Energy & Buildings 296 (2023) 113409

Contents lists available at ScienceDirect

Energy & Buildings

journal homepage: www.elsevier.com/locate/enb

A novel Transformer-based network forecasting method for building cooling loads

Long Li [a,*], Xingyu Su [a], Xianting Bi [b], Yueliang Lu [c], Xuetao Sun [d]

[a] School of Electrical and Electronic Engineering, Harbin University of Science and Technology, Harbin 150080, China
[b] Beijing Institute of Astronautics System Engineering, Beijing 100076, China
[c] Aerospace Times FeiHong Technology Company Limited, Beijing 102199, China
[d] Institute of Smart City, Shanghai University, Shanghai 200444, China

ARTICLE INFO

Keywords:
Building cooling load forecasting
Short term load forecasting
Transformer algorithm
Feature analysis
Attention mechanism

ABSTRACT

For cooling equipment management and scheduling optimization, accurate building cooling load forecasting technology is crucial. Currently, the physics-based forecasting models are too complex to achieve, and existing shallow-machine and deep learning algorithms are difficult to capture and retain sequential information from historical building cooling loads, leading to insufficient prediction accuracy. This paper considered the dependency relationship between time-series information in load data and proposed a building load prediction model based on a transformer network to improve the accuracy of building load prediction. This encoder-decoder block-based model can encode and decode all input data, capture sequence information from mapping vectors with user-defined dimensions, and learn important features through the Attention mechanism. In addition, input features were analyzed to verify the importance of each input feature, and to explaine the reasons for the impact of used features on the TRN-based model. Finally, the performance of the proposed model is evaluated using real data from an office building. Compared with other existing methods, the proposed model has the best prediction accuracy (RMSE, MAE, R^2 were 0.01, 0.03, and 0.98, respectively), and maintained the best predictive stability over a longer time (uncertainty ranged from −11% to + 11%). The results show that the proposed method can support the development and optimal operation of energy-saving HVAC systems, thereby lowing power consumption.

1. Introduction

The rapid growth of the world's economy and population is accelerating the increase of the global primary energy demand, from 145 billion MWh (in 2010) to an estimated value of 203 billion MWh (in 2046), an increase of 46% [1] over the 30 years. More than 39% of energy consumption comes from the construction industry, which leads to one-third of greenhouse gas emissions worldwide [2]. Saving energy in buildings is therefore extremely significant; it is reported that 30–80% of building energy consumption could be saved by using building technologies that are currently available [3]. Inside buildings, there are various service systems but the HVAC systems account for the largest proportion of the total building energy consumption, such that HVAC systems consumed more than 50% of the total building energy in the United States of America (USA) [4]. As a result, the most effective method for reducing the overall building energy consumption is to decrease the energy consumed by HVAC systems. The success of lowering the HVAC system's energy consumption depends largely on balancing the building thermal energy demand and the HVAC system's thermal energy supply. This is a tremendous challenge due to the constantly varying load caused by the time-variable weather conditions and the internal load, along with the HVAC system's limited operational flexibility [5]. Another issue with the conventional design method of the HVAC system is that, to make the HVAC system more reliable, the HVAC system's configuration is designed using the peak value of the building load (both heating and cooling loads). Moreover, a safety factor or redundancy HVAC system is adopted to ensure the designed system can fulfill all uncertain heating/cooling loads. The heating/cooling size of the designed HVAC system, therefore, is always oversized and the system's energy efficiency seriously decreases when the working conditions are different from the designed conditions [6]. As a result, predicting instantaneous building thermal load becomes necessary to reduce the HVAC system's energy consumption.

* Corresponding author.
E-mail address: lilong@hrbust.edu.cn (L. Li).

https://doi.org/10.1016/j.enbuild.2023.113409
Received 5 June 2023; Received in revised form 16 July 2023; Accepted 25 July 2023
Available online 28 July 2023

Nomenclature

Abbreviations

ANN	*artificial neural network*
DNI	*direct normal irradiance*
DOM	day of month
HOD	hour of day
HVAC	heating, ventilation, and air −conditioning
IoT	*Internet of Things*
LSTM	*long short −term memory*
CLM	*the hybrid prediction model* (*CatBoost*, *LightGBM*, *MLP*)
MAE	*mean absolute error*
MLP	*multilayer perceptron*
MOY	*month of year*
Q_c	*building cooling loads*
R^2	*coefficient of determination*
RC	*resistance and capacitance*
RH	*relative humidity*
RMSE	*root mean square error*
RNN	*recurrent neural network*
SVM	*support vector machine*
T_{db}	*dry bulb temperature*
TRN	*Transformer network*
V_{win}	*wind speed*
$\rho_{X,Y}$	*Pearson's correlation coefficient*

1.1. Literature review

At present, there are three types of traditional and commonly used methods for predicting building thermal load, namely, a white-box physical model based on physical information such as buildings, weather, and internal activities, a gray-box reduced-order model based on simplified wall structures and their thermal resistance and capacitance, and a black-box data-driven model based on pure data information.

For the white-box physics-based model, first, a detailed physical model needs to be completed for the simulated building, which should include the following information about the building: geometry, thermal property information (including thermal conductivity, heat capacity, etc.), and internal activity (such as the activity of people and electrical devices). Then, the model needs to be validated to ensure its accuracy, using correlated weather data, recorded internal activity, and monitored thermal load. Although commercially available and mature software programs (EnergyPlus, TRNSYS, etc.) already exist, which have built-in modules allowing users to easily implement building models [7], there are still several disadvantages to this method. First, this method is too complicated to use, because it needs many assumptions to implement the heat and mass transfer equations for calculating the thermal load. The difference between the assumptions and the actual heat/mass process could lead to serious differences in results [7]. Second, this method needs too many detailed inputs, which are both time and effort-consuming. In some cases, it is even unrealistic to obtain all the required inputs. Finally, the uncertain and inaccurate inputs used in the simulations could cause unacceptable differences between the model and actual results [8].

The gray-box reduced-order model can reduce the amount of input information required in white-box physics-based models [9], by simplifying wall structures along with their thermal resistance and capacitance (*RC*) to predict the heat flux through building walls [10]. The parameters used in the model calculations, such as the Rs and Cs, are obtained by minimizing prediction errors, rather than physical parameters of buildings derived from measurement data, effectively reducing required input information [11]. The existing RC models include the 1R1C model proposed by Wang et al. [12], 2R2C and 3R2C models developed by Wang and Xu [13], and 3R3C and 5R4C models analyzed by Blum et al. [14]. Compared to the higher-order models, the second-order RC model was verified as significantly reducing calculation time without compromising calculation accuracy [15]. However, a key issue for RC models is that they only calculate external heat gains without considering internal heat gains. To address this issue, researchers implemented studies for RC models that can consider internal gains. For example, Lin et al. [16] used the hourly cooling load factor method to predict the hourly cooling load of buildings. The internal gains can be estimated by using the sub-models; Ji et al. [17] developed an updated simplified thermal network model integrated with the sub-metering system to forecast the cooling load of buildings. In this model, the sub-metering system was adopted to reflect internal heat gains. The results indicated that the proposed model could forecast the building cooling load, along with internal heat gains, with high accuracy. Compared with the white-box physics-based model, the gray-box reduced-order model can significantly reduce the required input information and calculation time. It also has a self-adaptive ability to improve model accuracy when there is measured data for the Rs and Cs. But the precision of the model still cannot be guaranteed [18], because internal heat gains contribute an increasing proportion in modern buildings, especially when the building's envelope has a high efficiency level [19].

