Dynamically engineered multi-modal feature learning for predictions of office building cooling loads

Liu, Yiren; Zhao, Xiangyu; Qin, S. Joe

Published in: Applied Energy

Published: 01/02/2024

Document Version:
Final Published version, also known as Publisher's PDF, Publisher's Final version or Version of Record

License:
CC BY-NC-ND

Publication record in CityU Scholars:
Go to record

Published version (DOI):
10.1016/j.apenergy.2023.122183

Publication details:
Dynamically engineered multi-modal feature learning for predictions of office building cooling loads

Yiren Liu \textsuperscript{a}, Xiangyu Zhao \textsuperscript{a}, S. Joe Qin \textsuperscript{b,∗}

\textsuperscript{a} School of Data Science, City University of Hong Kong, Kowloon, Hong Kong Special Administrative Region
\textsuperscript{b} Institute of Data Science and Department of Computing and Decision Sciences, Lingnan University, Hong Kong Special Administrative Region

1. Introduction

It is reported that 73.2\% carbon emissions are from the energy (electricity, heat, and transport) sector \cite{1}. The entire life-cycle of buildings is responsible directly and indirectly for 37\% of the global energy- and process-related \text{CO}_2 emissions \cite{2}. As the largest energy consumer, the building sector faces great challenge in managing energy consumption load with the long-term target of curbing carbon intensity. Heating, ventilation, and air-conditioning (HVAC) accounts for 38\% of buildings’ energy consumption worldwide \cite{3}, which is even larger in Hong Kong due to the hot weather condition. According to a recent study \cite{4} that examined 30 commercial buildings in Hong Kong, HVAC systems were responsible for consuming an average of 68\% of the energy used in these buildings. Therefore, managing and predicting energy consumption of HVAC systems is of key importance for building energy saving, especially for high-density cities.

Optimal operation and advanced control are regarded as important technologies for reducing energy consumption and improving energy efficiency \cite{5}. Advanced HVAC control strategies can yield up to 60\% savings in energy \cite{6}. In order to optimize energy consumption, predictive control based on cooling load prediction is effective in demand and load reduction \cite{7}. In particular, for chiller plant operation control, cooling load prediction can be a vital method for optimizing the chiller sequencing control \cite{8,9}. With reliable cooling load prediction, temporary variations of cooling load can be ignored to avoid unnecessarily switching on and off HVAC equipment and chillers \cite{9}. Besides, accurate forecasting of cooling load can improve energy efficiency over the common practice of non-predictive operations \cite{10}.

Since cooling load prediction is greatly beneficial for HVAC energy savings, prediction accuracy is the most concerned problem. Building cooling load predictions can be categorized into four primary groups: residential, educational, commercial, and mixed use buildings \cite{11}. Different building types have different operation and occupant patterns, which requires a modeling method that is adaptable to different types of buildings \cite{12}. Differences in buildings’ ages, thermal mass properties, installed HVAC equipment, and occupant behavior \cite{12,13} make it difficult for first-principles based modeling.

With the development of data science and artificial intelligence (AI), several innovative and effective algorithms are applied in recent years and proven to have high accuracy for cooling load prediction. For example, Sekha et al. \cite{14} applied a deep neural network (DNN) to predict cooling load and was shown to be better compared with some other traditional algorithms. Al-Rakhami et al. \cite{15} propose an ensemble learning by applying XGBoost \cite{16} to build an efficient prediction model. This method can be also implemented by AutoGluon \cite{17}, which includes XGBoost for automated machine learning (AutoML). Sajjad et al. \cite{11} propose a multi-output sequential learning model followed by utility preprocessing with a unified framework. Fan et al. \cite{8}
propose an efficient regression model based on sensitivity analysis and the traditional autoregressive with exogenous input model. Wang et al. [18] propose a twofold algorithm, which first uses Long Short-term Memory (LSTM) [19] for short-term load prediction, then uses XGBoost for long-term load prediction. In addition, Simon et al. [20] demonstrate that building occupancy area and rate play a significant role in building cooling load prediction accuracy.

In a recent literature survey on building energy prediction using artificial neural networks (ANN) [21], LSTM has attracted the most attention among all RNN models. Gated Recurrent Unit (GRU) [22], as a simpler variant of LSTM, is reported to often achieve comparable or better performance than LSTM. For example, works in [23,24] apply LSTM and GRU on building energy prediction tasks, and GRU outperforms LSTM and other conventional models. Recently, Li et al. [25] propose an attention mechanism of RNN to understand the building thermal dynamics. Combined with a multi-input-multi-output strategy, 24-hour ahead building cooling loads are predicted in one go.

While previous studies have offered various techniques for predicting cooling loads, it is essential to conduct detailed analysis and engineering work for a specific building problem as the cooling load patterns can vary substantially. In this work, we propose a dynamically engineered multi-modal feature learning (DEMMFL) model for cooling load prediction with long-term prediction accuracy, even if only a few weather features are provided. The DEMMFL model is estimated with the Lasso-ridge regression [26] and compared with other well-known methods such as the Lasso. Our dynamic model is designed to focus on long-term predictions as the AI Challenge assessment criterion is set on three-month long prediction accuracy. In the field of system identification this is known as simulation errors since it cannot use past values of measured output [27].

To the best of the authors' knowledge, this feature engineered DEMMFL model is new for building cooling load prediction. An independent competition work [28] mentioned the importance of feature engineering for building energy prediction, but that work is focused on hybrid approach for cross-building load predictions. Further, our proposed model will be shown to outperform deep learning models such as LSTM, GRU, and AutoGluon based on data from this Global AI Challenge. As an anecdote for its superiority to competitive machine learning models, our model was awarded with a Grand Prize by the panel of experts in the Global AI Challenge organized by EMSD of Hong Kong SAR.

