White Rose Research Online
eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk
Universities of Leeds, Sheffield and York

Deposited via The University of Leeds.

White Rose Research Online URL for this paper:
https://eprints.whiterose.ac.uk/id/eprint/214203/

Version: Accepted Version

Article:
Yu, D., Liu, T., Wang, K. et al. (2024) Transformer based day-ahead cooling load forecasting of hub airport air-conditioning systems with thermal energy storage. Energy and Buildings, 308. 114008. ISSN: 0378-7788

https://doi.org/10.1016/j.enbuild.2024.114008

© 2024 Elsevier B.V. This is an author produced version of an article published in Energy and Buildings. Uploaded in accordance with the publisher's self-archiving policy. This manuscript version is made available under the CC-BY-NC-ND 4.0 license http://creativecommons.org/licenses/by-nc-nd/4.0/.

Reuse
This article is distributed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivs (CC BY-NC-ND) licence. This licence only allows you to download this work and share it with others as long as you credit the authors, but you can’t change the article in any way or use it commercially. More information and the full terms of the licence here: https://creativecommons.org/licenses/

Takedown
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

UNIVERSITY OF LEEDS
University of Sheffield
UNIVERSITY of York

Transformer Based Day-ahead Cooling Load Forecasting of Hub Airport Air-Conditioning Systems with Thermal Energy Storage

Die Yu[a], Tong Liu[b], Kai Wang[a,*], Kang Li[c], Mehmet Mercangöz[b], Jian Zhao[a,d], Yu Lei[d] and RuoFan Zhao[d]

[a] School of Automation, Chongqing University, Chongqing, 400044, China
[b] Department of Chemical Engineering, Imperial College London, London, SW7 2AZ, UK
[c] School of Electronic and Electric Engineering, University of Leeds, Leeds, LS2 9JT, UK
[d] Chongqing Airport group CO., Ltd, Chongqing, 401120, China

ARTICLE INFO

Keywords:
Day-ahead cooling load prediction
weighted-DTW-k-means
Interpretable deep learning model
temporal fusion transformer (TFT)
Performance evaluation

ABSTRACT

The air conditioning system constitutes more than half of the total energy demand in hub airport buildings. To enhance the energy efficiency and to enable intelligent energy management, it is vital to build an accurate cold load prediction model. However, the current models face challenges in dealing with dispersed load patterns and lack interpretability when black box approaches are adopted. To tackle these challenges, we propose a novel k-means-Temporal Fusion Transformer (TFT) based hybrid load prediction model. Specifically, the daily load patterns are grouped using an improved k-means clustering method that considers both input feature weights and dynamic time warping (DTW) distances. Additionally, the statistical features of the clustering output are inputted into the TFT. By further incorporating context information, the integration of data between different schema categories is achieved, thus reducing errors that may occur during the transition process. As a result, the prediction performance and interpretability are significantly improved. The Chongqing Jiangbei Airport T3A terminal is used as a case study, and experiments are conducted using cooling data from the No.1 energy station, as well as the airport traffic data and the meteorological station data. Results are compared with other mainstream models, confirming that the proposed day-ahead load forecasting model achieves improvements in several performance indicators, including MAE, MAPE, CV-RMSE, and R2, which are 384 kW, 3%, 5%, and 0.058 respectively.

1. Introdouction

China has committed to achieving the carbon neutrality target by 2060 to tackle climate change and energy security challenges [1]. Building energy consumption accounts for approximately 23% of the total energy consumption in China [2]. With the rapid growth of China's aviation industry, the cooling demand in airport terminals increased significantly [3]. These terminals are large-scale public transportation buildings with a range of functionality facilities, and air conditioning systems contribute to over 40%-60% of the building's total energy consumption [4, 5, 6]. To enhance energy flexibility, thermal energy storage (TES) systems have been widely deployed in large-scale building complex. These systems store cold thermal energy during off-peak hours at night and provide cooling during peak hours in the day. However, the system has a time lag in regulating the supply of chilled water and the temperature of the return water. Accurate prediction of the cooling load enables system to operate in a "feed-forward" mode, thereby meeting the building's cooling demand in real time. Forecasting the system load for the next day can guide operators in scheduling the operation hence optimizing the system performance, and also assist the planning of the cold thermal storage system and enable the energy saving techniques like grid peak shaving [7]. Model based forecasting also plays an important role in energy management, for example, the model predictive control (MPC) is commonly used for the optimal control in the building energy management system [8]. MPC involves predicting the future behavior of a system based on a mathematical model, and then optimizing the control inputs over a specified time horizon to achieve desired performance [9, 10, 11]. It is clear that the control performance of MPC is heavily reliant on the model accuracy [12, 13, 14]. Therefore, accurately predicting the cooling load in terminal buildings plays a key role in building intelligent, low-carbon energy systems and thus achieving sustainable development of modern airports.

1.1. Literature review

Building cooling load forecasting techniques can be grouped into three categories: physical simulation-based, data-driven, and hybrid [15]. The physical simulation approaches require substantial domain knowledge which are often incorporated into specialized software such as EnergyPlus, TRNSYS, etc [16], and they are often used at the planning stage of building systems. The hybrid approach can improve prediction accuracy by combining physical modeling and data-driven methods, but it is still time-consuming and challenging in adapting to complex situations, and prone to making false assumptions [17]. The data-driven approach, which simplifies the modeling process by using a large amount of operational data, has attracted substantial interests in recent years, including recent uses of machine learning and deep learning algorithms for predicting cooling

*Corresponding author
anna.kria123@gmail.com (D. Yu); tliu.soton@gmail.com (T. Liu); kaiwang@cqu.edu.cn (K. Wang); K.Li1@leeds.ac.uk (K. Li); m.mercangoz@imperial.ac.uk (M. Mercangöz)
ORCID(s):

load [18, 19]. Shallow machine learning models commonly adopted include Support Vector Regression (SVR) [20], Random Forests (RF) [21], Auto Regressive Integrated Moving Average (ARIMA) [22], XGBoost [23], and ensemble model [24]. Deep learning models include ANN, DNN, BP, Long Short-Term Memory (LSTM) [25], Convolutional Neural Networks (CNN) [26], Gated Recurrent Units (GRU), seq2seq [27], as well as the hybrid methods based on these models [28, 19, 29, 30]. Luo et al. [25] developed and compared twelve data-driven models used for predicting building heat loads. It was found that XGBoost and LSTM models yielded the most accurate results among the two categories of models. While traditional models like XGBoost are effective in approximating the nonlinear relationship between inputs and outputs, they do not fully exploit the inherent dependencies of continuous time series data. Previous research has demonstrated that several models can achieve satisfactory results in one-hour-ahead load forecasting. However, multi-step forecasting has yet to be fully researched. Currently, there are three groups of strategies: recursive, direct, and multi-input and multi-output (MIMO). The recursive strategy tends to accumulate a large error, while the direct strategy is computationally intensive and challenging to implement in practical applications [19]. The MIMO strategy effectively utilizes the dynamic and time-dependent characteristics of the sequence to directly generate multi-step prediction vectors. The study by Li et al. [17] suggests that the seq-seq structure is well-suited for multi-step prediction using the MIMO strategy. Furthermore, the performance of this structure can be improved through parameter tuning and the integration of an attention mechanism. Zhang et al. [12] utilized the seq2seq-LSTM model to predict the major operating parameters of a solar heating system. The predicted parameters are then integrated into a heating system operation model for whole system simulation to obtain the control signals for MPC, achieving a significant reduction in the energy consumption of the solar heating system. In recent years, a wide range of algorithms have been developed to improve the interpretability of these models [31]. Liang et al. [32] utilized domain knowledge analysis to decompose the energy consumption pattern into a global part and a local part to improve the interpretability of the model. A deep ensemble model and an ARIMA model were built for the two parts separately in order to create a hybrid prediction model. This model demonstrates significant robustness, though the model updating performance is limited. The interpretability of the model is crucial for users like operators to understand the underlying mechanism [33]. Temporal fusion transformer (TFT) [34], which is a special sequence-to-sequence model with a multi-head attention mechanism has shown to have good interpretability and is also suitable for multi-output prediction.

Furthermore, the load data distribution has significant impact on the forecasting accuracy, largely due to the complex distribution features embedded in the load data rather than the algorithms used [35]. The overall accuracy of the model tends to decrease when trained on heterogeneous sequences. Liu et al. [36] employed 1-D k-means clustering to investigate the gaps between the predicted energy usages and the actual values, thereby establishing a quantitative energy evaluation strategy. To improve the performance of predictive models, the utilization of clustering algorithms can be beneficial [37]. Li et al. [38] used the k-means algorithm to group thermal zones of a building into several clusters to identify the typical zones or centroids of the clusters. The thermal curves for these typical zones are then used as the basis for cross-scale load prediction. Luo et al. [16] used k-means clustering to group daily weather profiles and develop a GA-DNN model for each subgroup in order to enhance the forecasting accuracy by reducing the mean absolute percentage error by 11.9%. Yang et al. [39] employs the k-shape algorithm to analyze the energy usage patterns of buildings. This involves clustering the hourly energy consumption data points for each building and improving the accuracy of the prediction model by combining it with SVR. Zhang and Li [40] introduced a closed-loop clustering approach that utilizes the model fitness signal as the guiding criterion. This method incorporates a feedback mechanism to achieve Mean Squared Error guided clustering. The proposed approach demonstrates promising prediction outcomes on both simulated and real datasets. In a study conducted by Zhang et al. [41], the k-means clustering algorithm was employed to cluster the refrigeration loads of a plant. Furthermore, the k-nearest neighbor algorithm is used to determine the category of the forecasted day, resulting in a 10% improvement in forecasting accuracy. Luo et al. [42] developed a grouping algorithm for similar weather and day type, and proposed a the weather similarity index for selecting similar days which has been shown to improve the forecasting accuracy.

Based on the literature review, the following research gaps are identified. Firstly, the variability in data resulting from diverse energy consumption patterns poses challenges for accurate prediction of building loads. Although clustering techniques have been widely used in energy prediction in addressing the diverse energy consumption patterns, the classic clustering algorithms still have difficulties to the diversity of patterns in the energy data. Secondly, there is a lack of comprehensive framework for delivering interpretable model, which demands novel model architectures. Finally, it is still an open question in building energy forecasting to effectively integrate similar historic energy consumption patterns into an interpretable model.

1.2. Contributions

To address the aforementioned research gaps, this paper has made the following contributions:

1. An enhanced clustering algorithm is proposed to group cooling load data, which enables the extraction and differentiation of different patterns, taking into account the diversity of load patterns as well as the impact of input features and timing characteristics.
2. The clustering results are then fed into an Temporal Fusion Transformer model, which enables multi-step load

prediction using the MIMO strategy and producing interpretable result.
3. The proposed framework is evaluated on real-world airport cooling load data, and the test results are compared with the state-of-art approaches against different performance metrics, confirming the efficacy and superiority of the proposed method.
4. For buildings with diverse load patterns, our method can significantly save computing time and produce more accurate cool load predictions.

The remainder of the paper is organized as follows. Our proposed method is detailed in Section 2. The experimental work is presented in Section 3. Section 4 discusses the results and findings. Section 5 concludes the paper and future work is also discussed.

2. Methodology

Fig.1 provides an overview of the proposed framework. In summary, to deal with the missing values and outliers in the collected data, data preprocessing techniques were employed. The influence of key variables on the cold load was examined using the Pearson correlation coefficient. Additionally, an improved clustering method is developed to identify distinctive load patterns, and an TFT model is built to establish a multi-step output load prediction model for the next day. The accuracy of the predictions is then compared and interpretable results are finally presented.