To overcome the limits of the two above models, researchers turned attention to the black-box data-driven model, which is a purely data-driven model using either shallow-machine learning or deep learning algorithms. It is largely attributed to the development of the Internet of Things (IoT) technology for the model's success, which can monitor and collect increasing amounts of data. With enough historical data, it can fully learn the building load patterns and complete modeling. Black-box data-driven models have been studied since the 1980 s. Studies mainly focus on the investigation of various algorithms applied to forecast the thermal load of buildings. A linear regression model used to predict the thermal load of a large building was first proposed in 1984 [20]; afterward, researchers investigated support vector machines (SVM) [21], artificial neural networks (ANN) [22], extreme learning machines [23], multilayer perceptron (MLP) [27], regression tree [24], random forest [25], Hierarchical Mixture of Experts [26], XGBoost [18], and long short term memory (LSTM) models [28]. Researchers also compared the accuracies of various algorithms used to forecast the thermal load of buildings to determine the best algorithm. For instance, Li [29] compared SVM and ANN models to predict hourly building cooling loads. The findings indicated that the SVM algorithm could predict the cooling load with higher accuracy; Wang et al. [19] compared seven shallow machine learning, two deep learning, and three heuristic methods. They concluded that the LSTM model was better for predicting short-term (1 h ahead) building cooling load, and the XGBoost model was better for forecasting long-term (≥24 h ahead) building cooling load. Compared with the single model, the hybrid method has significant improvement in short-term load forecasting. Guo et al. constructed four hybrid models for improved prediction accuracy of heating and cooling loads [30]. Matthew Motoki et al. [31] used three algorithms to complete the load forecasting task—CatBoost [32], LightGBM [33], and MLP, and then ensembled the model predictions using weighted generalized mean, achieving better results in forecasting accuracy. Compared with the white-box and gray-box models, the black-box model just needs enough historically recorded data from the building without requiring too much building information (e.g. Rs and Cs), or exact information (e.g. internal gains).

The Transformer algorithm was developed by Vaswani et al. in 2017

[34]. The core principle of the Transformer model is the self-attention mechanism, which is mainly based on matrix multiplication in its specific implementation, so that it can capture the dependency between any vectors in the input sequence, independent of the distance between vectors. Meanwhile, the Self-Attention mechanism is less complex, has fewer parameters, and requires less computing power, the results of each step do not depend on the results of the previous step, so the effect is better. Transformer, through its special attention mechanism, not only supports parallel training, but also can learn the timing information in the sequence, and improve the model training speed by an order of magnitude on the premise of ensuring high accuracy. Based on the above characteristics, Transformer has attracted extensive attention from researchers since it was proposed, and has shown excellent application effects in many fields such as machine translation [35] and image identification [36]. Mohan Li et al. proposed an online attention mechanism, known as cumulative attention (CA), for streaming Transformer-based automatic speech recognition (ASR) [37]. LIM B et al. [38] achieved significant performance improvement in the prediction fields of power load, transportation, retail, stock, and so on. Transformer algorithm has also achieved remarkable results in Natural Language Processing (NLP). Since NLP and time series are sequence information, Transformer algorithm is also gradually applied to time series forecasting tasks. Li et al. [39] have achieved good results in time series forecasting. Thus, the Transformer algorithm is a more promising method for building thermal load forecasting.

1.2. Research gap and objectives

The building load consisted of time-series data containing sequential information, which reflected the variation in the internal heat gains. Whether the time series information contained in the load forecasting can be effectively processed directly affects the accuracy of the prediction.

At present, all the shallow-machine learning methods and deep-learning algorithms proposed in research cannot capture and retain information in input sequences (except the LSTM algorithm), and cannot efficiently and accurately analyze time-series data. The LSTM (a special form of the recurrent neural network (RNN) algorithm) alleviates the problem of gradient explosion and disappearance in RNN by adding three control gates and one unit state. Although the LSTM can handle the long-term and short-term dependence in time series data to a certain extent, when the input sequence is too long, problems such as historical information dilution and sequence information loss still exist. Every recursion of LSTM is accompanied by information loss, as a result, its ability to capture dependencies under the condition of inputting long sequences declines rapidly, that is, memory degradation occurs, and the sequential information captured by the LSTM algorithm became less relevant as the prediction horizon lengthened [18]. In addition, the cycle structure in the RNN model limits its need to input data in a serial manner, resulting in low training efficiency [40].

Another issue for the black-box data-driven model is that there is no research on performing a feature analysis (especially the analysis of solar radiation intensity, wind velocity, and time-related variables that could pattern the occupancy activity, internal heat gains, and building usage) for either the shallow-machine or deep-learning methods, due to limitations of the recorded data.

To solve the problem of traditional models being unable to capture and retain the time-series information, and fill the research gap in feature analysis of shallow machines or deep learning methods, this paper first analyzes the Transformer algorithm in depth, then proposed a novel prediction model based on the Transformer algorithm (TRN-based model) to forecast the building cooling and heating load (only cooling load was analyzed in this paper). Then, the paper implemented an overall analysis of the above-mentioned features to determine the importance of each feature. The contributions can be summarized as follows:

1. A Transformer algorithm based model (TRN-based model) was proposed for the building load forecasting.The attention mechanism of this model allows for modeling the dependence of input and output sequences, without considering their distance in the sequence, and can capture the correlation between sequences, which could capture and retain sequential information from long-term time-series data, effectively improving the accuracy of prediction.
2. Using the TRN-based model for feature analysis and ranking the importance of features, the reasons for the impact of features on the TRN-based model were analyzed.
3. Taking real data as an example, the performance of the proposed model was evaluated, and the advantage of the TRN-based model was analyzed by comparing its performance with shallow-machine (XGBoost),deep-learning models (LSTM), and hybrid prediction model (LightGBM, CatBoost, MLP).
4. The accuracy and result errors were analyzed for the TRN-based model when using different numbers of input features.

2. Research methodology

Fig. 1 presents the research outline for this study, as follows:

1) Generate sequential input dataset: First, an original dataset containing direct normal irradiance (DNI), wind speed (V_{win}), relative humidity (RH), dry bulb temperature (T_{db}), building cooling loads (Q_c) were created based on historical collected real data. Second, time-related features were extracted from the original dataset, including the month of the year (MOY), day of the month (DOM), and hour of the day (HOD). These new features were combined with the original data to form a new dataset, containing time-related variables that reflected internal heat gains. After determining the correlation between the two variables, a sequential input dataset was generated using the new dataset with the sequential information pattern.
2) Train the Transformer-based model: The sequential input dataset was imported into the TRN-based model, and the parameters and hyper-parameter setting was conducted on the Transformer-based model to train the algorithm.
3) Model validation and comparative analysis: Finally, the performance of the TRN-based model was evaluated using three performance metrics: Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and coefficient of determination (R^2), and compared with state-of-art approaches from both shallow-machine (XGBoost) deep-learning models (LSTM) and hybrid prediction model (CatBoost, LightGBM, MLP).
4) Feature analysis: The impact of the amount and the dimensionality of input features on model performance was discussed, the importance ranking of model input features was obtained, and the reasons for the impact of important features on the model were explained.

The following paragraphs discuss the extraction of time-related features and generation of the sequential input dataset, the algorithms used for the TRN-based model, as well as the state-of-art approaches from shallow-machine and deep-learning models used for comparison, and the evaluation of the model's performance.