The rest of the paper is organized as follows. Section 2 introduces the AI Challenge problem and exploratory analysis of data from two office buildings in Hong Kong. Section 3 gives the detail of the proposed DEMMFL method based on control system knowledge and feature engineering. Section 4 applies the method to the data from the AI Challenge for cooling load prediction of the buildings. Section 5 reports and interprets the modeling and prediction results by comparing the proposed model to deep learning and other statistical learning methods. Section 6 concludes the paper.

2. Problem description and exploratory analysis

2.1. The Global AI Challenge Problem

The Global AI Challenge Problem for Building E&M Facilities is posted by the Hong Kong Electrical and Mechanical Services Department (EMSD) and Guangdong Provincial Association for Science and Technology (https://globalaichallenge.com/en/competition/). Fig. 1 depicts the two buildings with a common chilled water system for the HVAC system cooling needs. Specifically, there are three 3517 kW water-cooled chillers, one 3157 kW water-cooled heat recovery chiller, two 1759 kW water-cooled oil free chillers, one 1500 kW water-to-water heat pump, and four 4244 kW and two 2122 kW cooling towers. In normal operations, some chillers can be turned on or off in real-time depending on the cooling demand, which generates multi-model operations that make the cooling load prediction challenging.

The two buildings have overall 1,237,850 square feet of constructed areas with a connected basement. There are 15 stories in the North Tower (NT) and 17 stories in the South Tower (ST). The cooling tower is on the top of the South Tower. As a result, the cooling effectiveness could be different between the two towers even though they share the same cooling system. The buildings have built-in green and sustainable features since its completion in 2019, which is a well-instrumented testbed for AI applications.

The Global AI Challenge problem was announced in Oct. 2021 with the available input and output data from Apr. 1, 2020 to Sep 30, 2021. The challenge is to predict the cooling load of the two buildings for the subsequent months of Oct. - Dec. 2021 based on input data only. The given input and output variables are listed in Table 1. The cooling load is calculated with

\[ \text{CoolingLoad} = mC \Delta T \]  

where \( m \) is the mass flow rate of water (kg/s), \( C \) is the heat capacity (i.e., 4.19 kJ/kg\(^\circ\)C), and \( \Delta T \) is the temperature difference between the returned and supply chilled water (\(^\circ\)C).

All data were collected with a sampling interval of 15 min. The evaluation performance index is the predicted root mean square error (RMSE) of the cooling load per hour from Oct. 1 to Dec. 31, 2021, which were not available at the time of the announcement of the challenge. The predictive model should minimize long-range prediction errors, which is known as simulation errors in system identification [27]. Therefore, we adopt a convolution model structure without using lagged output as model input. Further, since the training data are from April 1, 2020 to Sep 30, 2021, the COVID-induced work-from-home and social distancing restrictions made the building operations extremely irregular with unpredictable occupancy loads. Missing data and outliers are abundant in the training dataset. These issues increased the difficulty of the cooling load prediction problem. To address the missing information in occupancy loads and the number of active chillers, etc., we propose to employ feature engineering and multi-mode models depending on the day of the week and operation hours of the day.

3. Dynamically engineered multi-modal feature learning

The four input features given in Table 1 are limited solely to weather variables, although they are useful for predicting the cooling loads. For example, it is well known that people occupancy [13] has a significant

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Chiller system tag names, descriptions, and variable names.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tag name, unit</td>
<td>Description</td>
</tr>
<tr>
<td>CP.ST.CHWS.EM:</td>
<td>Chilled water supply flowrate</td>
</tr>
<tr>
<td>FLOWRATE, L/s</td>
<td>from South Tower header pipe</td>
</tr>
<tr>
<td>CP.ST.CHWS. TEMP, °C</td>
<td>Temperature of chilled water</td>
</tr>
<tr>
<td>supplied to South Tower</td>
<td></td>
</tr>
<tr>
<td>CP.ST.CHWR. TEMP, °C</td>
<td>Temperature of chilled water</td>
</tr>
<tr>
<td>supplied from South Tower header pipe</td>
<td></td>
</tr>
<tr>
<td>CP.NT.CHWS.EM:</td>
<td>Chilled water supply flowrate</td>
</tr>
<tr>
<td>FLOWRATE, L/s</td>
<td>from North Tower</td>
</tr>
<tr>
<td>CP.NT.CHWS. TEMP, °C</td>
<td>Temperature of chilled water</td>
</tr>
<tr>
<td>supplied to North Tower</td>
<td></td>
</tr>
<tr>
<td>CP.NT.CHWR. TEMP, °C</td>
<td>Temperature of chilled water</td>
</tr>
<tr>
<td>ST.RE.OUTDOOR. TEMP, °C</td>
<td>Outdoor air temperature (OAT)</td>
</tr>
<tr>
<td>at South Tower roof</td>
<td></td>
</tr>
<tr>
<td>ST.RE.OUTDOOR. HUMV, %</td>
<td>Relative outdoor air humidity</td>
</tr>
<tr>
<td>at South Tower roof</td>
<td></td>
</tr>
<tr>
<td>ST.RE.OUTDOOR. RAIN, mm</td>
<td>Precipitation at South Tower roof</td>
</tr>
<tr>
<td>ST.RE.OUTDOOR.UV</td>
<td>UV index at South Tower roof</td>
</tr>
</tbody>
</table>
Fig. 1. Chilled water system for the two office buildings.