2.1. Time-series clustering

2.1.1. Weighted-DTW-K-Means

The k-means algorithm was initially introduced by MacQueen in 1967 and has since gained significant popularity in the field of building energy study [16, 41]. The procedure involves partitioning the data into K groups, selecting K objects as the initial clustering centers randomly, and subsequently computing the distance between each object and each clustering center. Based on this distance calculation, each object is assigned to the closest clustering center, effectively forming a cluster. This iterative process continues until the clustering centers no longer undergo any changes. The conventional k-means clustering technique classifies sequences based on the Euclidean distance between individual data points. However, this method may not accurately categorize sequences with temporal offsets. To address this limitation, the Dynamic Time Warping (DTW) algorithm is introduced to evaluate the dynamic similarity of sequences. The DTW algorithm adjusts the time series by shortening or extending it to align different time points and measures the dynamic similarity between sequences by calculating the distance between corresponding points [43]. In this study, the data series are grouped into 24-hour intervals for the purpose of clustering. Specifically, the distance matrix D is a 24x24 matrix, each element $D_{i,j}$ in the matrix represents the distance between the i-th element of the first sequence and the j-th element of the second sequence, using Eq. (1), where $L_{\mathrm{c},1}$ and $L_{\mathrm{c},2}$ represent two typical cooling load series, with $L_{\mathrm{c},1} = (l_{c1,1}, l_{c1,2}, \dots, l_{c1,24})$ and $L_{\mathrm{c},2} = (l_{c2,1}, l_{c2,2}, \dots, l_{c2,24})$. An improved multidimensional clustering algorithm that takes into account the influence of feature data is proposed. The algorithm incorporates a weight factor vector W_x, which was derived from the Pearson correlation coefficient shown in Section 3.3.5, a value greater than 0.4 is considered to be most appropriate according to [44] and hence is adopted in this paper. Thus, the computed value for W_x in this paper is $[1, 0.73, 0.52, 0.45, 0.47]$, representing the cold load, temperature, humidity, solar radiation intensity and passenger flow data respectively. The clustering process utilizes the dynamic time warping (DTW) technique to define the distance, and objective function of weighted-DTW-k-means is given as Eq. (2), which aims to minimize the aggregation of the distance E for each sequence x in relation to the central sequence.

$$
\begin{aligned}
& D_{i,j} = \sqrt{\left(L_{\mathrm{c}1,i} - L_{\mathrm{c}2,j}\right)^2} + \\
& \min\left(D_{i-1,j} + D_{i,j-1} + D_{i-1,j-1}\right) \\
& 1 \leqslant i \leqslant 24, 1 \leqslant j \leqslant 24 \\
& D_{0,0} = 0 \\
& D_{i,0} = +\infty \\
& D_{0,j} = +\infty
\end{aligned} \tag{1}
$$

$$
\underset{u}{\operatorname{argmin}} E = \sum_{i=1}^{K} \sum_{x \in C_i} \mathrm{DTW}(x, u_i) \cdot W_x \tag{2}
$$

The data set is partitioned into K clusters, denoted as $C = \{c_1, c_2 \cdots, \ c_k\}$. For i-th cluster category of the training samples c_i, $u_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$ is the center of c_i.

2.1.2. Cluster number k

The Elbow method is a widely used in determining the optimal number of clusters in k-means clustering [45]. It involves calculating the sum of squared distances between the centroid of each cluster and the data points within that cluster. This measure is referred to as the degree of distortion. A higher degree of distortion suggests a more loosely structured cluster. In general, the degree of distortion decreases as the number of clusters increases. However, for datasets with a certain level of differentiation, there exists a critical number of clusters at which the degree of distortion significantly increases first, and then gradually decreases. This critical number can be considered as a potential number of clusters. By plotting the distortion score as a function of the number of clusters, we can observe the trend of the curve. At a certain inflection point, known as the elbow, the decline in the curve becomes less pronounced. Therefore, the number of clusters at this inflection point is considered to be the appropriate number of clusters. In this study, we present the results of applying the Elbow method to our dataset in Fig.2, where the blue curve illustrates the correlation between the rationscore of explicit images and the number of clusters (k), while the yellow curve depicts the relationship

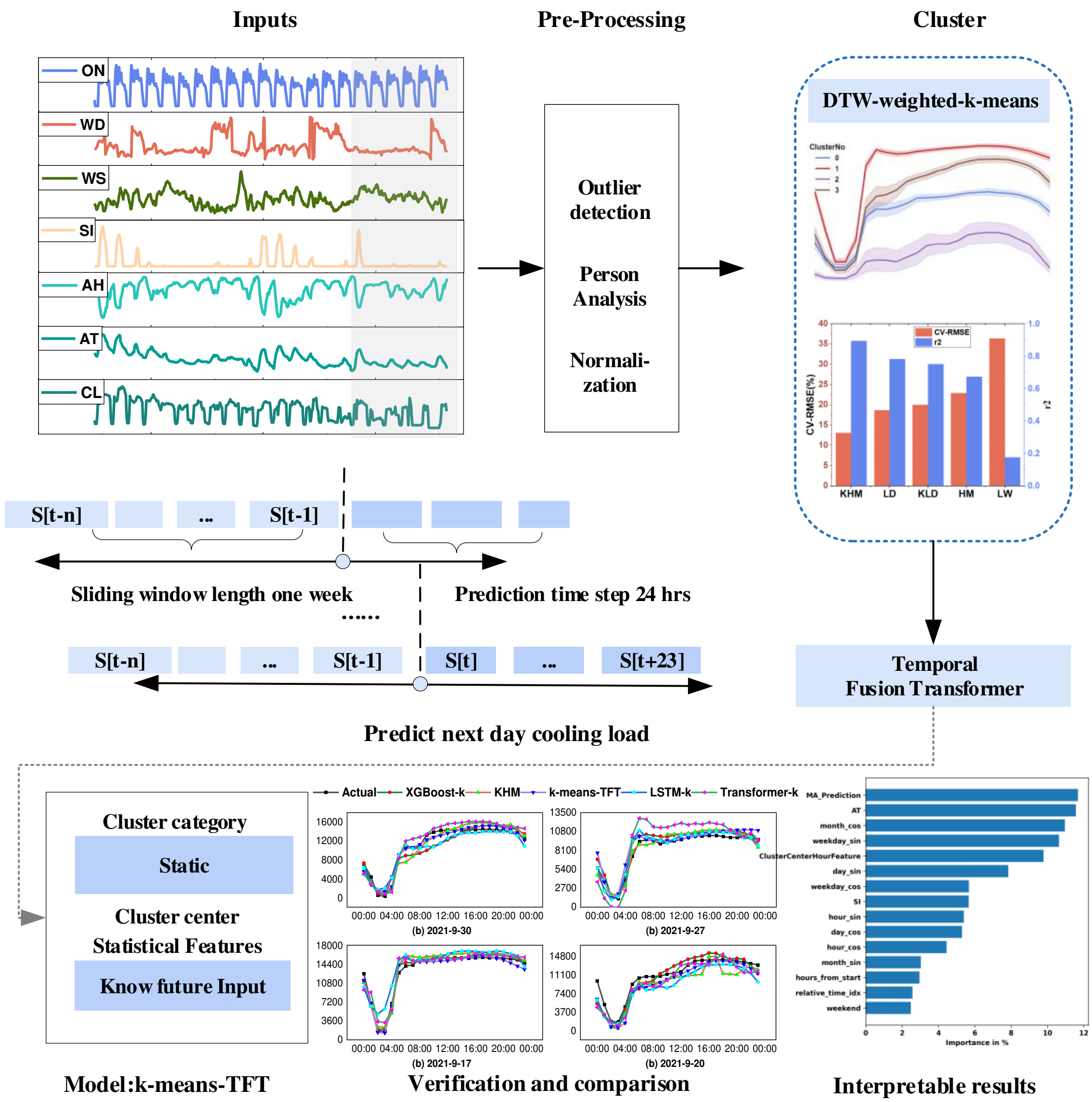

Fig.1. Overview of the proposed framework.

between fitting time and k, the optimal number of clusters is 4, as indicated by the elbow point.

2.1.3. *Pattern recognition of new data*

In the context of predicting future load patterns, the presence of unknown loads presents a challenge. However, in the test dataset utilized in this paper, where the loads are known, it is possible to calculate the distance to the center of mass in order to accurately classify the loads into their respective categories. When the method is deployed in real-world scenarios, it will be imperative to use mapping techniques such as Support Vector Machines (SVM), k-Nearest Neighbors (kNN) [41], and others to effectively handle unknown load patterns.

2.2. Temporal fusion transformer

Transformer-based models have gained significant popularity in the field of time series prediction [46]. These models utilize a self-attention mechanism to capture important features in the input vectors by connecting the encoder and decoder. In our study, we utilize an interpretable temporal fusion transformer to enhance the performance of cold load prediction [34]. Unlike traditional deep learning networks that treat all vectors equally during processing, TFT introduces a specific processing mechanism and network for input vectors in multivariate temporal prediction. This approach is advantageous in multi-horizon prediction. Additionally, TFT offers high interpretability by incorporating a carefully designed gating method for feature selection and a self-attention layer to capture both long-term dependencies and

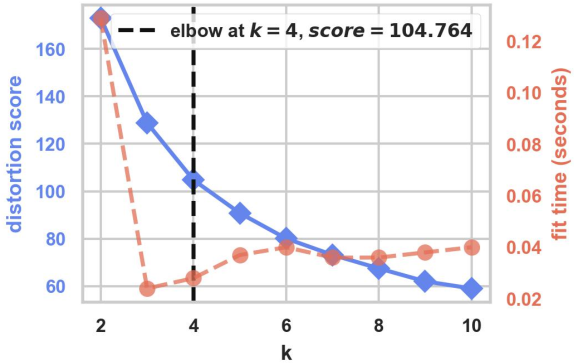

Fig.2. Distortion score and fit time variation with number of chosen clusters.

short-term correlations. These features elevate the model performance in various prediction tasks. The model structure of TFT is illustrated in Fig.3 and consists of five main components: a gating mechanism, a variable selection network, a static covariate encoder, an interpretable multi-head attention, a temporal fusion decoder, and quantile outputs and loss functions.