2.1. The algorithm for the TRN-based model

The TRN-based model proposed in this paper was developed from Vaswani et al.'s [34] Transformer algorithm, which is mainly composed of encoder-decoder blocks. Through four steps of input-encoding–decoding-output, the Transformer algorithm has good feature extraction ability, solving the problem of poor parallel computing ability caused by sequence dependencies in RNN, and improving the defect of information loss in seq2seq structure. The TRN-based model's

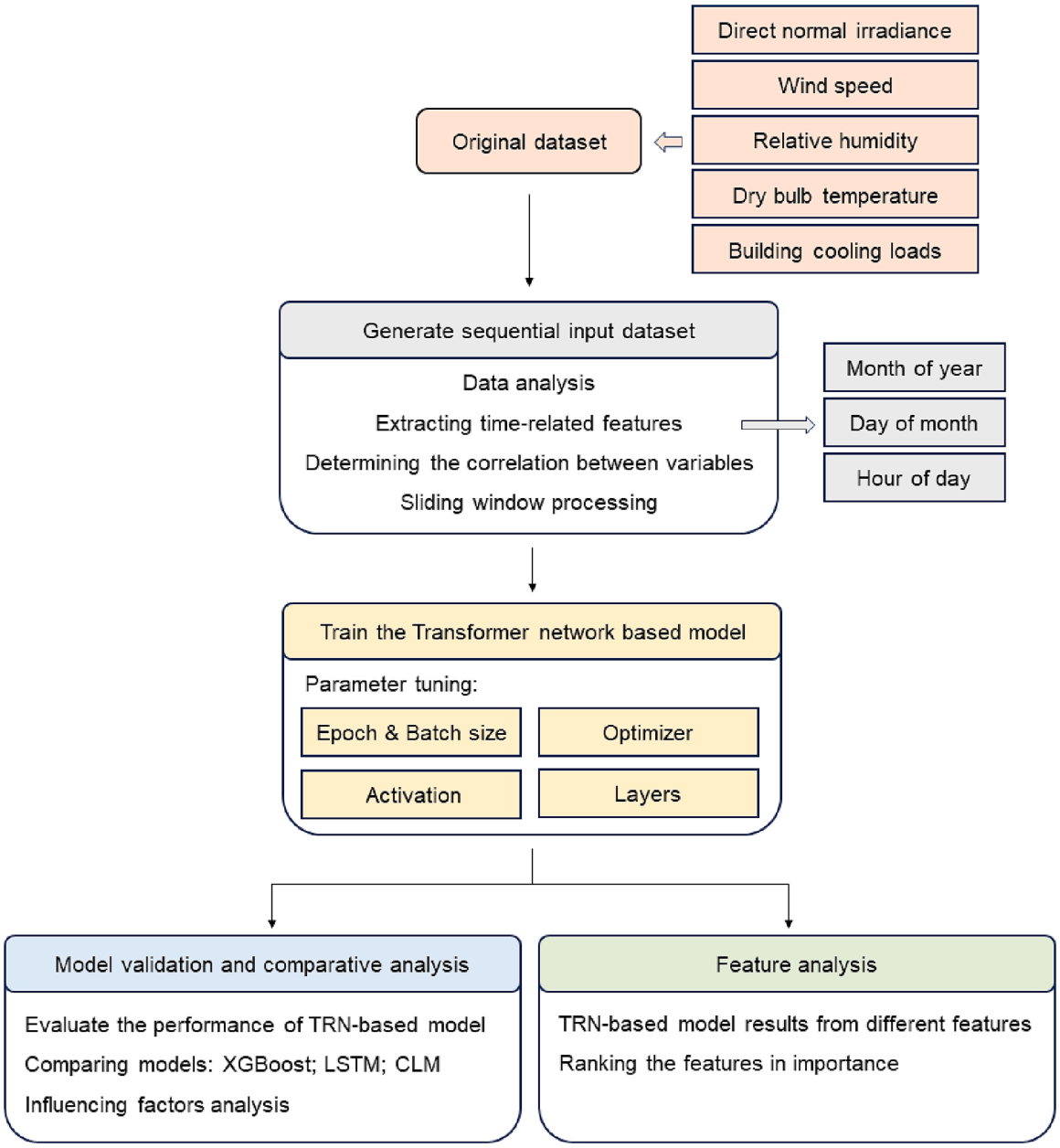

Fig. 1. Research roadmap.

architecture is shown in Fig. 2.

First, the sequential input data (shown as the 'Encoder Input') was mapped into a vector with d_{model}-dimensions, and then Positional Encoding with sine and cosine functions was used to encode sequential information in the time-series data through the element-wise addition of the input vector with a positional encoding vector. The positional encoding vector can represent the position of the current data and the distance between different data. The position formulation is shown in eqs. (1) and (2) [34].

$$\check{P}_{(\bar{\bar{p}},2i)} = sin\left(\frac{\bar{\bar{p}}}{10000^{\frac{2i}{d_{model}}}}\right) \tag{1}$$

$$\check{P}_{(\bar{\bar{p}},2i+1)} = cos\left(\frac{\bar{\bar{p}}}{10000^{\frac{2i}{d_{model}}}}\right) \tag{2}$$

where $\check{P}$ is the positional encoding result; $\bar{\bar{p}}$ is the position of the input variables; i is the position of the input variable and d_{model} is the dimension of the input data. Even positions are encoded using sine, and cardinal positions are encoded using cosine.

Then, the positional encoding vector was imported into the encoder block to produce a d_{model}-dimensional vector that feeds the decoder block. The output data (shown as the 'Decoder Input') was processed using the same methods as the encoder procedure and imported into the decoder block. The decoder block then used the encoder block's d_{model}-dimensional feed vector and the decoder input's positional vector to generate a unique vector that contained probability information to calculate the outputs. This unique vector then calculated the probability of the outputs using 'Linear Mapping' and 'Softmax' layers. Two ANN layers were connected to the 'Softmax' layer to convert the probability into the model's final outputs (shown as the 'Decoder Output').

The composition of the encoder and decoder blocks is shown in Fig. 3. The encoder block had two identical encoder layers, with each

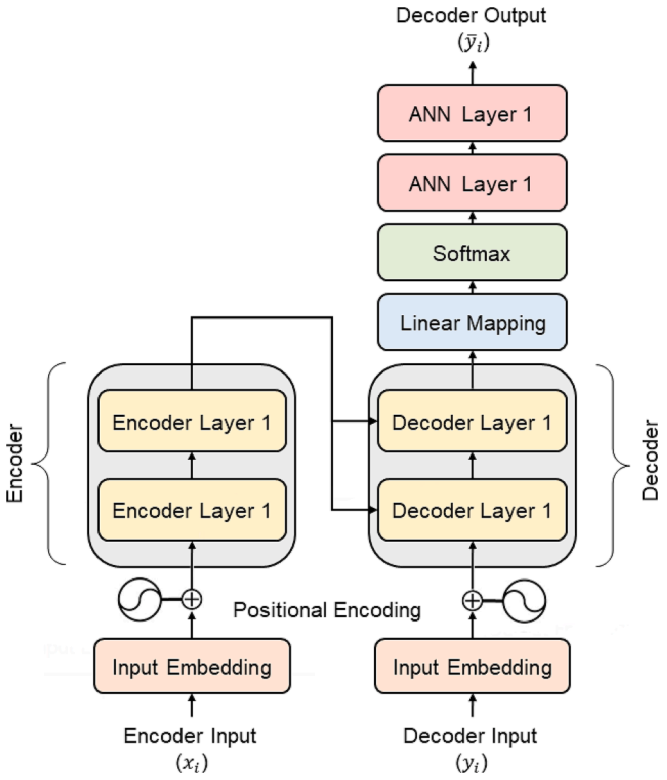

Fig. 2. The Transformer network-based model architecture.