**Legend**
1. Chilled Water System
2. Condensing Water System Pump
3. Chilled Water System Pump
4. Cooling Tower System
5. South Tower – Chilled Water Supply Main Header (CP.ST-CHWS)
6. North Tower – Chilled Water Supply Main Header (CP.NT-CHWS)
7. South Tower – Chilled Water Return Main Header (CP.ST-CHWR)
8. North Tower – Chilled Water Return Main Header (CP.NT-CHWR)
9. Condensing Water Return Header
10. Condensing Water Supply Header

impact on cooling loads, but they are not given. Other important features, such as the daily operation schedule with the on/off times of the chillers, are also not given. We apply feature engineering to create dynamic features to make up for the lack of critical features. By examining the operation patterns of the load curves and using the day of the week, we can engineer features that consider intra-day operation modes, the day types in a week, and holidays. Since building cooling loads have fast and slow dynamics due to building thermal mass, multi-scale dynamic features will be engineered based on dynamic control system knowledge. These engineered features compensate for the lack of detailed chiller and HVAC operation information.

### 3.1. Exploring intra-day operation mode features

Office buildings are operated with weekly cycles and daily cycles where daytime and nighttime operations are drastically different. Therefore, it is possible to extract intra-day and weekly operation modes based on cooling loads. Fig. 2 shows the traces of the total cooling loads per day on July 1–Sep. 30, 2021. The traces show that the cooling load in a day can be partitioned into three modes, which are Off-hours (yellow zone) for night time, On-hours (red zone) for day time, and Shutting-off (green zone) for the transition from On-hours to Off-hours. We divide the operation modes in a day as follows.

- The mode of Off-hours (yellow zone) is when the buildings are closed for operations from the evening of one day to the early morning of the next day. This mode is similar across all days of a week.
- The mode of On-hours (white zone) is the main operation mode that has high cooling demand. The load increases from Off-hours to On-hours from around 7:00 A.M. to 8:30 A.M. on weekdays requires turning on one or more chillers to supply the cooling load. Peak surges are sometimes observed due to the turning-on of the chillers [29]. This transition phase is similar to step
3.2. Exploring day-type features

Fig. 3 depicts five weeks’ cooling load trend from May-16-2021 to Jun-19-2021 as they are representative of various day-types. The black and red curves are the real cooling loads of ST and NT, while the purple bars show the Outdoor Air Temperature (OAT). It is observed that different days of the week have different trends and the loads of NT and ST towers bear different patterns. Mondays tend to have a higher surge peak at the start of the On-hours mode, because the buildings are not cooled in the past weekends. On the other hand, Fridays tend to have a relatively low morning peak, especially for NT. Therefore, the seven days of the week are created as day-type features from IsMon, IsTue, …, IsSun. The day-type features take a value of one for the day it represents, and a value of zero otherwise.
In addition, a holiday that happens on weekday has a very different load profile than other workdays. Therefore, holidays that happened on weekdays are treated as a separate day-type, \( \text{IsHol} \). Fig. 3 also illustrates how two holidays that occur on weekdays exhibit distinct behaviors. Specifically, May 19 (Birthday of the Buddha) behaves like a weekend, while Jun 14 (Tuen Ng Festival) shows a reduction in intensity but remains a typical workday pattern. This brings the challenge for Holiday On-hours mode. If we have information on people load during holidays, we could make better estimates for the Holiday On-hours mode.

### 3.3. Exploring nonlinear features

After defining the features of modes of a day and day-types, we explore the relationship between input variables and Cooling Load for each mode. Fig. 4 shows boxplots of Cooling Load vs. each input that is subdivided into multiple levels. When there is an interaction term \[ \text{[31]} \], the effect of one variable depends on the level of the other variable in the interaction. The computed correlation coefficients reveal significant correlations between OAT and Cooling Load, with workday On-hours having the highest correlation coefficients. Furthermore, the charts in the OAT column show a nonlinear relationship between Cooling Load and OAT. This nonlinear relationship is also found in some other papers \[ \text{[32,33]} \]. Therefore, a quadratic function of OAT will be used as a nonlinear feature in this work.

In addition, Humidity is positively correlated with Cooling Load across all modes. UV_Index shows a relatively high correlation with Cooling Load during On-hours, but the correlation is insignificant during Shutting-off and Off-hours mode. This is reasonable since the UV_Index indicates the radiation effect during On-hours. The relationship between Rainfall and Cooling Load is nearly negligible. However, it is widely recognized that Rainfall highly influences the other three variables. Therefore, All four input variables are included in the Cooling Load model.

### 3.4. Engineering dynamic features

From the positive correlations between OAT and Cooling Load depicted in Fig. 4, it is straightforward to think that OAT should be used as a model input. However, the On-hours Cooling Load responses in Fig. 3 clearly show an upward step response at the start of the On-hours mode and a downward step response at the end of it, while the OAT trends do not show such step-like features. Therefore, OAT before and after the On-hours is not as relevant to the Cooling Load as it is for the On-hours. We propose an \( \text{OAT-like} \) feature to reflect both aforementioned effects.

Fig. 5 depicts the procedure of creating the \( \text{OAT-like} \) feature. First, we build a rectangle function that takes a value of one during the On-hours and zero otherwise. To match the baseline with the night mode when the heavy duty chillers are turned off, a baseline temperature, \( \text{OAT-base} \), e.g., the winter average temperature, 12 °C, is used to threshold OAT with the standard ReLU function. Finally, the \( \text{OAT-like} \) feature is created by multiplying threshold OAT with the rectangle
function, that is,
\[
\text{OAT-like} = \text{ReLU}(\text{OAT} - \text{OAT-base}) + \text{OAT-base}
\]
during the On-hours and 0 otherwise. The advantage of the OAT-like feature is twofold. First, when OAT is higher than the baseline temperature, step-like features are created in \(\text{OAT-like}\), as shown in Fig. 5, which serves as a step input to the dynamic Cooling Load model to learn the step-like responses. Second, when OAT is lower than the baseline temperature as in cold winters, the heavy-duty chillers do not need to turn on even during the On-hours; therefore, no step-like features appear in \(\text{OAT-like}\).