2.2.1. Gating mechanisms

As pointed out in [34], the gating mechanism can be used to skip over unused components in the architecture, thus enable adaptive adjustment of the depth of the network to accommodate a wide range of datasets and scenarios. In this paper, we use a Gated Recurrent Network (GRN) in Fig.3 to calculate the primary information vector, p, and the context information vector, c, using Eq. (3). The intermediate layers $\eta_1 \in \mathbb{R}^{d_{\text{model}}}$ in Eq. (4) and $\eta_2 \in \mathbb{R}^{d_{\text{model}}}$ in Eq. (5) are utilized, with the shared weights denoted as w. Here, d_{model} is the hidden state size. The ELU function serves as the activation function for the Exponential Linear Unit, as defined in Eq. (6), where x is the input and ∂ is a hyperparameter. The LayerNorm function is used for standard layer normalization, while the Gated Linear Units (GLUs), as shown in Eq. (7), apply a linear transformation to the input γ using a sigmoid activation function $\sigma(\cdot)$ to achieve adaptive depth. Here, $W() \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$ represents the weights and biases, and $\odot$ denotes the element-wise Hadamard product.

$$\text{GRN}_{\omega}(p, c) = \text{LayerNorm}\left(p + \text{GLU}_{\omega}\left(\eta_1\right)\right) \tag{3}$$

$$\eta_1 = W_{1,\omega}\eta_2 + b_{1,\omega} \tag{4}$$

$$\eta_2 = \text{ELU}\left(W_{2,\omega}p + W_{3,\omega}c + b_{2,\omega}\right) \tag{5}$$

$$\text{ELU}(x) = \begin{cases} x, x > 0 \\ \partial\left(e^x - 1\right), x \leq 0 \end{cases} \tag{6}$$

$$\text{GLU}_{\omega}(\gamma) = \sigma\left(W_{4,\omega}\gamma + b_{4,\omega}\right) \odot \left(W_{5,\omega}\gamma + b_{5,\omega}\right) \tag{7}$$

2.2.2. Variable selection networks

The variable selection network utilizes a gating mechanism to identify significant features for vector embedding and eliminate irrelevant noise, thereby improving the accuracy of predictions. The implementation of this network involves the following steps:

Step 1: The input consists of the flattened vector of all past inputs $[\text{I}]_t = \left[\xi_t^{(1)^\tau}, \ldots, \xi_t^{(m_x)^\tau}\right]^\tau$, an external environment vector c_s. The Gated Recurrent Network (GRN) is used for feature selection, and the softmax function is utilized to generate weights V_{xt}, as defined in Eq.(8).

$$\text{V}_{xt} = \text{Softmax}\left(\text{GRN}_{\text{V}_x}\left([\text{I}]_t, c_s\right)\right) \tag{8}$$

Step 2: The GRN is used for nonlinear processing to generate the variable $\tilde{\xi}_t^{(j)}$, as shown in Eq.(9).

$$\tilde{\xi}_t^{(j)} = \text{GRN}_{\tilde{\xi}_{(j)}}\left(\xi_t^{(j)}\right) \tag{9}$$

Step 3: The weights obtained in Step 1 are multiplied by the input data at each moment in Step 2 and then summed. This process yields the selected data, where the weights serve to amplify certain inputs and attenuate noise, as defined in Eq.(10).

$$\tilde{\xi}_t = \sum_{j=1}^{m_x} v_{xt}^{(j)} \tilde{\xi}_t^{(j)} \tag{10}$$

where $v_{xt}^{(j)}$ denotes the j th element of the vector V_{xt}.

2.2.3. Interpretable multi-head attention

In machine learning, self-attention is an important mechanism to integrate the absolute or relative position information in the model structure. In this paper, we employ an interpretable multi-head attention component to capture the interdependence and significance of different distinctive subspace of representations. First, the self-attention mechanism is implemented through the use of query ($Q \in \mathbb{R}^{N \times d_{\text{atm}}}$), key ($K \in \mathbb{R}^{N \times d_{\text{att}}}$), and value ($V \in \mathbb{R}^{N \times d_{\text{madel}}}$) mechanisms that scale the values, with their relationship being as Eq.(11), where $A()$ is a normalization function. The scaled dot-product attention weights $A(Q, K)$ is a common choice as defined in Eq.(12).

$$\text{Attention}(Q, K, V) = A(Q, K)V \tag{11}$$

$$A(Q, K) = \text{Softmax}\left(QK^T / \sqrt{d_{attn}}\right) \tag{12}$$

The multi-head attention technique is employed to capture extended dependencies, with each head representing a

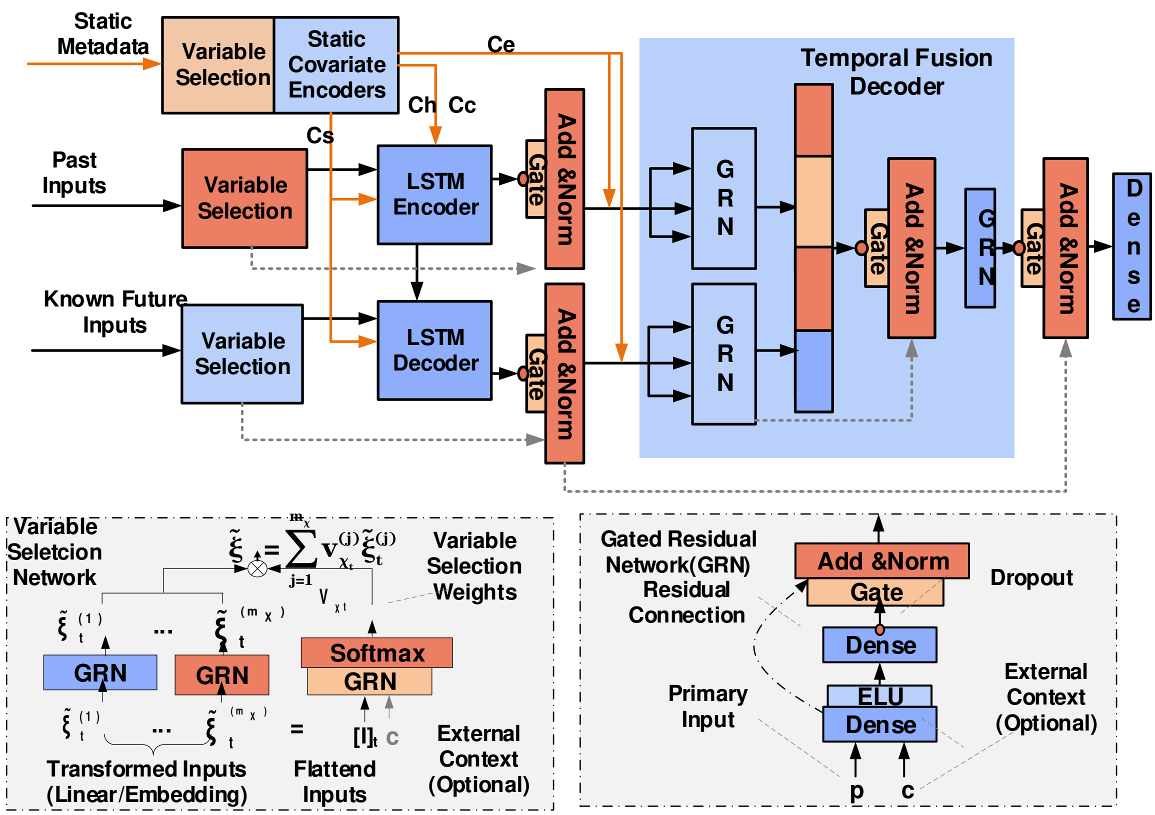

Fig.3. The model architecture of TFT.

distinct subspace of representations, as shown in Eq.(13) and (14).

$$\text{MultiHead}(Q, K, V) = \left[H_1, \dots, H_{m_H}\right] W_H \tag{13}$$

$$H_h = \text{Attention}\left(QW_Q^{(h)}, KW_K^{(h)}, VW_V^{(h)}\right) \tag{14}$$

where $W_Q^{(h)} \in \mathbb{R}^{d_{\text{model}} \times d_{\text{attn}}}$, $W_K^{(h)} \in \mathbb{R}^{d_{\text{model}} \times d_{\text{attr}}}$, and $W_V^{(h)} \in \mathbb{R}^{d_{\text{madel}} \times d_V}$ denote head-specific weights for Q, K, V, respectively. $W_H^{(h)} \in \mathbb{R}^{(m_H \bullet d_v) \times d_{\text{model}}}$ linearly combines outputs concatenated from all heads H_h.

To enhance interpretability, the TFT introduces additional weights that indicate the significance of various features for each attention head, as defined in Eq.(15) and Eq.(16), where $\boldsymbol{W}_V \in \mathbb{R}^{d_{\text{model}} \times d_V}$ are value weights shared across all heads, and $\boldsymbol{W}_H \in \mathbb{R}^{d_{\text{attn}} \times d_{\text{model}}}$ is used for final linear mapping.

$$\text{InterpretableMultiHead}(Q, K, V) = \tilde{H} W_H \tag{15}$$

$$\begin{aligned} \tilde{\boldsymbol{H}} &= \tilde{A}(\boldsymbol{Q}, \boldsymbol{K}) \boldsymbol{V} \boldsymbol{W}_V, \\ &= \left\{ \frac{1}{m_H} \sum_{h=1}^{m_H} A\left(\boldsymbol{Q}\boldsymbol{W}_Q^{(h)}, \boldsymbol{K}\boldsymbol{W}_K^{(h)}\right) \right\} \boldsymbol{V}\boldsymbol{W}_V \\ &= \frac{1}{m_H} \sum_{h=1}^{m_H} \text{Attention}\left(\boldsymbol{Q}\boldsymbol{W}_Q^{(h)}, \boldsymbol{K}\boldsymbol{W}_K^{(h)}, \boldsymbol{V}\boldsymbol{W}_V\right) \end{aligned} \tag{16}$$

2.2.4. *Temporal fusion decoder*

(1) Sequence-to-sequence layer

The sequence-to-sequence layer includes encoder and decoder layers which utilize LSTM units. The encoder uses feature vectors $\{\xi_{t-k}, \dots, \xi_t\}$ that represent past time instances, while fed $\{\xi_{t+1}, \dots, \xi_t + \tau_{\max}\}$ into decoder. These vectors are combined through gated jump connections to create a coherent set of temporal features $\phi(t, n) \in \{\phi(t, -k), \dots, \phi\left(t, \tau_{\max}\right)\}$. This process is formulated in Eq.(17).

$$\tilde{\phi}(t, n) = \text{LayerNorm}\left(\tilde{\xi}_{t+n} + \text{GLU}_{\tilde{\phi}}(\phi(t, n))\right) \tag{17}$$

The position index, denoted as $n \in \left[-k, \tau_{\max}\right]$, and it indicates the specific position within the sequence. Additionally, the context vectors c_h and c_c obtained from the static covariant encoder, are used to initialize the first LSTM cell and hidden state.

(2) Static enrichment layer

The static enrichment layer integrates the c_e values derived from the static features using GRN and the feature vectors $\tilde{\boldsymbol{\phi}}(t,n)$ abstracted through the LSTM layer. Subsequently, GRN is employed once again to enhance the features, as shown in Eq.(18).

$$\boldsymbol{\theta}(t,n) = \mathrm{GRN}_{\theta}\left(\tilde{\boldsymbol{\phi}}(t,n), \boldsymbol{c}_e\right) \tag{18}$$

(3) Temporal self-attention layer

The temporal self-attention layer combines improved temporal features and utilizes interpretable multi-head attention to calculate $\boldsymbol{B}(t)$, as defined in Eq.(19), where $\Theta(t) = [\theta(t,-k), \ldots, \theta(t,\tau)]^T$, and $\boldsymbol{B}(t) = \left[\boldsymbol{\beta}(t,-k), \ldots, \boldsymbol{\beta}\left(t, \tau_{\max}\right)\right]$. In contrast, the TFT includes a gating layer that aids in the training process, as shown in Eq.(20).

$$\boldsymbol{B}(t) = \mathrm{InterpretableMultiHead}(\boldsymbol{\Theta}(t), \boldsymbol{\Theta}(t), \boldsymbol{\Theta}(t)) \tag{19}$$

$$\delta(t,n) = \mathrm{LayerNorm}\left(\theta(t,n) + \mathrm{GLU}_{\delta}(\beta(t,n))\right) \tag{20}$$

(4) Position-wise feed-forward Layer

The position-wise feed-forward layer predicts the result by applying attention to the previous items for each prediction moment using a GRN with shared weights. This ensures that the output of each step is not diluted by a biased GRN, especially in multi-step prediction scenarios. The implementation is given in Eqs.(21) and (22).

$$\psi(t,n) = \mathrm{GRN}\,\psi(\delta(t,n)) \tag{21}$$

$$\tilde{\psi}(t,n) = \mathrm{LayerNorm}\left(\tilde{\phi}(t,n) + \mathrm{GLU}_{\tilde{\psi}}(\psi(t,n))\right) \tag{22}$$

2.2.5. Quantile outputs and loss functions

The TFT employs quantile prediction, where the output layer is linearly transformed using the dense layer to generate percentiles, such as the 10th, 50th, and 90th. During training, the objective is to minimize both the quantile loss and the sum of the outputs. The process is formulated in Eqs.(23) and (24).

$$\mathscr{L}(\Omega, W) = \sum_{y_t \in \Omega} \sum_{q \in \epsilon} \sum_{\tau=1}^{\tau_{\max}} \frac{QL(y_t, \hat{y}(q, t-\tau, \tau), q)}{M_{\tau_{\max}}} \tag{23}$$

$$\mathrm{QL}(y, \hat{y}, q) = \max(q(y-\hat{y}), (1-q)(\hat{y}-y)) \tag{24}$$

where Ω is the domain of training data containing M samples, Q is the set of output quantiles, W is the weights of TFT.