layer having four series-connected sectors. The first sector was called 'Multi-head attention', and its basic principle is based on the Dot-product Attention mechanism. The Dot-product Attention mechanism maps a query and a set of key value pairs to an output, and the received data is the input or output of the previous encoder. Multiplying by different weights yields the Query, Key, and Value vector matrices. The equation explanation is shown in (3), and the calculation method for attention values is shown in equation (4) [34].

$$\begin{cases} \zeta = \dot{M}^{\zeta} x_i \\ \lambda = \dot{M}^{\lambda} x_i \\ \omega = \dot{M}^{\omega} x_i \end{cases} \tag{3}$$

$$Attention(\zeta, \lambda, \omega) = softmax\left(\frac{\zeta \lambda^T}{\sqrt{d_k}}\right) \tag{4}$$

where ζ, λ and ω are the query, key and value vectors; $\dot{M}^{\zeta}, \dot{M}^{\lambda}$ and $\dot{M}^{\omega}$ are the matrix used to calculate query, key and value vectors; x_i is the input variables; d_k is the correction parameter used to adjust the attention value within a certain range.

The Multi-head Attention mechanism concatenates the single attention results of ζ, λ and ω through different linear transformations, allowing the model to focus on information from different representation subspaces, which has a stronger feature extraction ability than single attention with the same number of parameters. Its structure is shown in Fig. 4.

The second sector was called 'Add & Normalize', where the attention values were then normalized to ensure the model's robustness. After normalization, a feed-forward neural network layer was used to calculate the feed vector used in the decoder block, The feedforward neural network is composed of two linear transformations, in which ReLU is activated, which can enhance the nonlinear fitting ability of the model. Before the feed vector was sent to the decoder block, it was normalized again to ensure its robustness.

The decoder block had two identical decoder layers and the same sectors as the encoder layer; in addition, it had "Encoder – Decoder Attention" and "Add & Normalize" sectors. The difference between "Encoder - Decoder Attention" and the self-attention mechanism in the Encoder block is that its input is not from the same sequence, but rather applies the feed vector (from the encoder block) to the decoder input: the encoder output λ, ω and the decoder's multi-head attention output ζ. Its structure is shown in Fig. 5. By utilizing this structure, all information

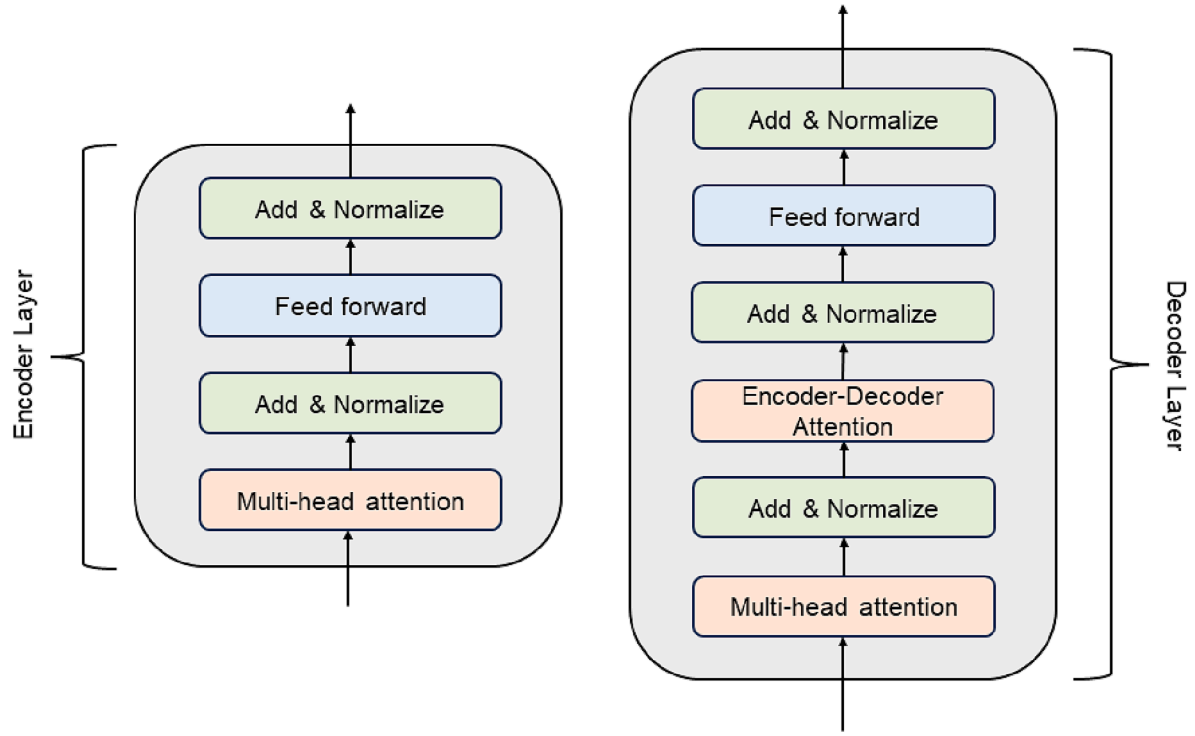

Fig. 3. The composition for the 'Encoder' and 'Decoder' layers.

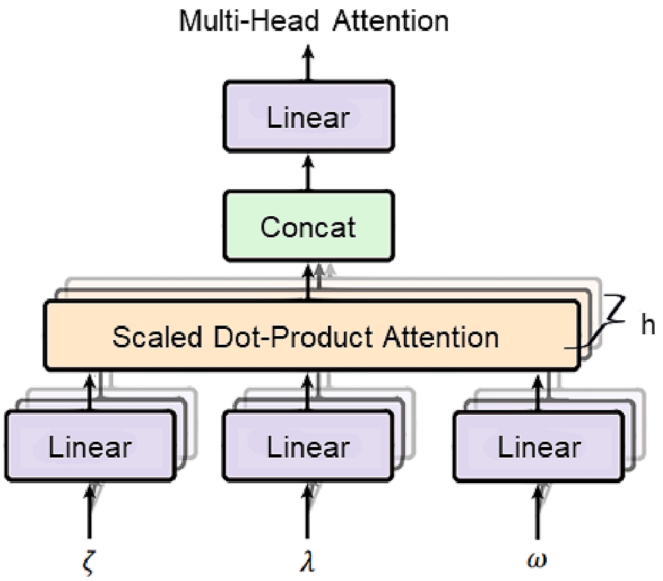

Fig. 4. The composition of the Multi-head Attention mechanism.

in the encoding can be fully utilized during decoding, generating attention between the data of the Encoder block and the Decoder block.

2.2. The state-of-art approaches from shallow-machine and deep learning models

XGBoost and LSTM are both mainstream models in prediction methods, considered the best shallow machine and deep learning models in relevant literature [18], and widely used in industry and theoretical experimental research. As a mixture model, the CLM model has been used in building prediction competitions and has shown good prediction ability [31]. Based on the above reasons, XGBoost, LSTM, and CLM models were selected as the benchmark models for the comparative experiment. Details are explained as follows.