The \(\text{OAT-like}\) will be used as a causal input to predict Cooling Load as the following convolution model,
\[
y(t) = \sum_{k=0}^{r} h(k) u_{t-1}^{\text{like}}(t-k) + \text{Other terms}
\]
(2)
where \(y(t)\) is the Cooling Load at the current time \(t\), \(u_{t-1}^{\text{like}}(t-k)\) are OAT-like with time lag \(k\), and \(r\) is the number of lags. Fig. 6 gives an illustration of the convolution operation for the \(\text{OAT-like}\) feature. At \(t = 6:45\), the past OAT-like values are zero, making no impact on Cooling Load. At \(t = 7:00\) and subsequent time intervals, the step-up OAT-like feature will create a surge in the Cooling Load response with compounded memory of the past horizon.

In addition, the \(h(k)\) coefficient in (2) must depend on day-types. Therefore, it is further specified with nonlinear interaction terms as
\[
y(t) = \sum_{k=0}^{r} \sum_{d=0}^{d} \sum_{j=1}^{j} j_{dij} \text{Day}_j \times u_{t-1}^{\text{like}}(t-k) + \text{Other terms}
\]
(3)
with \(d\) as the number of day-types in the model and \(j_{dij}\) as the coefficients.

### 3.5. Engineering slow thermal-mass features

OAT as the main source of building heat is the most important factor to affect cooling load consumption. Building thermal mass can store energy transferred through the building envelope, walls, and roofs exposed directly to ambient temperature [34]. The stored heat in the building thermal mass can take up to several days to release between the thermal mass and the indoor air [34]. Therefore, several slow thermal mass features are created, including Early AM Average OAT, Day Ago Average OAT, Two-days Ago Average OAT, and Previous 10 Days Average OAT.

### 3.6. Summary of all features

Table 2 gives a detailed summary of all original and engineered features for the cooling load prediction in this work, where fast dynamic features of one to two hour lags are included and denoted as \(u_{t-1}^{\text{like}}(t)\).

### 4. DEMMFL modeling via statistical and deep learning

#### 4.1. Deep learning and DEMMFL models

Based on the original and engineered features in Table 2, the prediction model of the cooling load is configured as
\[
y(t) = \varphi \left( \{ u_{i=1}^{12}, u_{t-1}^{\text{like}}(t-k) \} t_{i=1}^{10} \times \text{Day}_j t_{j=1}^{d} + \epsilon(t) \right)
\]
(4)
where \(\varphi(.)\) maps the input features to the output and \(\epsilon(t)\) is the model error term. To achieve long-term prediction accuracy, the model structure is a convolution of past inputs without autoregressive output terms. The mapping can represent a deep neural network, such as the LSTM, GRU, and AutoGluon. For the proposed DEMMFL model, \(\varphi(.)\) takes the following form by extending (3),
\[
y(t) = \sum_{i=1}^{12} \delta_i u_{i=1}^{12} + \sum_{k=0}^{r} \sum_{d=0}^{d} \sum_{j=1}^{j} j_{dij} \text{Day}_j \times u_{t-1}^{\text{like}}(t-k) + \sum_{j=1}^{d} \delta_j \text{Day}_j + \epsilon(t)
\]
(5)
where \(\delta_j\) acts as the intercept when \(\text{Day}_j = 1\) but has no effect when \(\text{Day}_j = 0\).

Since the Day-type variables are binary and mutually exclusive, only one day-type is active among the \(\text{Day}_j\) related features. Therefore, a maximum of \((r + 2)\) features have non-zero values among \(\text{Day}_j\) related features. As a result, there are at most \((r + 14)\) active features in the model.

Fig. 7 shows the model configurations with the original and engineered features. For the DEMMFL model, all features are used for both On-hours and Shutting-off modes. The Off-hours mode excludes the nonlinear and dynamic features because this mode experiences no step changes. On the other hand, for the LSTM, GRU, and AutoGluon models, nonlinearity and dynamics are left for the deep neural networks to learn, since they are regarded as dynamic and universal approximators [35]. Due to the differences between the NT and ST buildings, they
4.2. Estimating the DEMMFL models

The DEMMFL model (5) is linear in parameters. Therefore, it can be estimated with regularized statistical learning methods [36] to achieve the best variance and bias trade-off for predictions. We employ the Lasso [37], ridge [38], and the recently developed Lasso-ridge [26, 39] regression with cross validation (CV) to optimize the learning hyper-parameters. Let

\[ \beta = \text{vec}(\{\beta_i\}_{i=1}^{12}, \{\{\gamma_{jk}\}_{j=1}^{40} \}_{k=1}^{40}, \{\delta_i\}_{i=1}^{40}) \in \mathbb{R}^p \]

\[ x(t) = \text{vec}(\{u_j(t)\}_{j=1}^{12}, \{\{\text{Day}_j\}, \{\text{Week}_j\}_{k=1}^{40} \}_{k=1}^{40}, \{\text{Day}_j\}_{j=1}^{40}) \in \mathbb{R}^p \]

include all features listed in (5), where \( p \) is the number of unknown parameters to be estimated. These features and Cooling Load are scaled into [0, 1] for \( N \) samples in the training set. The DEMMFL model (5) becomes

\[ y(t) = x(t)^\top \beta + \nu(t). \]  

The Lasso regularizes the \( \ell_1 \) norm of the unknown coefficients as

\[ \hat{\beta}_{\text{Lasso}} = \arg \min_{\beta} \frac{1}{2N} \sum_{i=1}^{N} (y(t) - x(t)^\top \beta)^2 + \lambda \|eta\|_1. \]

where \( \lambda \) is the hyperparameter to be tuned by cross validation.