2.3. Algorithm summary

The prediction of cooling load is not only related to date characteristics such as hour, day type, month and weekend but also to meteorological station data like outdoor temperature, relative humidity, solar radiation intensity and passenger number. To incorporate clustering results into the TFT model, we assign each cluster category, C_i, corresponding cluster center, u_i and statistical features of cooling load at the same historical time of the same category, and use them to construct the model input F. In this study, we categorize the inputs for the TFT model into three types as Fig.3: static metadata, past inputs, and known future inputs, denoted as F_S, F_P, and F_K, respectively. F_S contains C_i, the encoder length, which is 168 as a time sliding window, F_P includes cooling load and passenger number, and F_K encompasses the date characteristics, meteorological data, clustering tatistical features and u_i. The interpretable outputs of the model in Section 4.3. also provide a comprehensive view of the model inputs. The algorithm flow is presented in the Algorithm 1.

3. A Case Study

3.1. Site description

The historical cooling load data used in this study was obtained from Energy Station No.1 at Chongqing Jiangbei International Airport. The station utilizes a TES system, consisting of two cold energy storage tanks and six refrigeration units with a rated power of 8500 kWth (kW). The system has a capacity of 43447 RT.H and operates as a closed-loop, two-pipe control system with a three-stage pump variable-flow system, as depicted in Fig. 4, where the upper part is the physical diagram of the pipeline, and the arrow below represents the main flow direction. The energy station provides cooling to T3A Terminal with a building height of 48 meters and a total floor area of approximately 537,000 square meters.

3.2. Dataset description

The cooling data was collected from the TES system and is used as input for the empirical formula Eq. (25) to calculate the cooling load, shown as Fig. 5.

$$Q(i) = \rho c q(i) \Delta T(i) \tag{25}$$

where $Q(i)$ represents the value of the cold load at time period i, measured in kW, ρ denotes the density of water in the cooling system pipeline, measured in kg/m^3, c represents the specific heat capacity of water, measured in J/(kg·°C), $q(i)$ indicates the flow rate of chilled water during time period i, measured in m^3/h, $\Delta T(i)$ represents the temperature difference between the return water and supply water, measured in °C. For the terminal building, meteorological data such as temperature, humidity, solar radiation, wind speed, and other relevant information for the same period of time outside are provided by the Open-Meteo. com Weather API [47]. Passenger flow data is counted by the airport operations system. Overall, the data was recorded at 1-hour interval

Algorithm 1 Cooling load forecasting by k-means-TFT

Data preprocessing
Construct data set $\boldsymbol{D}$= {cooling load data, meteorological data, airport traffic data} as Section 3.2.
Detect outliers and remove them as Section 3.3.2.
Fill missing data as Section 3.3.1.
Obtain W_x from Pearson correlation coefficient as Section 3.3.5.
Split the $\boldsymbol{D}$ into $\{\boldsymbol{D}_{train}, \boldsymbol{D}_{valid}, \boldsymbol{D}_{test}\}$ as Section 3.2, and normalize each set separately as Section 3.3.3.
Training process
Obtain cluster center u_i and cluster category C_i by applying improved k-means for $\boldsymbol{D}_{train}$ as Section 2.1, $i = 0, , ..., K$, cluster target as Eq. (2).
Then construct model inputs F=$\{F_S, F_P, F_K\}$.
for all $j = 1, 2, \dots, \mathrm{len}(\boldsymbol{D}_{\mathrm{train}})$ **do**
  if $\boldsymbol{D}_{\mathrm{train}_j}.C_i == \boldsymbol{D}_{\mathrm{train}_{j\_pre}}.C_i$ **then**
    compute statistical_features$_j$.
    $F_j = \{F_{S_j}, F_{P_j}, F_{K_j}\}$
  end if
end for
Train TFT model using F, $\boldsymbol{D}_{valid}$ for monitoring training process.
Testing process
Apply improved k-means for $\boldsymbol{D}_{test}$.
for all $j = 1, 2, \dots, \mathrm{len}(\boldsymbol{D}_{\mathrm{test}})$ **do**
  for all $i = 1, 2, \cdots, K$ **do**
    Solve all distance $d_{ji} = \mathrm{DTW}(\boldsymbol{D}_{\mathrm{test}_j} - u_i) \cdot W_x$ and let $\boldsymbol{D}_{\mathrm{test}_j}$ with the minimum d_{ji} belong to C_i.
    Choose C_i as the current cluster category.
  end for
end for
Then construct model inputs like Training process.
for all $i = 1, 2, \dots, \mathrm{len}(\boldsymbol{D}_{\mathrm{test}})/24$ **do**
  $\hat{Q}_i$=TFT (F).
end for
return Evaluation metrics using $(Q_i, \hat{Q}_i)$ as Section 3.4.

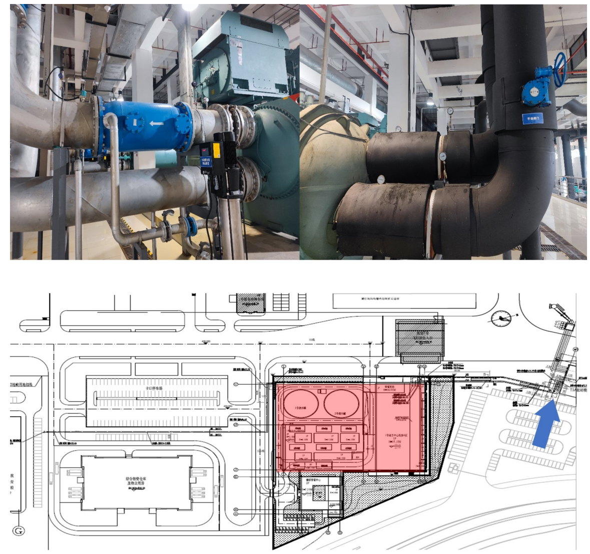

Fig.4. Chilled water pipe (upper) and Layout Plan (below) of Chongqing Jiangbei T3A District Cooling System.

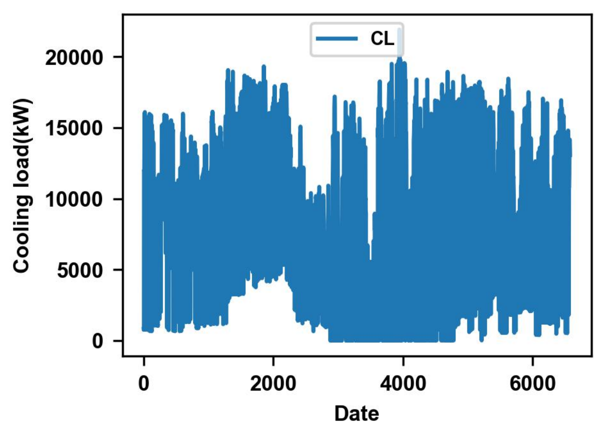

Fig.5. Historical cooling load data in T3A terminal.

and covered two cooling seasons: June-September 2020 and May-September 2021. This resulted in a total of 6576 entries over 274 days as $\boldsymbol{D}$. Out of these, 200 days are used as the training set $\boldsymbol{D}_{\mathrm{train}}$, 60 days for validtion $\boldsymbol{D}_{\mathrm{valid}}$ and 14 days are used as the test set $\boldsymbol{D}_{\mathrm{test}}$. The proposed algorithms are implemented using Python 3.8, and the experiments are performed on a computer with 16GB of RAM and an Intel (R) Core (TM) i5-7400 CPU @ 3. 00GHz processor. The deep learning model was implemented on the Google Colab Pro platform using a high-performance GPU (NVIDIA Tesla P100 PCIe 16G) to accelerate the computation [48]. The PyTorch framework 2.0.1 was utilized.

3.3. Data prepossessing

3.3.1. Missing data

In the process of measurement, individual data points may be absent due to various factors such as equipment failures or electromagnetic interference. In cases where there are data points available both before and after the missing values, a method known as linear interpolation can be used to estimate and replace the missing values [49], as represented by Eq. (26).

$$Q = \frac{Q_{i-1} + Q_{i+1}}{2} \tag{26}$$

In cases where a substantial amount of data is missing, the missing values can be replaced using the method of averaging the values of neighboring points [50]. This method relies on two variables: k, which represents the number of days of the same type, and m, which represents the selected number of days of the same type. These variables are defined in Eq. (27).

$$Q = \frac{Q_1^k + Q_2^k + ... + Q_n^k}{m} \tag{27}$$

3.3.2. Outlier detection

In order to reduce the impact of abnormal noise on the accuracy of future predictions, it is necessary to smooth out data points that deviate significantly from the actual data. Quartiles, which divide the data into four equal parts based on their values, can be used to evaluate statistical dispersion and variability. The quartile spacing, also known as the interquartile range (IQR), is defined as the difference between the third quartile and the first quartile, serves as a measure of data variability. In this context, a global outlier is typically identified as an observation that satisfies Eq. (28), as specified below.

$$\text{outlier} > Q3 + 1.5 \times \text{IQR}, \text{outlier} < Q1 - 1.5 \times \text{IQR} \quad (28)$$

3.3.3. Data normalization

In this study, the input features of the model exhibit variations in unit and range of values. Additionally, the cold load, which ranges from a few kilowatts to tens of thousands of kilowatts, requires separate normalization for the training and test sets. To address this, min-max normalization is employed to normalize the input features, shown in Eq. (29).

$$Q_{\text{norm}} = \frac{Q - Q_{\text{min}}}{Q_{\text{max}} - Q_{\text{min}}} \quad (29)$$

where Q_{norm} represents normalized data, Q represents data before normalization, Q_{min} represents the minimum value of the data, and Q_{max} represents the maximum value of the data.

3.3.4. Passenger number

The correlation between airport passenger flow and cold load is a significant factor to consider [51, 52]. We examine the passenger flow data of T3A, as illustrated in the accompanying Fig. 6. The data demonstrates a consistent peak in the morning hours, specifically between 6-9am, followed by a relatively stable period from 9am to 11pm. During the late night rest period, there is a significant decrease in passenger traffic. This observed pattern aligns with both the empirical data and the practical functioning of the terminal building. As a result, it effectively characterizes the operational status of the building, mirroring the daily fluctuations in load at the terminal building.