2.2.1. For XGBoost model

Gradient Boosting Decision Tree (GBDT) is an additive model based on boosting ensemble idea [41]. During training, the forward distribution algorithm is used for greedy learning, and each iteration learns the t_{th} tree to fit the residual between the predicted results of the previous t_{th-1} tree and the true values of the training samples. The basic idea of XGBoost is the same as that of GBDT, but some optimizations have been made, such as using the second-order Taylor formula to expand and optimize the loss function to improve the calculation accuracy; using a weak predictor and the regularization term to simplify the model and avoid overfitting; adopting the Blocks storage structure to combine multiple predictors systematically, which can be calculated in parallel,

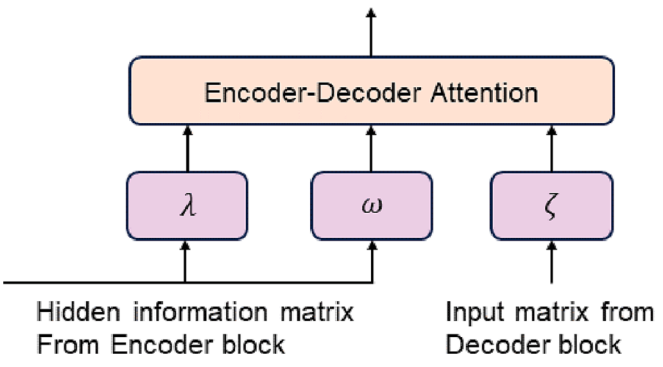

Fig. 5. The composition for Encoder – Decoder Attention.

etc [42]. This will enhance both the model's prediction accuracy and generalization capacity. The main equation[41] for XGBoost model is explained as follows:

$$\gamma^t = \sum_i^n l\left(y_i, \left(\breve{y}_i^{t-1}\right) + f_t(X_i)\right) + \Omega(f_i) \tag{5}$$

where γ^t is the objective of the optimization; i is the i-th predicted sample, n is the total sample number; y_i is the true sample value; $\breve{y}_i$ is the predicted sample value; f_t is the base learner added at the t-th iteration; X_i is the feature used for the i-th sample; and $\Omega(f_t)$ is the regularization term to avoid over-fitting (for more details about the XGBoost method, see [41]).

2.2.2. For LSTM model

The RNN is a family of neural network specifically designed to solve sequence and time-dependent event prediction problems. The LSTM algorithm is a variant of the RNN, which is capable of learning long-term dependencies by adding four gates to make all cell states incorporative. Among these, the most important gates are the input, forget, and output gates for the input, hidden, and output states. The other gate is a sigmoid function, which is used to modulate the output of these gates. By using these four gates, the problems of gradient vanishing and gradient explosion, found in the conventional RNN algorithm, can be avoided. Because the unit states can be remembered for longer time steps, this can remove the multiplication of small/big numbers so many times from each cell state. Equations [43] used in the LSTM algorithm are explained as follows:

$$f_t = \sigma\left(W_f \bullet |h_{t-1}, x_t| + b_f\right) \tag{6}$$

$$i_t = \sigma(W_i \bullet |h_{t-1}, x_t| + b_i) \tag{7}$$

$$o_t = \sigma(W_o \bullet |h_{t-1}, x_t| + b_o) \tag{8}$$

$$C_t = tanh(W_C \bullet |h_{t-1}, x_t| + b_C) \tag{9}$$

$$h_t = o_t \times tanh(C_t) \tag{10}$$

where f_t is the forget gate; i_t is the input gate; o_t is the output gate; $\sigma(x)$ is the sigmoid function; W_f, W_i, W_o are weight matrices used to update the state of forget, input, and output gates, respectively; b_f, b_i, b_o are bias vectors used to calculate the forget, input, and output gates, respectively; h_{t-1} is the activation value at time step t-1; x_t is the input at time step t; C_t is the memory candidate of the cell at time step t; $\tanh(\cdot)$ is the hyperbolic function used as the activation function; W_C is the weight matrix used to calculate the memory candidate; b_C is the bias vector used to update the memory candidate; and h_t is the activation value at time step t (for more details about the LSTM model, see [43]).

2.2.3. For CLM model

The model first used three algorithms, CatBoost, LightGBM, and MLP, to train individual models. CatBoost and LightGBM effectively improve the computational efficiency of GBDT. CatBoost is a GBDT framework based on symmetric decision trees, which is implemented with fewer parameters, supports categorical variables, and has high accuracy. The main pain point is to efficiently and reasonably process categorical features. The proposal of LightGBM has solved the problems encountered by GBDT in massive data, allowing GBDT to be better and faster used in industrial practice. MLP is a forward structured artificial neural network that maps a set of input vectors to a set of output vectors (for more details about the CatBoost, LightGBM and MLP, see [32,33,27]). Then, the cross validation methods were used to adjust the hyperparameters of all models. Finally, the weighted average method was used to combine individual model predictions to obtain the final predicted value, which reduced the risk of overfitting and improved

robustness. The workflow of the model is shown in Fig. 6.

2.3. Metrics used to evaluate performance

To compare the performance of the three selected models under different input feature scenarios, it is important to select several comparison indexes. In this analysis, the mean absolute error (MAE), root mean squared error (RMSE), coefficient of determination (R^2), and uncertainty were selected as the evaluated metrics. The MAE is a measure of errors between predicted and observed values that indicate the performance of the predicted to observed values: either over-prediction or under-prediction. The RMSE is a frequently used measure of the differences between values predicted by a model and those observed. The R^2 is the proportion of the variance in the dependent variable that is predictable from the independent variables; it measures how well the observed values replicate the model values. The uncertainty is the range of relative errors [44], which represents the degree to which the predicted value deviates from the true value. The equations of these three metrics are explained as follows:

$$MAE = \frac{\sum_{i=1}^{n}|\hat{y}_i - y_i|}{n} \tag{11}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}} \tag{12}$$

$$R^2 = \frac{COV(\hat{y}, y)}{\sqrt{Var(\hat{y})Var(y)}} \tag{13}$$

$$uncertainty = max\left(\frac{y_i - \hat{y}_i}{y_i}\right) - min\left(\frac{y_i - \hat{y}_i}{y_i}\right) \tag{14}$$

where i is the i-th observed/predicted value; n is the total number of the values; y is the observed value; $\hat{y}$ is the predicted value; $COV(\bullet)$ is the covariance function between observed and predicted values; $VAR(\bullet)$ is the variance function.

3. Case study

The analyzed data in this study were derived from an office building in the University of New South Wales, which was designed to monitor the performance of a solar-driven desiccant cooling system integrated with a dew-point evaporative cooler [45]. The whole building could be separated into two thermal zones. While Zone 1 was used for the working space, with an area of 46.7 m^2, Zone 2 was the storage room, with an area of 5.4 m^2. There were two windows in the west wall of Zone 1, each with an area of 2.2 m^2. Shading devices were used around the windows to prevent solar radiation from entering the building. In the room setting, the temperature was 25.5 °C and relative humidity was 60%. Dry bulb temperature (T_{db}), relative humidity (RH), direct normal irradiance (DNI), and wind speed (V_{win}) were recorded as weather data. The total cooling energy supply from the desiccant cooling system was recorded and calculated using the temperature difference between the sending and returning cooled air. Since the temperature sensors were closely installed in the inlet and outlet of the air ducts entering and leaving the building room. Thus, the total cooling energy supply (from the air-conditioning system) was thought equal to the total building cooling load (the sum of sensible and latent cooling loads). The recorded time period was from 8:00 am to 5:00 pm, between 1 October 2015 and 30 March 2016, and the recorded time interval was every hour. The view of the office building is shown in Fig. 7.