The ridge regression applies the \( \ell_2 \) norm of the coefficients as

\[ \hat{\beta}_{\text{ridge}} = \arg \min_{\beta} \frac{1}{N} \sum_{i=1}^{N} (y(t) - x(t)^\top \beta)^2 + \mu \|eta\|_2^2. \]

where \( \mu \) is the hyperparameter to be tuned with cross validation.

The Lasso uses one \( \lambda \) value to drive some variables to zero coefficients while the same \( \lambda \) value is applied to estimate the nonzero coefficients. It has been pointed out [26,40] that the two tasks do not necessarily share the same optimal \( \lambda \) value. First, there is the \( \lambda \) for the \( \ell_1 \) penalty that leads to the optimal selection of variables. Once the subset of variables is selected, one should re-tune the hyperparameter to achieve optimal estimates for the non-zero coefficients.

The Lasso-Ridge in [26] is an improved two-step sparse learning model of each subset of variables from the Lasso step. The ridge hyperparameter is tuned via cross validation for each subset of variables. The subset of variables that gives the overall cross validation error is chosen as the best model. The details of Lasso-ridge are given as follows.

Step 1: Use all training data to estimate \( \hat{\beta}_{\text{Lasso}} \) in Lasso solution (7) for a grid of \( \lambda \) values. The subsets of selected variables are denoted as \( x^{1}(t), x^{2}(t), \ldots, x^{m}(t) \), without repetition, where \( x^{i}(t) \) is the vector of selected variables in Subset \( i \).

Step 2: Divide the training set into \( s \) fold to perform cross validation. Estimate the \( j \)th fold ridge regression model using the training set \( T_j \) with \( N_j \) observations and the remaining samples as the \( j \)th validation set \( V_j \). The ridge solution for the model with predictors \( \{x^{i}(t)\}_{i=1}^{m} \) is obtained from

\[ \hat{\beta}_{\mu}^{(j)} = \arg \min_{\beta} \frac{1}{N_j} \sum_{i=1}^{N_j} (y(t) - \beta^\top x^{i}(t))^2 + \nu \|eta\|_2^2. \]

Then for Subset \( i \) calculate the mean squared error predicted (MSEP) as

\[ \text{MSEP}_{\nu}^{(i)} = \frac{1}{N} \sum_{j=1}^{s} \sum_{t \in T_j} (y(t) - (\hat{\beta}_{\nu}^{(j)} \top x^{i}(t))^2. \]

and find the \((i^*, \nu^*)\) that yields the smallest MSEP, which is denoted as \( \text{MSEP}_{\nu^*}^{(i^*)} \). The final optimal model is obtained by applying ridge regression Eq. (8) using all training set with variables in Subset \( i^* \) and hyperparameter \( \nu^* \). The optimal coefficients are denoted as \( \hat{\beta}_{\nu^*}^{(i^*)} \).

4.3. LSTM and GRU models

A recent survey [21] reveals that 324 papers were published in the last five years that utilize neural networks for building energy prediction. LSTM and GRU are popular structures exhibiting improved performance. For this reason, LSTM and GRU are implemented and reported in [41] to facilitate comparative analysis. Fig. 8 shows how the input features are used for LSTM and GRU models.

LSTM uses three types of gates, the forget gate \( f(t) \), input gate \( i(t) \), and output gate \( o(t) \), which give the cell state \( C(t) \) and hidden unit state \( h(t) \) as

\[ C(t) = f(t) \cdot C(t-1) + i(t) \cdot \sigma_h(W_c x(t) + U_i h(t-1) + b_i) \]

\[ h(t) = o(t) \cdot \sigma_h(C(t)) \]

where

\[ o(t) = \sigma(W_o x(t) + U_o h(t-1) + b_o) \]
is the output gate implemented with one layer of recurrent network, while the input gate and forget gate are implemented similarly, $\sigma(\cdot)$ is usually chosen as a sigmoidal function whose range is between 0 and 1, while $\sigma_h(\cdot)$ is the hyperbolic tangent. The asterisk * denotes an element by element product.

GRU employs a reset gate and a update gate to give the hidden unit output as [42]

$$
\begin{align*}
\hat{h}(t) &= \sigma(x(W_h x(t)) + U_h (r(t) + h(t-1)) + b_h) \\
h(t) &= z(t) \cdot h(t-1) + (1 - z(t)) \cdot \hat{h}(t)
\end{align*}
\tag{13}
$$

where the update gate $z(t)$ and reset gate $r(t)$ take a similar recurrent network layer to (12).

4.4. AutoML via AutoGluon

AutoGluon is an AutoML library that streamlines deep learning tasks. It encompasses a range of renowned gradient boosting algorithms, fundamental ANN models, and an ensemble model. Notably, AutoGluon excels at training diverse models within specified time constraints. For instance, AutoGluon can efficiently train multiple models within a specified training time. It generates a leaderboard that ranks the performance of each model, with the highest-scoring model as the final selection.

Previous studies [18,43] and a recent paper [44] have underscored the effectiveness of gradient boosting algorithms in predicting cooling load. Moreover, in the ASHRAE Kaggle competition for cooling load prediction (https://www.kaggle.com/competitions/ashrae-energy-prediction), the most successful participants achieved notable results through the utilization of boosting algorithms such as LightGBM and XGBoost. Consequently, considering the gradient boosting algorithm as a baseline is reasonable. In our present research, we replace the boosting algorithm with AutoGluon, which seamlessly incorporates these boosting algorithms.