3.3.5. Pearson correlation analysis

The cooling load of an airport terminal building exhibits nonlinear dynamics and is influenced by various factors, such as meteorological changes and indoor personnel movement [53, 52, 54]. Pearson correlation analysis helps identify the interrelationships between variables and allows for the exclusion of data that is not significantly correlated, thereby reducing the dimensionality of the input data. In this study, we focus on the outdoor temperature (AT), humidity (AH), solar radiation intensity (SI), wind speed (WS), wind direction (WD), and passenger flow data of the T3A terminal building (ON). A correlation heat map is constructed to visualize the correlations between each variable and the cooling

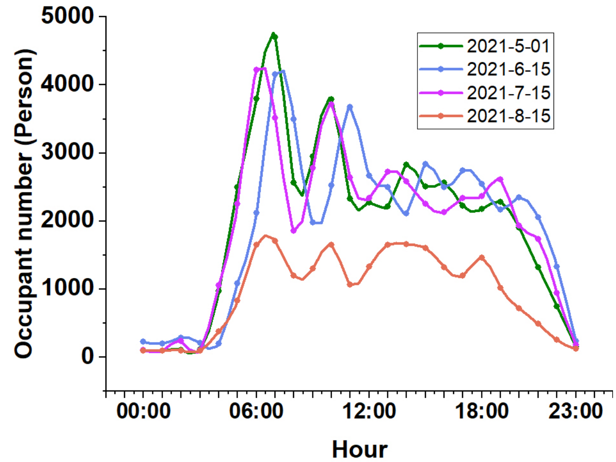

Fig.6. Number of indoor occupants in the T3A terminal over a 24-hour period.

load (CL) over a 24-hour period (Fig. 7). It is worth noting that wind speed and wind direction, which exhibit a low correlation, are excluded from the analysis. Furthermore, correlation curves are plotted to examine the relationship between physical quantities and cooling loads at various delay times ranging from 0 to 24 hours (Fig. 8). These curves reveal that certain delay features also impact the cooling load. The moments that exhibit the highest correlation are highlighted with blue boxes.

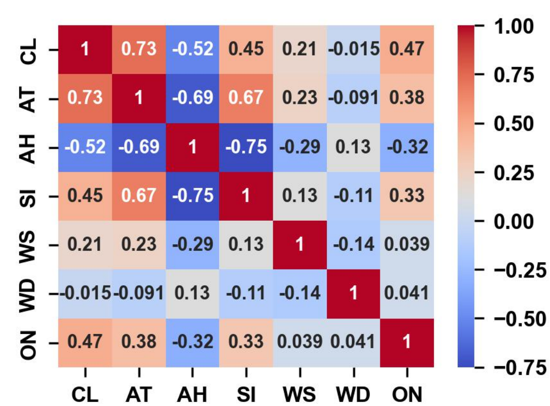

Fig.7. Correlation heatmap between physical quantity and cooling load.

3.4. Evaluation metrics

Various metrics were employed to assess the performance of model, including Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Cross-Validated Root Mean Square Error (CV-RMSE), and Coefficient of Determination (R2) [55]. These metrics are commonly used to measure the difference between observed and true values. A smaller value for MAE, CV-RMSE, and MAPE indicates a higher level of accuracy in the performance of the predictive model. R2, on the other hand, determines the extent to which the model aligns with the data. The R2 value ranges from 0 to

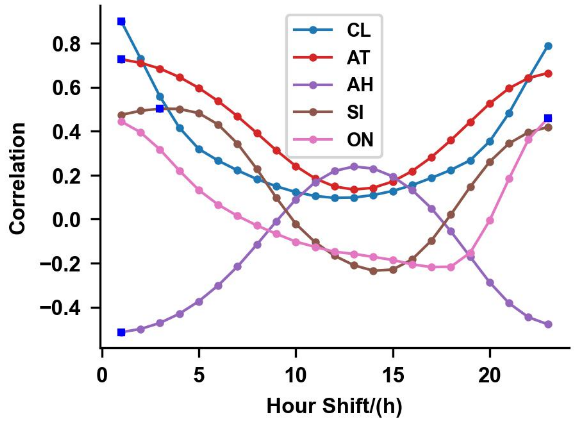

Fig.8. Correlation between physical quantity and cooling load at different historical time.

1, with a higher value indicating a stronger fit. These metrics are defined as follows:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} |\hat{Q}_i - Q_i| \tag{30}$$

$$\mathrm{MAPE} = \frac{1}{n}\sum_{i=1}^{n} \left|\frac{\hat{Q}_i - Q_i}{Q_i}\right| \tag{31}$$

$$\mathrm{CV-RMSE} = \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{Q}_i - Q_i)^2}}{\frac{1}{n}\sum_{i=1}^{n} Q_i} \tag{32}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(\hat{Q}_i - Q_i)^2}{\sum_{i=1}^{n}(Q_i - \bar{Q})^2} \tag{33}$$

In the above equations (30)-(33), n represents the quantity of predicted moment points, Q_i denotes the actual cold load, $\hat{Q}_i$ represents the predicted cold load, and $\bar{Q}$ signifies the average value of the actual load.

3.5. XGBoost and LSTM

XGBoost is an enhanced algorithm derived from gradient boosting trees. It utilizes decision tree sub-models and guides the training process of these sub-models through negative gradients to achieve highly accurate integrated models. It possesses a robust capability for nonlinear fitting and rank high in energy forecasting competitions such as ASHRAE Great Energy Predictor III. The XGBoost model is implemented using sklearn API [56]. And LSTM, a specialized form of recurrent neural network (RNN) [57], introduces three gates to regulate the flow of information and address the issue of vanishing or exploding gradients that arise when the RNN time step is too large. The input gate determines the inclusion of new information, the forget gate determines whether the previous information is retained, and the output gate determines the output information. LSTM is widely used in time series modeling tasks. In order to incorporate weather forecast data, the LSTM model is designed with multiple inputs [35]. The Optuna API is adopted to search for hyperparameters [58], which employs the Bayesian optimization to explore various parameter combinations, demonstrating notable efficiency. All the model parameters optimized by the Optuna API are listed in Table 1.

Table 1
Hyper-parameters of four models Using Optuna API

Hyperparameters	Search ranges	Selected
LSTM		
Learning rate	[0.001, 0.005, 0.01]	0.005
Batch size	[16, 32, 64]	16
Hidden size of LSTM	[32, 64, 128, 160]	64
Num layers	[1, 2, 3, 5]	2
Dropout	[0.2, 0.3, 0.5]	0.2
XGBoost		
Max depth	[1, 10]	5
N estimators	[50, 2000]	200
Min child weight	[1, 10]	3
Colsample bytree	[0.1, 1]	0.733
Subsample	[0.1, 1]	0.355
Eta	[0.01, 0.3]	0.07
Transformer		
d_ff	[512,1024,2048]	2048
d_model	[128,512,1024]	512
n_heads	[4,6,8]	8
Learning rate	[0.001,0.005,0.1]	0.001
Batch size	[16,32,64]	16
Dropout	[0.1,0.2,0.3]	0.1
TFT		
Gradient clip val	[0.01, 0.1, 1.0]	0.1
Hidden size of LSTM	[64, 128, 160]	128
Hidde continuous size	[64, 128, 160]	128
Attention head size	[1, 4, 6]	4
Learning rate	[0.001, 0.005, 0.1]	0.005
Batch size	[16, 32, 64]	16
Dropout	[0.1, 0.2, 0.3]	0.2

4. Results and discussions

4.1. Persistent model prediction

Persistent daily and weekly algorithms, also referred to as "naïve" forecasts, have demonstrated deterministic properties in various competitions [25, 35, 59]. To assess the effectiveness of these models and evaluate the performance for 24-hour-ahead forecasts, four baseline models are proposed. These models are based on previous statistical data. The first model, namely the last day (LD), directly utilizes the load data from the previous day as a 24-hour forecast, assuming the presence of a daily pattern in the load data. Similarly, last week (LW) uses the load data from the same day and time of the previous week as the predicted value, assuming a weekly periodicity in the load data. Hour mean (HM)

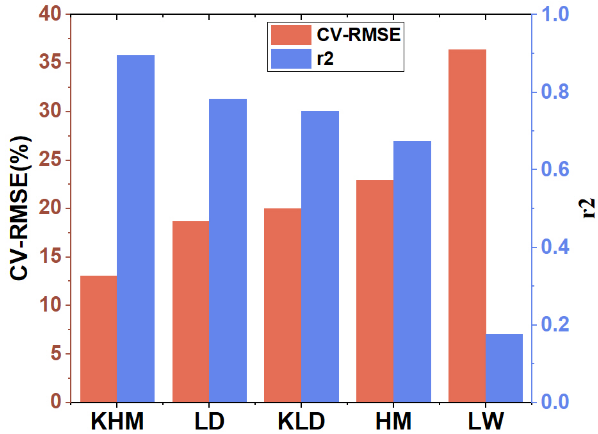

Fig.9. Forecasting performance of naive persistent model with or without k-means.

calculates the average value of the corresponding hour, while our k-means last day (KLD) applies our clustering algorithm to identify the closest day in the same type. Our k-means hour mean (KHM) uses its load pattern to predict the mean value for the corresponding moment in the same cluster, shown in Eq. (34). These models rely on expert knowledge and past similar circumstances to make weighted average decisions, which have been shown to be reasonably accurate. The results, as shown in Fig.9, indicate that the KHM model achieves the highest performance, with an R2 score of 0.885 and a CV-RMSE of 13.590%. Consequently, this model will serve as a baseline for future comparisons. The heuristic baseline model is characterized by its simplicity, data efficiency, ease of implementation, and maintenance, making it suitable for small scale projects with limited budgets.

$$CL_{n+1,t} = \frac{CL_{1,t} + CL_{2,t} + \ldots + CL_{n,t}}{n} \tag{34}$$

4.2. Clustering results of load modes

The data is subjected to clustering using the improved k-means method, as described in the current study. The outcomes of the data clustering process are graphically represented in Fig.10, revealing the existence of four distinct load patterns observed in the T3A terminal during the cooling season. It is noteworthy that the peak stabilized load values differ significantly among these load patterns. Fig.11 illustrates the original heatmap before clustering, while Fig.12 displays the heatmap organized by category after clustering, similar data is effectively clustered together. Furthermore, the clustering algorithm will be used on the Hour mean model to evaluate the accuracy of the subsequent 24-hour cooling load predictions and assess the effectiveness of the k-means approach. The findings of this investigation demonstrate that the clustering algorithm, specifically when employing weight-based DTW distances, yields superior outcomes, as presented in Table 2.

Table 2
Hour mean model performance

Previous Hour mean	CV-RMSE (%)	R2
Origin	22.906	0.673
k-means-navie	15.843	0.844
Weighted-DTW-k-means	**13.590**	**0.885**

4.3. Model based prediction

To evaluate the performance of the enhanced k-means method when combined with the model, as well as the model's performance using only raw data without making predictions, the results in test set are presented, we selected 4 days for presentation. Fig.13 demonstrates that TFT and XGBoost achieve superior results without using k-means, outperforming the baseline model. On the other hand, Fig.14 illustrates that our proposed model yields the best prediction among all the results. Furthermore, the introduction of our k-means improves the performance of all the models compared to the initial model without clustering. Detailed results of the comparisons can be found in Table 5. To ensure statistically significant results, the data in the table represent the means of 10 experiments. The TFT model produces easily understandable results by incorporating clustering labels as static enhancement vectors. These vectors enhance the temporal self-attention layer's capability to capture overall loading patterns, as illustrated in Fig.15. Moreover, the model incorporates exogenous variables into the clustering centers namely ClusterCenterHourFeature and like-mean features namely MA_Predictlon, with the variable selection weights offering valuable insights. The clustering features are found to play a significant role, as illustrated in Fig.16 and Fig.17. Furthermore, Fig.18 displays the relationship between the time step and attention, revealing a stronger attention near the predicted date, this also demonstrates the model's capability to recall long-term inputs. Further, an early stopping strategy was adopted in the training of each model and was repeated for 10 times. The results including the mean and standard deviation are shown in Table 3. While the time of conducting the clustering algorithm is 3.977 seconds. It is shown that the training time of the proposed algorithm after employing k-means was shorter, which implies that the adoption of the improved k-means helped the acceleration of the training convergence. The inference time of the model is also listed in the table 4. It is revealed that while XGBoost demonstrates strong competitiveness, k-means TFT exhibits superior accuracy within the context of deep modeling, and its processing time is also deemed satisfactory.