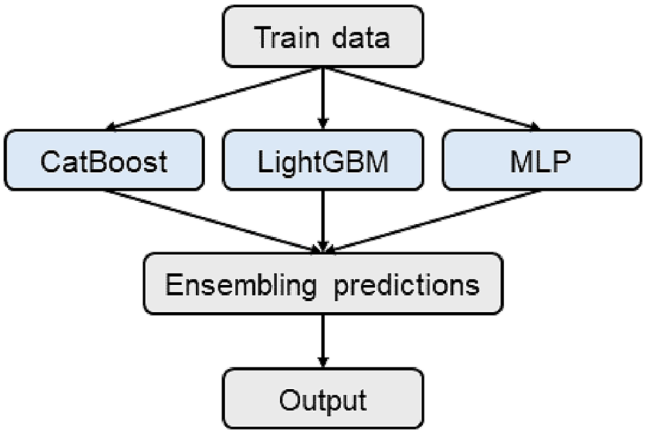

Fig. 6. Overview of the CLM model.

3.1. The creation of dataset with time-related features

Fig. 8 shows the plots for the weather data and building cooling load (Q_c). We could found that the recorded T_{db}, RH, DNI, wind velocity, and cooling load varied from 7.6 °C to 42 °C, 39% to 96%, 0 W/m^2 to 1010 W/m^2, 0.3 m/s to 15.1 m/s and 0 kW to 4.3 kW, respectively. It should be noted that not every day of the cooling season had a cooling load.

Fig. 9 shows the plots that indicated the hourly building cooling load for each day from October 2015 to March 2016, which helped to extract the time-related features. Three main points could be concluded from the plots. First, the cooling loads had various scales in different months. For example, the cooling load had scales of 0–2.2 kW (October 2015), 0–2.8 kW (November), 0–4.1 kW (December), 0–4.3 kW (January 2016), 0–3.2 kW (February), and 0–2.1 kW (March). Second, the cooling load changed every day in each month. For instance, in October 2015, the cooling load varied from 0 kW to > 2 kW for Day 28, and 0 kW to 1.5 kW for Day 25, while the cooling load was 0 kW or changed from 0 kW to < 1 kW for the remaining days. Finally, the cooling load firstly increased from 8:00 am to about 3:00 pm, then decreased after 4:00 pm for most of the recorded days. Thus, the initial analyzed features should include all weather-related variables (T_{db}, RH, DNI, V_{win}) and time-related variables (MOY, DOM, and HOD).

3.2. The generation of sequential input dataset

Before generating the sequential input dataset, the correlation between input variables was first implemented using Pearson's correlation coefficient ($\rho_{X,Y}$). $\rho_{X,Y}$ measures the strength of the association between two variables; it is the most popular method in the machine learning field for evaluating the linear correlation between two variables, X and Y. When the variables have perfectly positive or negative correlated features in the input dataset, there is a high chance of the model suffering from multicollinearity [46]. Multicollinearity is a phenomenon in which the predicted results can be skewed, when one or more variables in a multiple regression model can be predicted from the others with a high degree of accuracy. The $\rho_{X,Y}$ was calculated using the following equation [47]:

$$\rho_{X,Y} = \frac{COV(X,Y)}{\sigma_X \sigma_Y} = \frac{E\left[(X-\mu_x)(Y-\mu_y)\right]}{\sigma_X \sigma_Y} \tag{14}$$

where $COV(X, Y)$ is the covariance of X and Y; σ_X and σ_Y are the deviations of X and Y; and μ_x and μ_y are the means of X and Y. $\rho_{X,Y}$ has a range from –1 to + 1; Table 1 explains the relation between X and Y (when $\rho_{X,Y}$ has different values). Typically, ±0.6 is the most commonly adopted value to reduce correlated input variables. Fig. 10 shows the results of the $\rho_{X,Y}$ for the input variables.

As shown in Fig. 10, the $\rho_{X,Y}$ value varied from –0.51 (between DNI and RH) to + 0.4 (between T_{db} and HOD), which were within the range of ±0.6. This result showed that all features had a poor linear correlation with others, indicating that the input weather and time-related features

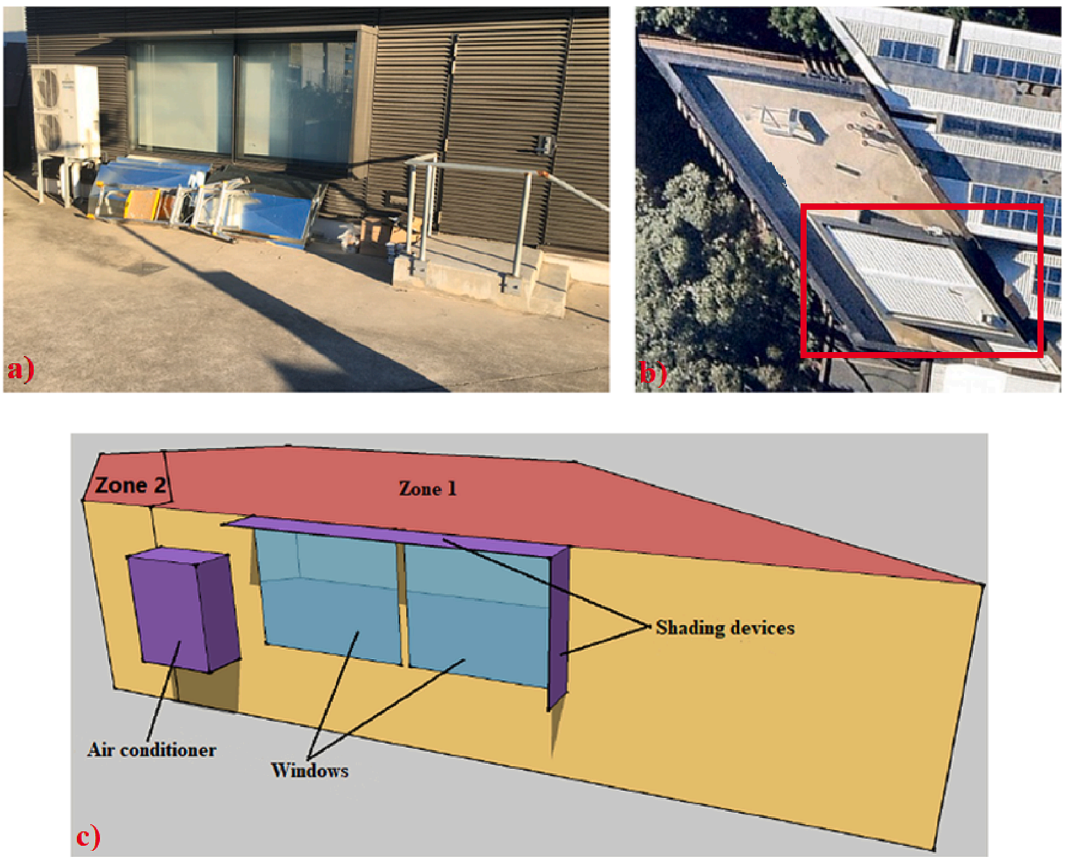

Fig. 7. (a). Front view of the office building; (b) Top view of the office building; (c) The overview of the office building in Sketchup [40].

