effective solutions for reducing energy consumption in buildings.

Two major approaches have been widely conducted in building energy consumption prediction: physical modelling approach and data-driven approach. Physical modelling approaches are simulation procedures based on thermodynamic principles. DeST, EnergyPlus and ESP-r are all examples of physics-based utility simulation software. Physical models offer insights on building energy consumption from a detailed heat balance mechanism, however require vast building and environmental information, which, under some occasions, may not be available for analysis.

Data-driven models, on the other hand, acquire knowledge from the historical data and do prediction for future scenarios. Since the model is built upon historical data and based on mathematical algorithms, the physical mechanism may not be reflected within this model. However, this approach doesn’t require detailed building information, and proves to be very efficient under multiple scenarios. Current data-driven models utilizes machine learning algorithms including support vector machines (SVM), artificial neural networks (ANN), extreme gradient boosting (XGB) etc. These algorithms are very popular and efficient in load prediction, but the result from a same machine learning algorithm may vary with different parameter combinations. The application of the machine learning algorithms in cooling load prediction rely on the tuning of model parameters.

To improve the performance of data-driven models in building energy consumption prediction, this paper analyzes the basic principle and structure of 3 major machine learning algorithms. The paper then introduces an auto tuning method to automatically select the optimal combination of parameters and evaluate the performance of the optimal model. Followed with methodology section, a case of
cooling load prediction is introduced to evaluate the effect of auto-tuning method. In the discussion part, the efficiency and accuracy of local optimal model and global optimal model are compared and analyzed. Finally, in the conclusion, the application and further research perspective of this research is analyzed.

2. Methodology
The performance of a data-driven model is largely depended on its model parameters. The combination of the value of the parameters may vary significantly for different scenarios. Thus, the model parameters need to be tuned based on historical data. According to the basic model training procedure, all historical data is to be split into training set and validation set. Training set is used to training data with different parameter combinations, while validation set is used to evaluate the accuracy of the model. The ratio of training and validation set in this research is 70-30, with 70% as training set and 30% as validation set. Three major machine learning algorithms including MLP, SVR and XGB are introduced in the energy consumption prediction analysis.

Multi-Layer Perception (MLP) is a neural-network-based computing algorithm. The structure of the MLP algorithm is shown below. The overall structure contains three general layers, input layer, output layer and hidden layers. Input layer contains all data of input variables. Output layer contains information of output variables, which, in this case, is the monitored or predicted building energy consumption or cooling/heating loads. While the hidden layers contain “neurons” that connect the input layer with output layer. In this framework, each neuron contains a non-linear function. The data flow between adjacent layers are connected by linear combinations.

Key model parameters in MLP algorithm are listed below:
- n_layer: the number of hidden layers in MLP network.
- n_neuron: the number of neurons in each hidden layer.
- activation: the form of non-linear activation function in each neuron.
- alpha: L2 penalty parameter. Increasing alpha may increase the risk of overfitting.

The proper range of the value of the parameter is listed in the table below.

| Parameter | Selected Values |
|-----------|-----------------|
| activation| identity, logistic, tanh, relu |
| n_layer   | [1, 2, 3, 4]    |
| n_neuron  | [10, 20, 50, 100, 200] |
| alpha     | [1e-6, 1e-4, 1e-2, 1, 10, 100] |

Support Vector Regression (SVR) is oriented from support vector machine mechanism. In SVR models, each sample is represented by a point in space. The algorithm maps each point to a high-dimension space and uses a linear “surface” to do the regression. New samples are then mapped into the same high-dimension space and make the prediction. A sketch of the mechanism is illustrated in the figure below.
Figure 2. The structure of SVR algorithm (Credit to www.saedsayad.com)

Key model parameters in SVR algorithm are listed below:
- kernel: the kernel function which related to the mapping from the original data space to high dimension space
- C: penalty parameter of the error term. Increasing C may increase the risk of overfitting.
- epsilon: specify the width of the tube within SVR model in which no penalty is conducted in the training loss function.

The proper range of the value of the parameter is listed in the table below.

| Parameter | Selected Values |
|-----------|-----------------|
| kernel    | linear, polynomial, sigmoid, rbf |
| C         | [0.1, 1, 10, 100, 1000, 10000] |
| epsilon   | [0.001, 0.01, 0.1, 1, 10, 100, 1000] |

Extreme Gradient Boosting (XGB) is oriented from decision tree theory. During the training process, a series of trees are setup with scoring on each leaf nodes. The sum of the prediction scores for each sample on each tree is the predicted result of the proposed sample. The structure of the XGB algorithm is illustrated in the figure below.