4.4. Discussions

In order to address the contradiction between the changing cold load pattern and the accuracy of prediction caused by the diverse energy use behavior in large spaces, this study collects cold load data from the main pipeline of an airport refrigeration station, as well as relevant meteorological data. Previous clustering algorithms have not simultaneously taken into account the time characteristics of the clustering

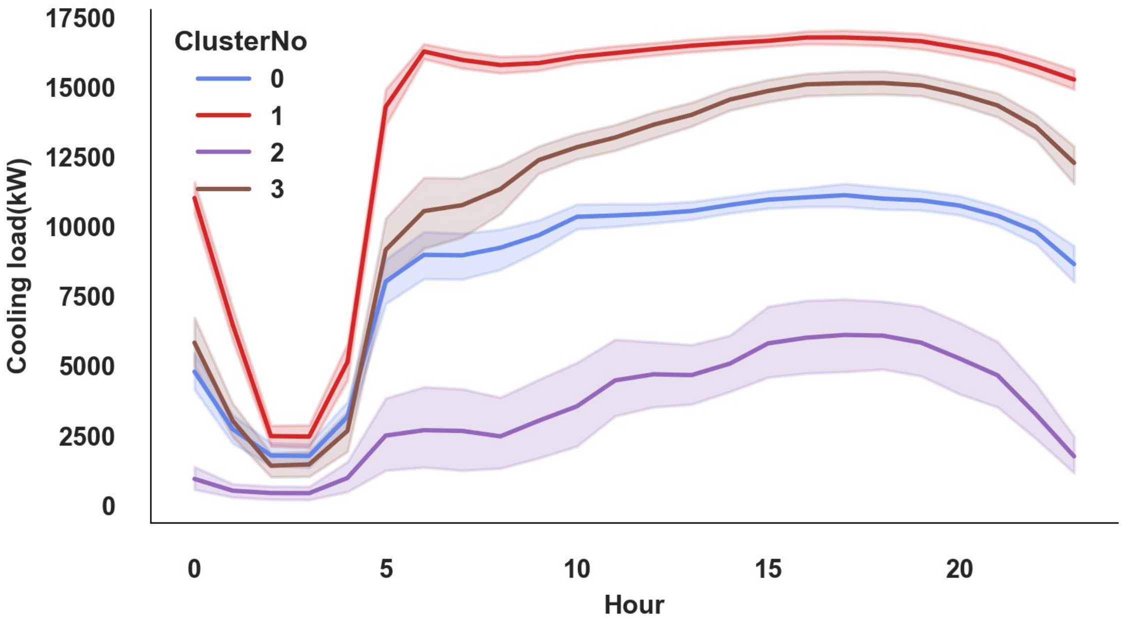

Fig.10. Typical load patterns by weight-DTW-k-means.

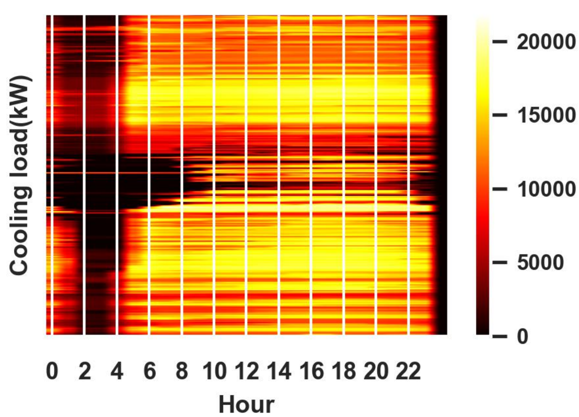

Fig.11. Heat map for load profiles of the original data.

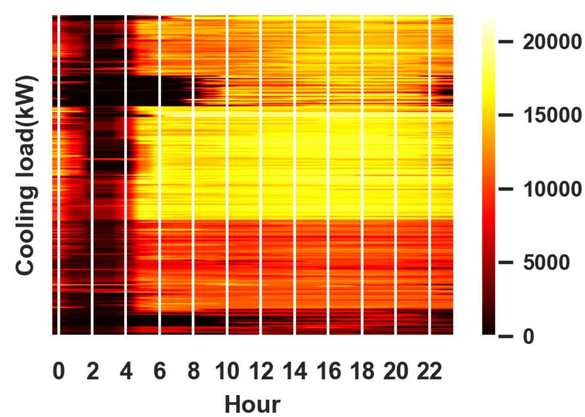

Fig.12. Heat maps for cluster by weighted-DTW-k-means.

Table 3
Training time of different models

Algorithm	Time (s)	Algorithm-k	Time (s)
XGBoost	4.510 ± 0.108	**XGBoost-k**	3.264 ± 0.092
LSTM	48.777 ± 7.352	LSTM-k	47.533 ± 4.579
Transformer	217.829 ± 3.729	Transformer-k	217.230 ± 4.235
TFT	536.331 ± 32.599	TFT-k	486.943 ± 8.860

Table 4
Inference time of different models

Algorithm	Time (s)	Algorithm-k	Time (s)
XGBoost	0.016 ± 0.001	**XGBoost-k**	0.019 ± 0.004
LSTM	0.027 ± 0.003	LSTM-k	0.031 ± 0.005
Transformer	7.844 ± 1.98	Transformer-k	7.588 ± 1.055
TFT	8.292 ± 1.394	TFT-k	8.875 ± 0.500

object and the influence of feature weight. Therefore, this paper proposes the use of Dynamic Time Warping distance as the clustering metric and Pearson's correlation coefficient as a weighting factor for data clustering. Furthermore, the interpretability of the TFT is utilized to showcase the efficacy of statistical features derived from k-means clustering, leading to enhanced prediction outcomes. This has significant implications for predicting various modes of refrigeration load and for regulating and controlling chillers. Furthermore, for projects with limited budgets, it is suggested that good

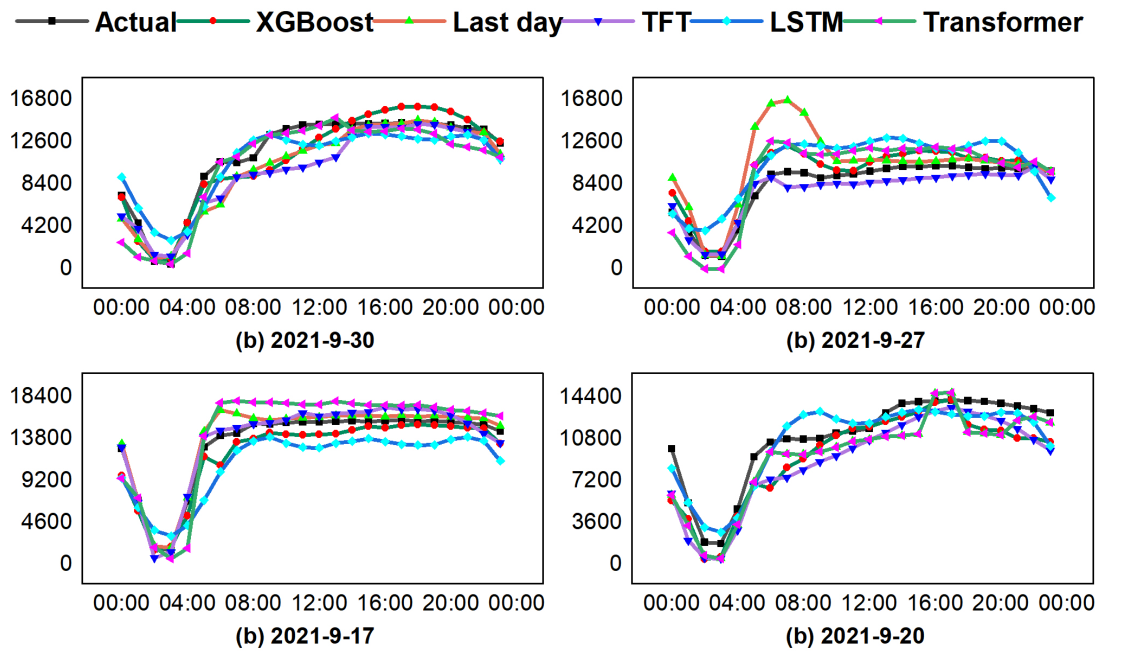

Fig.13. KHM, XGBoost, LSTM ,Transformer, TFT model for CL(kW) prediction without k-means.

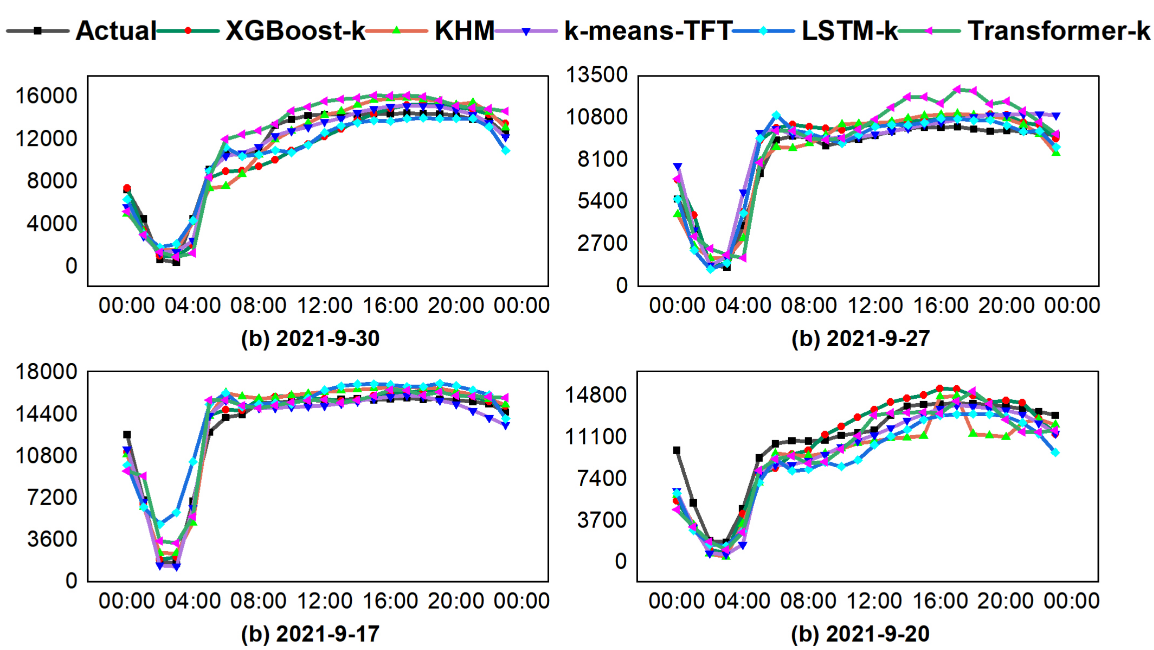

Fig.14. KHM, XGBoost, LSTM ,Transformer,TFT model for CL(kW) prediction with improved k-means.

results can still be achieved by using a simplified model after clustering. However, it is important to note that when new energy use patterns and behaviors emerge, the clustering algorithm will need to be recalculated. This process is considered as an offline approach. Therefore, future research should focus on developing methods for differentiating online patterns and automatically detecting new patterns. For public buildings with advanced building intelligence, dividing the functional partitions to create a more detailed

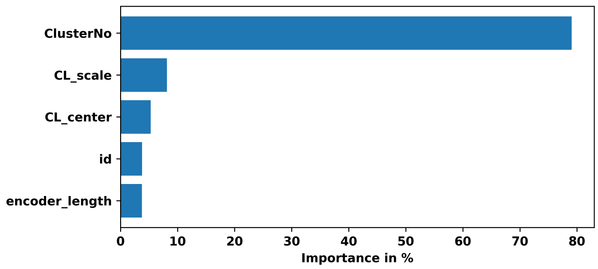

Fig.15. Static covariates importance.