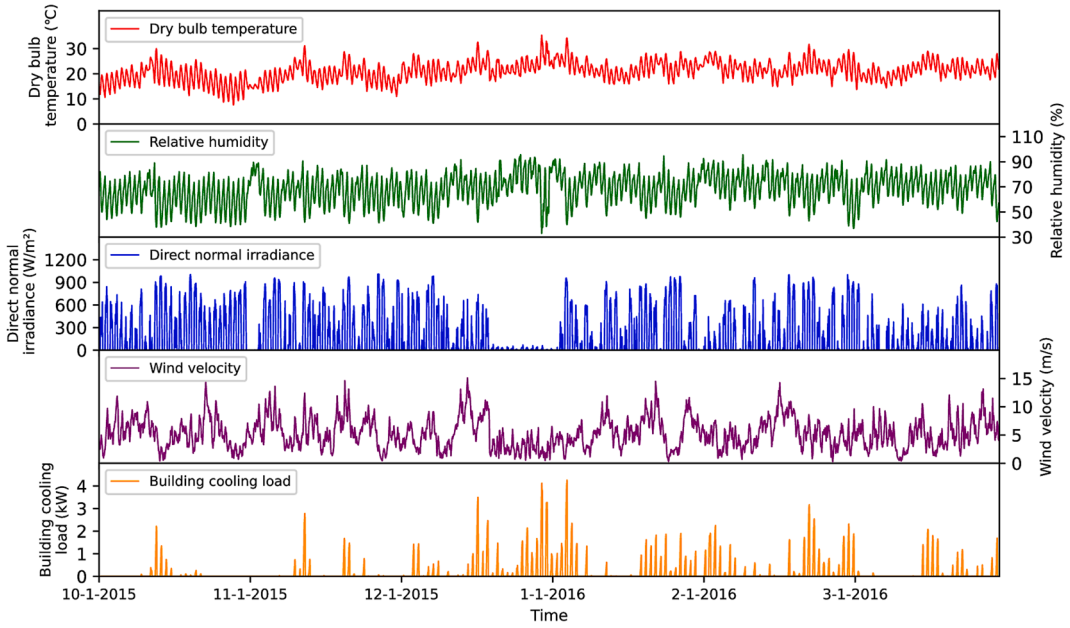

Fig. 8. Plots for weather and building cooling load.

were linearly independent of each other. Thus, all inputs could be used to train the TRN-based model.

Because the building cooling load has attributes of daily seasonality, it was expected that the prediction accuracy could be improved using measurements from the previous 24 h [48]. Thus, the previous 24-hour input variables were connected in series as sequential input data at hourly sampling intervals to predict the building cooling load in the 25th hour. The scheme for the final sequential input dataset is shown in Fig. 11.

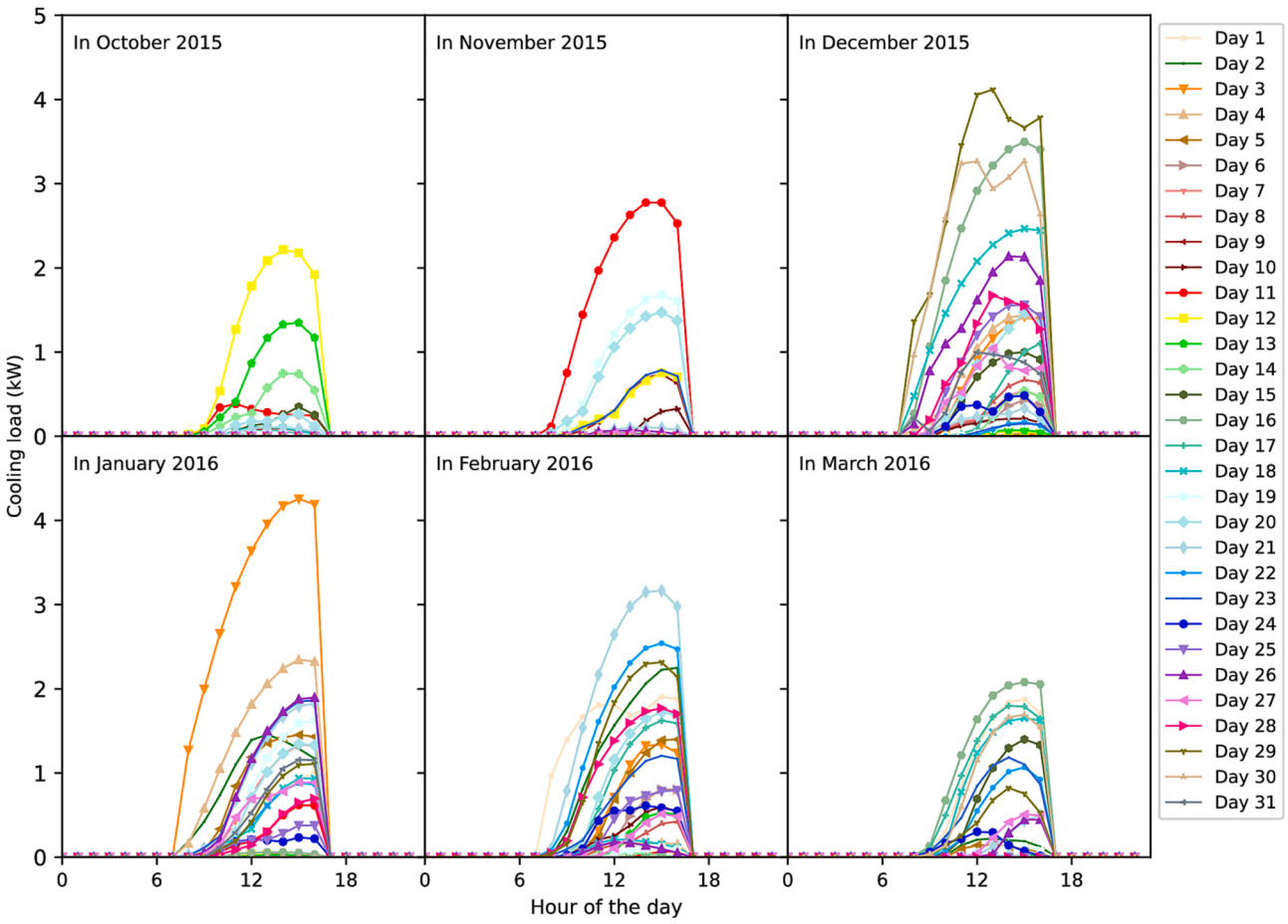

Fig. 9. Hourly cooling load for each day from October 2015 to March 2016.

Table 1
The settings of hyper-parameter.

Hyper-parameter	Value
Embedding size for each token	32
Number of attention heads (Layer 1)	2
Hidden layer size in feed forward network (Layer 1)	32
Dropout ratio for each hidden layer	0.1
Number of attention heads (Layer 2)	1
Hidden layer size in feed forward network (Layer 2)	32
Batch size	128
Optimizer	Adam
Epochs	80

4. Results and discussion

4.1. Model parameters and hyper-parameter setting

The adjustment of model hyperparameter is crucial for obtaining the best prediction performance. Through many experimental tests and comparisons, we have obtained the parameters and hyper-parameter of the TRN-based model as shown in Table.1, and an early stop step was defined to avoid overfitting.