5. Experimental results on two Government office buildings

5.1. Outline of the research tasks

Fig. 9 depicts the research tasks for cooling load prediction in this study. First, the cooling load data is preprocessed where outliers and abnormal values are removed. To explore the internal relationships between the original features and Cooling Load, correlation analysis is used as a basis for feature engineering. Then statistical DEMMFL and deep learning methods are compared for long-term prediction performance. Train/validation or cross-validation is applied to optimize the hyper-parameters. A testing dataset composed of data from Sept. 2021 is set aside to benchmark the model performance. The prediction RMSE for each method is calculated to evaluate their performance. Additionally, we perform a sensitivity analysis on the models with respect to the OAT feature to check its compliance with basic physics. It is expected that as OAT increases, cooling load consumption will also increase.

5.2. Optimizing the DEMMFL model

We construct multi-mode models for the NT and ST separately. Based on Fig. 2, three operation periods in a day are separated by 7:00 (On-time), 18:00 (Off-time), and 20:00 (Night time). The On-hours mode is further divided into three sub-modes based on Workday, Weekend, and Holiday. The engineered features are composed as outlined in Table 2. Within each mode, we normalize the input and output variables using min–max normalization, scaling them to the range of [0, 1]. After obtaining the predicted Cooling Load, we scale them back to calculate the RMSE. The detailed implementation of each mode of DEMMFL is presented below.

- **Workday On-hours (7:00–17:45)**
  Throughout most workdays, Cooling Load starts to rise around 7:00 and often shows an overshoot around 8:30. The $\text{OAT - like}$ feature is designed to capture the above phenomenon. A close examination of Fig. 2 reveals that a small portion of days tend to have an earlier start, and Fig. 3 indicates that these early starts occur to the North Tower on Fridays.

- **Weekend On-hours (7:00–17:45)**
  Weekends, despite being non-working days, exhibit increased Cooling Load during daytime as shown in Fig. 3. This distinguishes them from nighttime hours. Additionally, Saturday shows slightly higher consumption compared to Sunday.

- **Holiday On-hours (7:00–17:45)**
  Holiday On-hours are the most uncertain part of the model, since the people occupancy condition is unknown. It is observed that some holidays alike a partial workday, while others do not, resulting in uncertain conditions. For the On-hour modes, the number of lags for $\text{OAT - like}$ is tuned to be $r = 40$ for optimal performance, which leads to $r + 14 = 54$ active features in the On-hour model. It implies that the effect of the initial step up of $\text{OAT - like}$ at 7:00 can last as long as 10 h. The number of day-type features in the Workday On-hours mode is five while that for the Weekend On-hours mode is two.

- **Shutting-off (18:00–19:45)**
  At 18:00, the chillers also appear to gradually switch off. This results in a fast decrease in Cooling Load until a low state (Off-hours) is reached. The designed $\text{OAT - like}$ feature is also applied to learn the transition. Since this mode only lasts two hours, the
number of lags for OAT − like is set at \( r = 20 \). It has been experimentally determined that the Shutting-off mode should be treated similarly to On-hours with workday, weekend, and holiday. The empirical result shows little gain in model accuracy. Therefore, we decide to use one uniform Shutting-off mode across 8 day-types.

- **Off-hours (20:00 – 6:45 next day)**

This mode covers the night operation, which is uniform across all day-types. Figs. 2 and 3 show that some Cooling Load values have zig-zag patterns, especially on South Tower. This is likely due to the HVAC on–off control, which adds difficulty in the prediction based on 15-min data. To address this issue, the night-time data is averaged on an hourly basis. Nonlinear and dynamic features are excluded from this mode, since the operation during this mode stays in the low-load region.

To estimate (5) for DEMMFL, we employ four regression techniques: ordinary least square (OLS), Lasso, Ridge, and the Lasso-Ridge regression. Hyper-parameters for the Lasso, Ridge, and Lasso-Ridge regression methods were tuned using cross-validation. For consistency, we divide the training set into the same ten consecutive folds for all cross-validation procedures. The hyper-parameters \( \lambda \) and \( \mu \) are optimized over a grid of values. The grid values are defined as \( e^{-g} \), where \( g \) ranges from 0 to 10 with an increment of 0.2.

### 5.3. Deep learning model tuning

We experiment with several deep learning models including AutoML models such as XGBoost and LightBoost [45], realized via AutoGluon, as well as LSTM and GRU. The loss function is chosen as the mean squared error and the training method is ADAM [46]. We randomly set aside 10% of data from the training set as the validation set to optimize hyper-parameters. The LSTM and GRU searched and optimized hyper-parameters are shown in Table 3.

<table>
<thead>
<tr>
<th>Table 3</th>
<th>Grid search result for LSTM &amp; GRU hyper-parameter tuning.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Range</td>
</tr>
<tr>
<td>Number of hidden units</td>
<td>[128, 256, 512]</td>
</tr>
<tr>
<td>Number of RNN layers</td>
<td>[1, 2, 3]</td>
</tr>
<tr>
<td>Number of epochs</td>
<td>50</td>
</tr>
<tr>
<td>Mini-batch size</td>
<td>500</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 4</th>
<th>Test RMSE’s of the four statistical learning methods for DEMMFL.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Workday On-hours RMSE</strong></td>
<td>Train: 15-min</td>
</tr>
<tr>
<td>OLS</td>
<td>380.8</td>
</tr>
<tr>
<td>Ridge</td>
<td>383.3</td>
</tr>
<tr>
<td>Lasso</td>
<td>385.2</td>
</tr>
<tr>
<td>Lasso-ridge</td>
<td>387.3</td>
</tr>
</tbody>
</table>

| Week-end On-hours RMSE | Train: 15-min | Test: 15-min | Train: 1-h | Test: 1-h |
| OLS | 356.3 | 336.6 | 202.2 | 210.1 |
| Ridge | 365.4 | 338.5 | 213.7 | 220.9 |
| Lasso | 360.5 | 336.3 | 205.4 | 217.7 |
| Lasso-ridge | 360.7 | 331.5 | 203.1 | 209.9 |