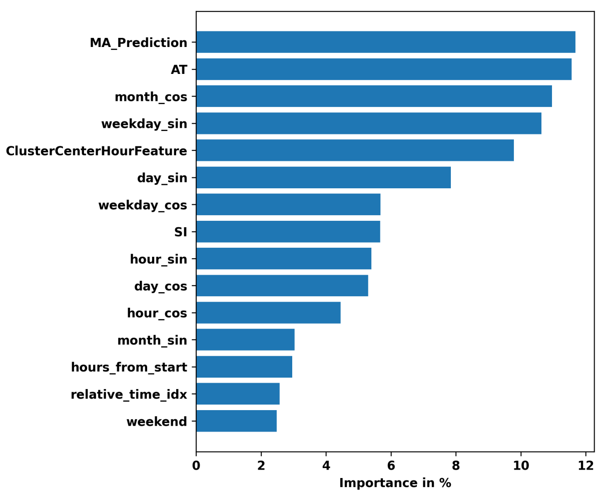

Fig.16. Importance of exogenous variables in the encoder.

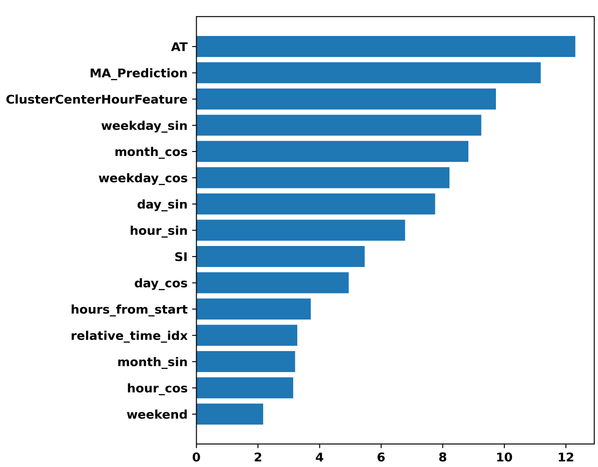

Fig.17. Importance of exogenous variables in the decoder.

partitioning model can significantly improve the model's accuracy. Additionally, in buildings with more consistent energy usage patterns and fewer scattered load patterns, clustering algorithms may increase computational demands without significantly enhancing accuracy. Therefore, it is

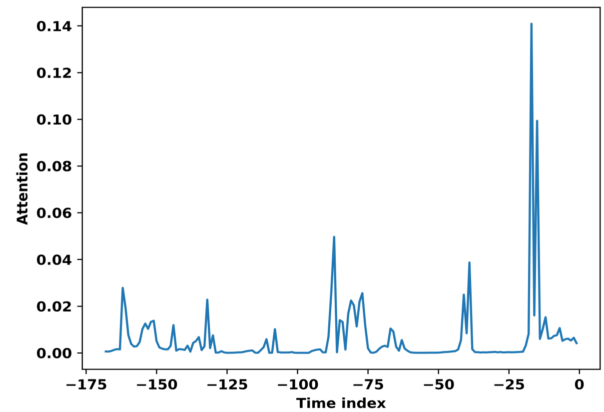

Fig.18. Attention weight patterns. Attention at previous time steps.

necessary to evaluate the data before redesigning the model.

5. Conclusions and future work

In many engineering applications of building energy forecasting, the accuracy of predictions is compromised due to the variability of the patterns of cold loads which makes forecasting cooling loads in large public buildings more challenging. This paper proposes an enhanced version of the k-means algorithm for clustering load data. It introduces the evaluation distance Dynamic Time Warping, which is better suited for temporal sequences. Additionally, it incorporates a weighting factor that considers the influence of feature variables. This allows the assessment of different loading patterns using a single model. To incorporate the clustering results into TFT and capture both long and short-term dependencies, cluster labels are embedded as static variables, and the statistical characteristics of similar data are embedded in TFT, utilizing a multi-head attention mechanism to achieve improved and interpretable results. The proposed approach is compared to the best baseline model, as well as the machine learning model XGBoost and the deep learning model LSTM, all of which incorporate clustering results. However, unlike these models, the proposed approach does not group the results in order to avoid reducing the size of the dataset. Instead, the clustering results are used in the form of feature vectors to assess the effectiveness of the model. Experimental results demonstrate that this method effectively enhances prediction accuracy, supports the scheduling of chiller units, and reduces unnecessary energy waste. Although the experimental results demonstrate the superiority of the method, there are a few issues to be addressed as the future work:

- The influence of passenger behavior on energy consumption is considered, yet the uncertainty regarding passenger numbers on the second day poses a challenge for load forecasting. Treating it as a predictive variable for load forecasting, prior to using it as an

Table 5
Comparison of quality performance indicators for cooling load prediction of different prediction models.

Day-ahead Model	Our k-means	Performance index			
		MAPE	CV-RMSE (%)	MAE (kW)	R^2
Best Baseline (LD)		0.1719	18.712	1442	0.782
LSTM [35]		0.1717	16.251	1353	0.835
XGBoost [56]		0.1682	15.607	1291	0.864
Transformer [60]		0.3380	19.515	1603	0.845
TFT [34]		**0.1567**	**13.915**	**1139**	**0.879**
BestBaseline (KHM)	✓	0.1540	13.590	1020	0.885
LSTM-k-means	✓	0.1468	14.506	1213	0.868
XGBoost-k-means	✓	0.1372	11.642	923	0.915
Transformer-k-means	✓	0.2049	13.981	1148	0.891
k-means-TFT (Ours)	✓	**0.1201**	**10.873**	**917**	**0.922**

input, deserves further research. Further, to assess the impact of such uncertainties on the forecast results of the proposed approach in this paper warrants further investigation [61].

- While the proposed model has exhibited highly desirable performance, to maximize its potential in support building energy management, one of the future research directions would be integrate the model into the control and optimization framework to enhance the operational efficiency and energy saving in the airport air-conditioning system [62].

CRediT authorship contribution statement

Die Yu: Conceptualization, Data curation, Formal analysis, Methodology, Software, Writing - original draft. **Tong Liu:** Conceptualization, Resources, Supervision, Writing review & editing. **Kai Wang:** Conceptualization, Funding acquisition, Project administration, Supervision, Writing review & editing. **Kang Li:** Conceptualization, Supervision, Writing - review & editing. **Mehmet Mercangöz:** Supervision, Writing review & editing. **Jian Zhao:** Data curation, Investigation, Validation. **Yu Lei:** Data curation, Project administration, Validation. **RuoFan Zhao:** Data curation, Investigation.

Acknowledgement

The authors gratefully acknowledge the support of this research by the Chongqing Technological Innovation and Application Development Project (CSTB2022TIAD-KPX01 14).

References

[1] T. S. C. I. O. of the People's Republic of China, Sustainable development of transport in china, 2021. URL: http://www.scio.gov.cn/zfbps/32832/Document/1695297/1695297.htm.

[2] Y.-J. Hu, H. Huang, H. Wang, C. Li, Y. Deng, Exploring cost-effective strategies for emission reduction of public buildings in a life-cycle, Energy and Buildings 285 (2023) 112927. URL: https://www.sciencedirect.com/science/article/pii/S0378778823001573. doi:https://doi.org/10.1016/j.enbuild.2023.112927.

[3] H. Lund, P. A. Østergaard, T. B. Nielsen, S. Werner, J. E. Thorsen, O. Gudmundsson, A. Arabkoohsar, B. V. Mathiesen, Perspectives on fourth and fifth generation district heating, Energy 227 (2021) 120520.

[4] T. Kannan, C. Lork, W. Tushar, C. Yuen, N. C. Wong, S. Tai, L. Lan, K. L. Wood, Energy management strategy for zone cooling load demand reduction in commercial buildings: A data-driven approach, IEEE Transactions on Industry Applications 55 (2019) 7281–7299. doi:10.1109/TIA.2019.2930599.

[5] A. Carratt, G. Kokogiannakis, D. Daly, A critical review of methods for the performance evaluation of passive thermal retrofits in residential buildings, Journal of Cleaner Production 263 (2020) 121408. URL: https://www.sciencedirect.com/science/article/pii/S0959652620314554. doi:https://doi.org/10.1016/j.jclepro.2020.121408.

[6] X. Liu, L. Lin, X. Liu, T. Zhang, X. Rong, L. Yang, D. Xiong, Evaluation of air infiltration in a hub airport terminal: On-site measurement and numerical simulation, Building and Environment 143 (2018) 163–177.

[7] Y. Yao, J. Chen, Global optimization of a central air-conditioning system using decomposition–coordination method, Energy and Buildings 42 (2010) 570–583.

[8] M. Fiorentini, J. Wall, Z. Ma, J. H. Braslavsky, P. Cooper, Hybrid model predictive control of a residential hvac system with on-site thermal energy generation and storage, Applied Energy 187 (2017) 465–479.

[9] T. Liu, S. Chen, S. Liang, D. Du, C. J. Harris, Fast tunable gradient rbf networks for online modeling of nonlinear and nonstationary dynamic processes, Journal of Process Control 93 (2020) 53–65.

[10] N. Lanzetti, Y. Z. Lian, A. Cortinovis, L. Dominguez, M. Mercangöz, C. Jones, Recurrent neural network based mpc for process industries, in: 2019 18th European Control Conference (ECC), IEEE, 2019, pp. 1005–1010.

[11] M. Mercangöz, F. J. Doyle III, Distributed model predictive control of an experimental four-tank system, Journal of process control 17 (2007) 297–308.

[12] Z. Zhang, Y. Zhou, X. Xin, J. Qian, Y. Liu, A day-ahead operation regulation method for solar water heating based on model predictive control, Energy and Buildings (2023) 113715.

[13] T. Liu, Z. Tian, S. Chen, K. Wang, C. J. Harris, Deep cascade gradient rbf networks with output-relevant feature extraction and adaptation for nonlinear and nonstationary processes, IEEE Transactions on Cybernetics (2022).

[14] T. Liu, S. Chen, P. Yang, Y. Zhu, C. J. Harris, Efficient adaptive deep gradient rbf network for multi-output nonlinear and nonstationary industrial processes, Journal of Process Control 126 (2023) 1–11.

[15] Y. Ding, Q. Zhang, T. Yuan, Research on short-term and ultra-short-term cooling load prediction models for office buildings, Energy and Buildings 154 (2017) 254–267.