Figure 3. The structure of XGB algorithm (Credit to XGBoost Documentation)

Key model parameters in XGB algorithm are listed below:
- max_depth: maximum depth of a tree.
- min_child_weight: minimum sum of instance weight in a child
- alpha: L1 regularization term on weights. Increasing this value may increase overfitting risk.
- lambda: L2 regularization term on weights. Increasing this value may increase overfitting risk.
- subsample: subsample ratio of the training instances. Increasing this value may increase overfitting risk.
- colsample_bytree: the subsample ratio of columns when constructing trees. Increasing this value may increase overfitting risk.

The proper range of the value of the parameter is listed in the table below.
Table 3. Parameters and selected range of parameter values for XGB algorithm

| Parameter          | Selected Values |
|--------------------|-----------------|
| max_depth          | [3, 5, 7, 9]    |
| min_child_weight   | [0.1, 0.5, 1, 5, 10] |
| alpha              | [0, 0.01, 0.1, 0.5, 1] |
| lambda             | [0, 0.01, 0.1, 0.5, 1] |
| subsample          | [0.5, 0.7, 0.9, 1] |
| colsample_bytree   | [0.5, 0.7, 0.9, 1] |

The auto-tuning method used for the above-mentioned three algorithms is Grid Search, which iterates all possible combinations of parameters to setup models, using training set to train each models, and using validation set to evaluate the accuracy. Grid Search is very efficient in estimating proper combinations from a parameter space.

In the tuning process, the optimal model is selected as the one with the highest accuracy. The evaluating metric for accuracy in this research is RMSE (Root Mean Square Error), which can be expressed as follows:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum (y_i - \hat{y}_i)^2}$$  \hspace{1cm} (1)

3. Case Study

In order to evaluate the prediction performance of the auto-tuning method, a case of cooling load prediction is introduced in this section. The case building is a residential building located in Nanjing, Jiangsu Province of China. The building is equipped with central cooling system, and thus cooling load data can be calculated from the monitored water flow and supply/return chilled water temperature. The monitored cooling load data covers a time span from Jun 20th, 2014 to Jul 23rd, 2014. Associated input parameter can also be acquired from other sources, including outdoor air temperature, relative humidity, solar radiation, hour indicator and weekday indicator of the corresponding time span.

In this case, the cooling load from the previous 33 days is used to train and select the optimal case model, and the cooling load of Jul 23rd, 2014 is used to evaluate the predicting results. For each of the above mentioned 3 machine learning algorithms, 70% of the previous 33-day data are randomly selected as the training set, while the rest 30% are selected as the validation set. The optimal model is selected as the one with the least validation RMSE (V-RMSE).

Using the Grid Search method mentioned above, the models with optimal parameter combination is selected. Additionally, a time-series based algorithm named Holt-Winter is introduced in the comparison of the predicting effect. The predicted daily cooling loads of the case building for the last day (Jul 23rd, 2014) as well as the real load curve is illustrated in the figure below.

![Cooling load prediction results of cooling loads for different algorithms in case building](image)
The metrics for evaluating the final accuracy of the model is the prediction RMSE (P-RMSE). The detailed metrics (both V-RMSE and P-RMSE) of each algorithm is listed in the table below.

| Algorithm       | V-RMSE (kW) | P-RMSE (kW) |
|-----------------|-------------|-------------|
| Holt-Winter     | 215.48      | 390.3       |
| MLP             | 165.48      | 181.6       |
| SVR             | 164.62      | 187.2       |
| XGB             | 148.50      | 208.9       |

From the figure and statistics above, it can be inferred that, after auto-tuning of the model parameters, the 3 machine learning algorithms achieve higher accuracy than traditional methods, suggesting that the auto-tuned machine learning algorithms significantly outperform the traditional Holt-Winter algorithm.

4. Discussion

In previous sections, Grid Search algorithm is used for tuning. The optimal parameter combination for the model is global optimum, which is the model with the highest accuracy in the parameter space. However, searching the global optimum is always time-consuming and cost-inefficient. In this section, an approach of searching local optimum is proposed. The advantage of its time efficiency and disadvantage of inaccuracy is evaluated in this part.