| Holiday On-hours RMSE | Train: 15-min | Test: 15-min | Train: 1-h | Test: 1-h |
| OLS | 736.7 | 1163.7 | 586.3 | 1047.0 |
| Ridge | 738.8 | 1073.8 | 589.0 | 948.1 |
| Lasso | 739.1 | 1003.2 | 589.4 | 877.1 |
| Lasso-ridge | 758.6 | 813.5 | 607.3 | 674.8 |

| Shutting-off RMSE | Train: 15-min | Test: 15-min | Train: 1-h | Test: 1-h |
| OLS | 347.1 | 395.7 | 176.6 | 209.4 |
| Ridge | 354.7 | 393.6 | 182.5 | 214.7 |
| Lasso | 352.8 | 391.8 | 181.3 | 210.7 |
| Lasso-ridge | 352.6 | 392.3 | 181.7 | 210.3 |

<table>
<thead>
<tr>
<th><strong>Overall RMSE</strong></th>
<th>OLS</th>
<th>Ridge</th>
<th>Lasso</th>
<th>Lasso-ridge</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMSE</td>
<td>236.1</td>
<td>281.1</td>
<td>239.5</td>
<td>274.0</td>
</tr>
</tbody>
</table>
5.4. Experimental results on training and testing datasets

5.4.1. DEMMFL learning results

We compare the performance of statistical learning of the DEMMFL model on the training set and test set for each mode. The test dataset is the whole month of Sept. 2021. This represents long term prediction results, which are specified by the AI Challenge competition. Table 4 shows the RMSE values for 15-minute and 1-hour intervals, respectively. Notably, for Off-hours, only 1-hour RMSE values are given since the data is hourly averaged. The Lasso-ridge outperforms the other three techniques in all modes except Off-hours, where Lasso-ridge is marginally the second best. Lasso-ridge achieves the overall test set performance with 4.2% improvement over the second-best method. In addition, as a sparse learning method, Lasso-ridge achieves higher predictive performance while using fewer variables, which will be given in the sequel.

We give an example of detailed learning of DEMMFL models using the South Tower Workday On-hours mode, where the cross-validated MSEPs for Lasso, ridge, and Lasso-ridge are depicted in Fig. 10. The top panel uses the blue curve to show the Lasso MSEP and circles to show the cardinal of the subset of variables selected by the Lasso for a given $\lambda$ value. The Lasso picks the $\lambda^*$ that yields the smallest MSEP and retrains the model with the whole data set using this $\lambda^*$ value. The Lasso-ridge uses each of all subsets of the selected variables in the Lasso step and then applies ridge regression to optimize over a grid of $\mu$ values with the MSEPs shown as thin curves in the bottom panel. The Subset $i^*$ with the ridge parameter $\mu_{LR}^{i^*}$ value that achieves the overall minimum MSEP across all subsets of variables is chosen as the best subset, with $\mu_{LR}^{i^*}$ shown as the vertical red line. The final Lasso-ridge model is obtained by retraining a ridge model on the whole data set with Subset $i^*$ and the optimal $\mu_{LR}^{i^*}$ value. Additionally, the red circle marks the minimum MSEP ($\text{MSEP}_{L-R}^{i^*}$) achieved with Subset $i^*$ for Lasso-ridge. The cardinal of the optimal subset is shown as the solid red dot in the top panel, which is smaller than that of the Lasso model.

The ridge regression results are shown in the bottom panel of Fig. 10 with the thick blue curve representing the cross-validated MSEP and the vertical blue line representing the optimal $\mu_{\text{ridge}}^{i^*}$ value. Ridge regression can be considered as a special case of the Lasso-ridge that uses all variables. Therefore, as depicted in Table 4, Lasso-ridge achieves the best predictive performance with the best selected variables.

5.4.2. DEMMFL model interpretation

To interpret the models from the Lasso-ridge, Fig. 11 depicts the coefficients for NT (top panel) and ST (bottom panel) Workday On-hours mode. In the left panels, the first 8 variables are related to OAT features. The majority of these variables are selected by the model, indicating the dominance of OAT on cooling load prediction. It is observed that Rainfall is deselected in the models of both towers. In addition, the one-hot variables $\text{IsMon}$ through $\text{IsFri}$ behave differently for the two towers. These variables represent the intercept for each day-type.

The right panels show the OAT-like convolution coefficients for each weekday. Lags without circles are deselected in the model by Lasso-ridge. It is observed that the large positive values occur within the range of six lags (inside the green dashed rectangles), reflecting the duration of the rapid surge period of approximately 1.5 h after On-time. Further observations are given as follows.

1. The shapes of Mon–Fri convolution coefficients within each tower are similar, but they are different between the towers, justifying the separate modeling of NT and ST.
2. For both towers, Friday OAT-like coefficients tend to have lower values than other days, which reflects the fact that Friday has a lower morning surge peak than other days.
3. Only NT on Friday shows positive values for OAT-like with 0 ~ 1 lags, which accounts for the fact of earlier On-time on Fridays.

Beyond six lags, all Mon-Fri OAT-like coefficients are small and some are zero, indicating that the dynamic features successfully learn the cooling load trend as well as the differences between days and towers.

For the Shutting-off mode, all eight day-types are built with one model, whose coefficients are depicted in Fig. 12. In the left panels of the figure, the OAT features again have large magnitudes. However, the selected dynamic OAT features are those with one hour lag and the previous 10-day average OAT only. In the right panels, the convolution coefficients of the first nine lags of Mon-Fri OAT-like (inside green dashed rectangles) are large, while those beyond nine lags are mostly zero. The convolution coefficients for weekends are uniformly small, while those for holidays are somewhat more significant. It is evident that the model coefficients in Figs. 11 and 12 are different due to the different mechanisms in turning-on and turning-off the cooling.