[16] X. Luo, L. O. Oyedele, A. O. Ajayi, O. O. Akinade, H. A. Owolabi, A. Ahmed, Feature extraction and genetic algorithm enhanced adaptive deep neural network for energy consumption prediction in buildings, Renewable and Sustainable Energy Reviews 131 (2020) 109980.

[17] G. Li, F. Li, T. Ahmad, J. Liu, T. Li, X. Fang, Y. Wu, Performance evaluation of sequence-to-sequence-attention model for short-term multi-step ahead building energy predictions, Energy 259 (2022) 124915.

[18] Y. Sun, F. Haghighat, B. C. Fung, A review of the-state-of-the-art in data-driven approaches for building energy prediction, Energy and Buildings 221 (2020) 110022.

[19] L. Zhang, J. Wen, Y. Li, J. Chen, Y. Ye, Y. Fu, W. Livingood, A review of machine learning in building load prediction, Applied Energy 285 (2021) 116452.

[20] H. Liu, J. Yu, J. Dai, A. Zhao, M. Wang, M. Zhou, Hybrid prediction model for cold load in large public buildings based on mean residual feedback and improved svr, Energy and Buildings 294 (2023) 113229. URL: https://www.sciencedirect.com/science/article/pii/S0378778823004590. doi:https://doi.org/10.1016/j.enbuild.2023.113229.

[21] X. Kang, X. Wang, J. An, D. Yan, A novel approach of day-ahead cooling load prediction and optimal control for ice-based thermal energy storage (tes) system in commercial buildings, Energy and Buildings 275 (2022) 112478.

[22] F. Wu, R. Jing, X.-P. Zhang, F. Wang, Y. Bao, A combined method of improved grey bp neural network and meemd-arima for day-ahead wave energy forecast, IEEE Transactions on Sustainable Energy 12 (2021) 2404–2412. doi:10.1109/TSTE.2021.3096554.

[23] Y. Wang, J. Chen, X. Chen, X. Zeng, Y. Kong, S. Sun, Y. Guo, Y. Liu, Short-term load forecasting for industrial customers based on tcn-lightgbm, IEEE Transactions on Power Systems 36 (2021) 1984–1997. doi:10.1109/TPWRS.2020.3028133.

[24] Z. Dong, J. Liu, B. Liu, K. Li, X. Li, Hourly energy consumption prediction of an office building based on ensemble learning and energy consumption pattern classification, Energy and Buildings 241 (2021) 110929. URL: https://www.sciencedirect.com/science/article/pii/S0378778821002139. doi:https://doi.org/10.1016/j.enbuild.2021.110929.

[25] Z. Wang, T. Hong, M. A. Piette, Building thermal load prediction through shallow machine learning and deep learning, Applied Energy 263 (2020) 114683.

[26] J. Song, L. Zhang, G. Xue, Y. Ma, S. Gao, Q. Jiang, Predicting hourly heating load in a district heating system based on a hybrid cnn-lstm model, Energy and Buildings 243 (2021) 110998.

[27] K. Benidis, S. S. Rangapuram, V. Flunkert, Y. Wang, D. Maddix, C. Turkmen, J. Gasthaus, M. Bohlke-Schneider, D. Salinas, L. Stella, et al., Deep learning for time series forecasting: Tutorial and literature survey, ACM Computing Surveys 55 (2022) 1–36.

[28] A. Li, F. Xiao, C. Zhang, C. Fan, Attention-based interpretable neural network for building cooling load prediction, Applied Energy 299 (2021) 117238. URL: https://www.sciencedirect.com/science/article/pii/S0306261921006590. doi:https://doi.org/10.1016/j.apenergy.2021.117238.

[29] C. Lu, J. Gu, W. Lu, An improved attention-based deep learning approach for robust cooling load prediction: public building cases under diverse occupancy schedules, Sustainable Cities and Society (2023) 104679.

[30] K. Bandara, C. Bergmeir, H. Hewamalage, Lstm-msnet: Leveraging forecasts on sets of related time series with multiple seasonal patterns, IEEE transactions on neural networks and learning systems 32 (2020) 1586–1599.

[31] A. Li, F. Xiao, C. Zhang, C. Fan, Attention-based interpretable neural network for building cooling load prediction, Applied Energy 299 (2021) 117238.

[32] X. Liang, S. Chen, X. Zhu, X. Jin, Z. Du, Domain knowledge decomposition of building energy consumption and a hybrid data-driven model for 24-h ahead predictions, Applied Energy 344 (2023) 121244.

[33] C. Zhang, J. Li, Y. Zhao, T. Li, Q. Chen, X. Zhang, A hybrid deep learning-based method for short-term building energy load prediction combined with an interpretation process, Energy and Buildings 225 (2020) 110301. URL: https://www.sciencedirect.com/science/article/pii/S0378778820309324. doi:https://doi.org/10.1016/j.enbuild.2020.110301.

[34] B. Lim, S. Ö. Arık, N. Loeff, T. Pfister, Temporal fusion transformers for interpretable multi-horizon time series forecasting, International Journal of Forecasting 37 (2021) 1748–1764.

[35] M. Hu, B. Stephen, J. Browell, S. Haben, D. C. Wallom, Impacts of building load dispersion level on its load forecasting accuracy: Data or algorithms? importance of reliability and interpretability in machine learning, Energy and Buildings 285 (2023) 112896.

[36] J. Liu, Q. Zhang, Z. Dong, X. Li, G. Li, Y. Xie, K. Li, Quantitative evaluation of the building energy performance based on short-term energy predictions, Energy 223 (2021) 120065.

[37] K. Bandara, C. Bergmeir, S. Smyl, Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach, Expert systems with applications 140 (2020) 112896.

[38] W. Li, G. Gong, H. Fan, P. Peng, L. Chun, X. Fang, A clustering-based approach for "cross-scale" load prediction on building level in hvac systems, Applied Energy 282 (2021) 116223.

[39] J. Yang, C. Ning, C. Deb, F. Zhang, D. Cheong, S. E. Lee, C. Sekhar, K. W. Tham, k-shape clustering algorithm for building energy usage patterns analysis and forecasting model accuracy improvement, Energy and Buildings 146 (2017) 27–37.

[40] C. Zhang, R. Li, A novel closed-loop clustering algorithm for hierarchical load forecasting, IEEE Transactions on Smart Grid 12 (2021) 432–441. doi:10.1109/TSG.2020.3015000.

[41] Q. Zhang, Z. Tian, Y. Ding, Y. Lu, J. Niu, Development and evaluation of cooling load prediction models for a factory workshop, Journal of Cleaner Production 230 (2019) 622–633.

[42] X. Zhang, Y. Sun, D. ce Gao, W. Zou, J. Fu, X. Ma, Similarity-based grouping method for evaluation and optimization of dataset structure in machine-learning based short-term building cooling load prediction without measurable occupancy information, Applied Energy 327 (2022) 120144. URL: https://www.sciencedirect.com/science/article/pii/S0306261922014015. doi:https://doi.org/10.1016/j.apenergy.2022.120144.

[43] T. Rakthanmanon, B. Campana, A. Mueen, G. Batista, B. Westover, Q. Zhu, J. Zakaria, E. Keogh, Searching and mining trillions of time series subsequences under dynamic time warping, in: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, 2012, pp. 262–270.

[44] Y. Ding, Q. Zhang, T. Yuan, K. Yang, Model input selection for building heating load prediction: A case study for an office building in tianjin, Energy and Buildings 159 (2018) 254–270.

[45] M. Nejati, N. Amjady, A new solar power prediction method based on feature clustering and hybrid-classification-regression forecasting, IEEE Transactions on Sustainable Energy 13 (2022) 1188–1198. doi:10.1109/TSTE.2021.3138592.

[46] C. Wang, Y. Wang, Z. Ding, T. Zheng, J. Hu, K. Zhang, A transformer-based method of multienergy load forecasting in integrated energy system, IEEE Transactions on Smart Grid 13 (2022) 2703–2714. doi:10.1109/TSG.2022.3166600.

[47] P. Zippenfenig, Open-meteo.com weather api [computer software], https://doi.org/10.5281/ZENODO.7970649, 2023. Zenodo.

[48] E. Bisong, E. Bisong, Google colaboratory, Building machine learning and deep learning models on google cloud platform: a comprehensive guide for beginners (2019) 59–64.

[49] D. A. Newman, Missing data: Five practical guidelines., Organizational Research Methods 17 (2014) 372 – 411. URL: https://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=98555312&lang=zh-cn&site=ehost-live.

[50] N. Karmitsa, S. Taheri, A. Bagirov, P. Mäkinen, Missing value imputation via clusterwise linear regression, IEEE Transactions on

Knowledge and Data Engineering 34 (2022) 1889–1901. doi:10.1109/TKDE.2020.3001694.

[51] L. Lin, X. Liu, T. Zhang, X. Liu, X. Rong, Cooling load characteristic and uncertainty analysis of a hub airport terminal, Energy and Buildings 231 (2021) 110619.

[52] X. Liu, L. Li, X. Liu, T. Zhang, Analysis of passenger flow and its influences on hvac systems: An agent based simulation in a chinese hub airport terminal, Building and Environment 154 (2019) 55–67. URL: https://www.sciencedirect.com/science/article/pii/S0360132319301659. doi:https://doi.org/10.1016/j.buildenv.2019.03.011.

[53] D. Fischer, B. Stephen, A. Flunk, N. Kreifels, K. B. Lindberg, B. Wille-Haussmann, E. H. Owens, Modeling the effects of variable tariffs on domestic electric load profiles by use of occupant behavior submodels, IEEE Transactions on Smart Grid 8 (2017) 2685–2693. doi:10.1109/TSG.2016.2544141.

[54] L. Yu, T. Jiang, Y. Zou, Online energy management for a sustainable smart home with an hvac load and random occupancy, IEEE Transactions on Smart Grid 10 (2019) 1646–1659. doi:10.1109/TSG.2017.2775209.

[55] L. Von Krannichfeldt, Y. Wang, G. Hug, Online ensemble learning for load forecasting, IEEE Transactions on Power Systems 36 (2021) 545–548. doi:10.1109/TPWRS.2020.3036230.

[56] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, G. Varoquaux, API design for machine learning software: experiences from the scikit-learn project, in: ECML PKDD Workshop: Languages for Data Mining and Machine Learning, 2013, pp. 108–122.

[57] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Computation 9 (1997) 1735–1780. doi:10.1162/neco.1997.9.8.1735.

[58] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, Optuna: A next-generation hyperparameter optimization framework, in: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, 2019, pp. 2623–2631.

[59] M. Farrokhabadi, J. Browell, Y. Wang, S. Makonin, W. Su, H. Zareipour, Day-ahead electricity demand forecasting competition: Post-covid paradigm, IEEE Open Access Journal of Power and Energy 9 (2022) 185–191.

[60] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, I. Polosukhin, Attention is all you need, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, volume 30, Curran Associates, Inc., 2017.

[61] Y. Zhuang, Z. Zhou, B. Alakent, M. Mercangöz, Semi-supervised variational autoencoders for regression: Application to soft sensors, in: 2023 IEEE 21st International Conference on Industrial Informatics (INDIN), IEEE, 2023, pp. 1–8.

[62] B. S. Korkmaz, T. Liu, M. Mercangöz, Adaptive real-time exploration and optimization of safety-critical industrial systems with ensemble learning, in: 2023 IEEE 21st International Conference on Industrial Informatics (INDIN), IEEE, 2023, pp. 1–8.