The kernel of the proposed approach is to reduce the dimension of the parameter space. This approach decomposes the high-dimension parameter space into multiple low-dimension parameter spaces, and search optimum parameter combination in each low-dimension space consequently. An n-dimension local optimum refers to the result of consequent Grid Searches within parameter spaces that don’t exceed n dimension. Taking XGB algorithm as example. The six parameters (denoted as P1, P2, P3, P4, P5, P6) that remain to be tuned for XGB model will form a six-dimension parameter space. The result of the direct Grid Search in this six-dimension parameter space (P1-P2-P3-P4-P5-P6 space) is global optimum. While the result of consequent Grid Searches in 3 two-dimension parameter spaces (P1-P2 space, P3-P4 space and P5-P6 space) is two-dimension local optimum.

In this part of analysis, the fore mentioned building is still used as case study. XGB algorithm is selected in this case to investigate the difference of accuracy and efficiency between global optimum and local optimum. The comparison is conducted between global optimum with 5 local optimums.

From the accuracy perspective, the V-RMSE for each optimum is shown in the figure below. It can be inferred from the result that global optimum has significantly lower error than all local optimum. The difference between different local optimum is much smaller than that between local optimum and global optimum.

![Figure 5. V-RMSE for different local optimum (LO) and global optimum (GO)](image)

From the efficiency perspective, the time consumed in the tuning process for each optimum is shown in the table below. With the increase of the dimension, the time consumed explodes sharply. The time
consumed for searching global optimum is more than 500 times greater than the 1-dimension local optimum.

| Table 5. Computing time for different local optimum (LO) and global optimum (GO) |
|---------------------------------|
| 1-dim LO | 2-dim LO | 3-dim LO | 4-dim LO | 5-dim LO | GO |
| Computing time (s) | 1.36 | 2.96 | 8.72 | 25.07 | 91.52 | 760.89 |

It can be concluded that, if the accuracy of local optimum is insufficient for application, then global optimum might be the best choice. However, if the accuracy of local optimum is sufficient for application, then 1-dimension local optimum is supposed to be used to improve efficiency.

5. Conclusion
This paper proposed a new auto-tuning method for building energy consumption prediction with machine learning algorithms. Three major machine learning algorithms are specifically analysed. From the case of building cooling load prediction, it can be inferred that the auto-tuned data-driven model has significantly higher accuracy than traditional models. This research also discussed the accuracy and computing efficiency of global optimum and local optimum within a high-dimension parameter space of a machine learning algorithm. The result suggests that global optimum represents higher accuracy. Within local optimums, the low-dimension one is sufficient in accuracy and advanced in computing efficiency.

Future research of building energy consumption prediction will mainly focus on improving the efficiency of auto-tuning. Moreover, selection and pre-processing of input parameters are also essential perspectives in the building energy prediction.

Acknowledgement
This research was supported by Beijing Municipal Natural Science Foundation of China and Tsinghua University Tutor Research Fund.

Reference
[1] Amasyali, K. and El-Gohary, N.M., 2018. A review of data-driven building energy consumption prediction studies. Renewable and Sustainable Energy Reviews, 81, pp.1192-1205
[2] Zhao, H.X. and Magoulès, F., 2012. A review on the prediction of building energy consumption. Renewable and Sustainable Energy Reviews, 16(6), pp.3586-3592.
[3] Kwok, S.S., Yuen, R.K. and Lee, E.W., 2011. An intelligent approach to assessing the effect of building occupancy on building cooling load prediction. Building and Environment, 46(8), pp.1681-1690.
[4] Bergstra, J. and Bengio, Y., 2012. Random search for hyper-parameter optimization. Journal of Machine Learning Research, 13(Feb), pp.281-305.
[5] Li, Q., Meng, Q., Cai, J., Yoshino, H. and Mochida, A., 2009. Predicting hourly cooling load in the building: A comparison of support vector machine and different artificial neural networks. Energy Conversion and Management, 50(1), pp.90-96.
[6] Smola, A.J. and Schölkopf, B., 2004. A tutorial on support vector regression. Statistics and computing, 14(3), pp.199-222.
[7] Chen, T. and Guestrin, C., 2016, August. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (pp. 785-794). ACM.