5.4.3. Sensitivity analysis
After establishing the models, we evaluate the sensitivity of Cooling Load with respect to OAT in each model, similar to the work in [33]. We calculate the model output by increasing the OAT by one degree Celsius, and calculate the values of other OAT-related variables accordingly, but keep the remaining variables unchanged. Fig. 13 depicts the resulting changes in Cooling Load for all ten models, which represent the numerical sensitivity. It is observed that ST has higher sensitivity.
than NT, which is not surprising since ST is a larger building. Further, the disparity of the sensitivity between the two towers is increased in all but the On-hours mode.

5.4.4. Deep learning results

The deep learning models via LSTM, GRU, and AutoGluon are optimized on the same training data and compared on the same test set of Sept. 2021, with the hourly predicted RMSE results shown in Table 5. Neural model training is terminated when RMSE of the validation data achieves a minimum to avoid over-training. It is seen from the results that DEMMFL with knowledge-based feature engineering achieves the best RMSE. AutoGluon gives the second best RMSE.

Table 5  
Cooling Load model training RMSE, testing RMSE and CV_{RMSE} of DEMMFL, LSTM, GRU, and AutoGluon.

<table>
<thead>
<tr>
<th></th>
<th>LSTM</th>
<th>GRU</th>
<th>AutoGluon</th>
<th>DEMMFL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hourly RMSE on training set</td>
<td>171</td>
<td>182</td>
<td>117</td>
<td>239</td>
</tr>
<tr>
<td>Hourly RMSE on test set</td>
<td>678</td>
<td>698</td>
<td>415</td>
<td>263</td>
</tr>
<tr>
<td>CV_{RMSE} on test set</td>
<td>23.0%</td>
<td>23.7%</td>
<td>14.1%</td>
<td>8.9%</td>
</tr>
</tbody>
</table>

To put the prediction errors into a relative perspective, the coefficient of variation of RMSE, denoted as CV_{RMSE}, is adopted from [47,48], which is the ratio of RMSE over the average Cooling Load of the test
Fig. 13. Sensitivity of Cooling Load changes with respect to 1°C rise in Average-OAT.

Fig. 14. Detailed prediction results of the four models for the test dataset (Sept. 2021).
set. The $\text{CV}_{\text{RIDGE}}$ values for Sept. 2021 are also given in Table 5. It is seen that the proposed DEMMFL achieves less than 10% $\text{CV}_{\text{RIDGE}}$ with a month-long prediction horizon, which is significantly better than the deep learning results. DEMMFL shows its clear advantage because of the knowledge-driven engineered features.

Fig. 14 depicts the actual versus predicted Cooling Load on the test data of Sept. 2021 by the four methods. LSTM and GRU show visibly poor prediction results on the first three days of the month. Very large prediction errors are seen for the day of Sept. 22 with the deep learning models, which is the Mid-Autumn holiday, but the DEMMFL model predicts very well.

6. Conclusions

With control-knowledge-driven engineered features for building cooling load prediction, this paper develops an effective dynamically engineered multi-modal feature learning method, denoted as DEMMFL, for predicting the cooling load of office buildings. The model’s long-term prediction accuracy is best achieved by applying the Lasso-ridge regression learning. Comparative evaluations with other well-known methods such as the Lasso, LSTM, GRU, and AutoGluon demonstrate the superior performance of the proposed model. By leveraging the modes of operations and control system knowledge, the proposed DEMMFL method generates accurate long-term predictions for Cooling Load. The contribution lies in the novel approach that applies control system knowledge to engineer useful features for long term predictions.

In this case study defined by the Global AI Challenge, the input data consist solely of weather information, which increases the difficulty to achieve an accurate model. However, the DEMMFL model is able to achieve less than 10% coefficient of variation for long term predictions on a test data set. With this AI Challenge being a common situation where the available data are limited, this work shows that knowledge-driven feature engineering outperforms deep learning methods, which are believed to be able to learn important features from the data. The DEMMFL method could further improve the accuracy of predicting the cooling load in holidays if building occupancy load data is available. Future work will apply the method to other commercial buildings with known operation status of chillers and air handling units.

CRedit authorship contribution statement

Yiren Liu: Writing – original draft, Visualization, Software, Formal analysis, Data curation. Xiangyu Zhao: Writing – review & editing, Supervision, Methodology, Investigation. S. Joe Qin: Writing – review & editing, Supervision, Projects, Resource administration, Methodology, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper. The authors declare the manuscript has not been published before nor submitted to another journal for the consideration of publication.

Data availability

The authors do not have permission to share data.

Acknowledgments

The work described in this paper was partially supported by a grant from a General Research Fund by the Research Grants Council (RGC) of Hong Kong SAR, China (Project No. 11303421), a Collaborative Research Fund by RGC of Hong Kong (Project No. C1143-20G), a grant from the Natural Science Foundation of China (U20A20189), a grant from ITF - Guangdong-Hong Kong Technology Cooperation Fund- ing Scheme (Project Ref. No. GHP/145/20), a Math and Application Project (2021YFA1003504) under the National Key R and D Program, a Shenzhen-Hong Kong-Macau Science and Technology Project Category C (9240086), and an InnoHK initiative of The Government of the HKSAR for the Laboratory for AI-Powered Financial Technologies.

The authors gratefully acknowledge the support of the EMSD E&M AI Lab of the Hong Kong SAR for providing the data. The authors acknowledge the assistance of Yixiao Huang, Shenglong Yao, and Guo Han on the LSTM, GRU, and AutoGluon work for comparison.

References