The XGBoost, LSTM, CLM and TRN-based models were executed in Python using Sklearn [49], XGBoost [50], and Keras library [51], respectively. Before training the models, the sequential input dataset was split into two parts with a ratio of 0.8 to 0.2, meaning that 80% of the data was used as training data, with the remaining 20% used as testing data. The model's performance was then evaluated using the three metrics mentioned above. In order to increase the credibility of the results of each model, we trained all three models 10 times and selected the average of the results as the final model training result.

4.2. Results for the performance of the XGBoost, LSTM, CLM and TRN-based models

Table.2 shows the model performance results of the XGBoost, LSTM, CLM and TRN-based models. Among them, the LSTM model had the worst performance, with *RMSE* of 0.02/0.02, *MAE* of 0.05/0.06, and R^2 of 0.91/0.89. This was because, in the calculation process, the sequential information was obtained through concatenation. Although the LSTM model can handle sequential information, the prediction horizon was too long (in this paper, the horizon was 24), and the calculated values would be very large/small. The model suffered from the problem of gradient explosion/disappearance, which reduced the accuracy of the prediction results. The performance results of the XGBoost model were in the middle, with *RMSE* of 0.01/0.02, *MAE* of 0.04/0.05, and R^2 of 0.96/0.94. Although it did not suffer from gradient explosion/vanish as in the LSTM model, it was not specifically designed for processing chronological data, could not capture and retain sequence information, and was prone to overfitting problems. As a composite model, the performance results of the CLM model were relatively good, with *RMSE* of 0.01/0.02, *MAE* of 0.03/0.04, and R^2 of 0.97/0.96, but it also cannot solve the problem of long-term dependency learning of sequences. Therefore, its performance was lower than that of TRN-based models. The TRN-based model had the best performance, with *RMSE* of 0.01/0.01, *MAE* of 0.03/0.03, and R^2 of 0.98/0.98. Because it modeled sequential information in parallel through attention values between every two inputs from the input dataset. The accuracy was thus

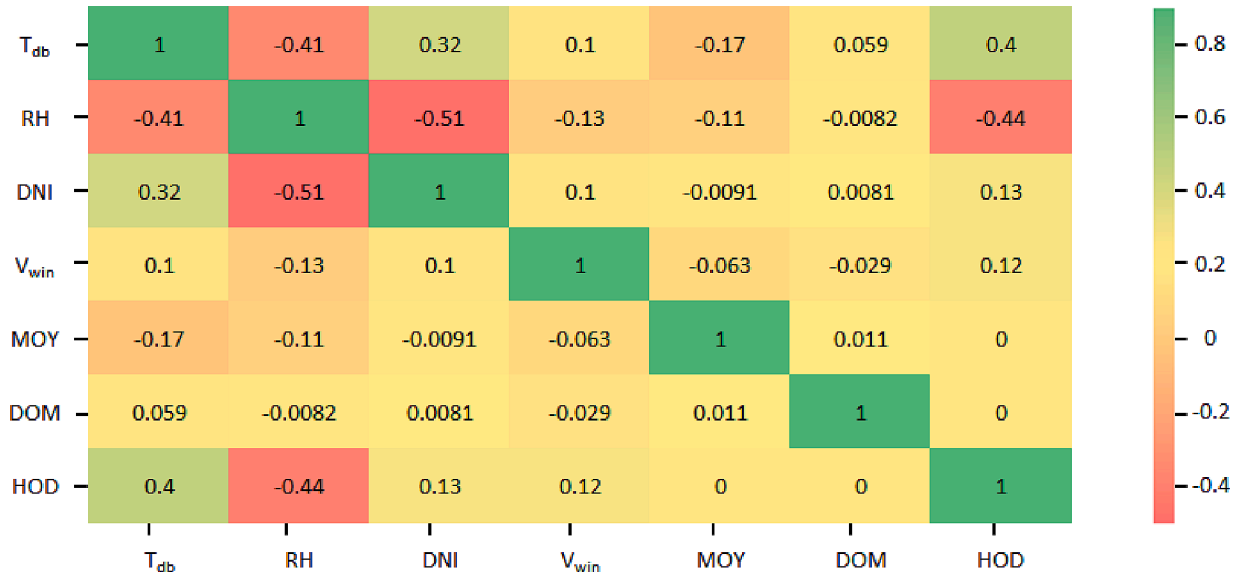

Fig. 10. Heat map for $\rho_{X,Y}$ values among the input features.

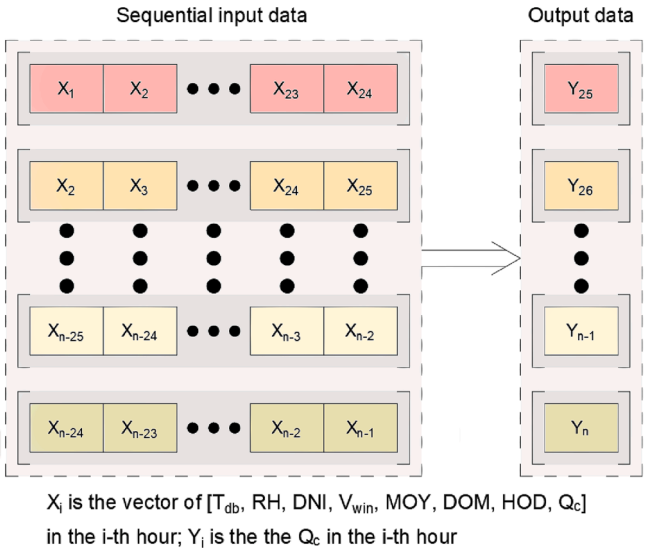

Fig. 11. The scheme for the final sequential input dataset.

Table 2
Model performance results for different models.

Model performance results						
Model name	For training dataset			For testing dataset		
	RMSE	*MAE*	R^2	*RMSE*	*MAE*	R^2
XGBoost model	0.01	0.04	0.96	0.02	0.05	0.94
LSTM model	0.02	0.05	0.91	0.02	0.06	0.89
CLM model	0.01	0.03	0.97	0.02	0.04	0.96
TRN-based model [a]	**0.01**	**0.03**	**0.98**	**0.01**	**0.03**	**0.98**

[a] The bold part indicates the proposed model.

guaranteed, no matter the length of the prediction horizon.

To investigate the uncertainty of the prediction results, the four models were used to predict the cooling loads of 10 typical days extracted from three different summer months. The results are plotted in Figs. 12–15. Fig. 12 shows that the XGBoost model could predict the building cooling loads with an uncertainty between –21% and + 31%. Fig. 13 shows the LSTM model's prediction, with an uncertainty between –17% and + 33%. Fig. 14 shows the CLM model's prediction, with an uncertainty between –32% and + 27%. Fig. 15 shows the TRN-based model's model's prediction, with an uncertainty between –11% and + 11%. The largest uncertainties always occurred in the first or last two working hours of the day. This is because when the system started/stopped working, it could not keep the inside space of the building in a thermally stable condition. Thus, the temperature difference between the sending and returning cooled air would have big fluctuations that would not reflect the real building cooling loads. As a result, the predicted cooling loads would have a large uncertainty. Another issue is that the TRN-based model had better and smoother predicted results, because it could maintain an uncertainty of ±11%, which did not suffer a large variation (such as between 0% and 31% for the XGBoost model, between 0% and + 33% for the LSTM model, and between 0% and 32% for the CLM model).

4.3. *Results for feature importance analysis in TRN-based model*

Feature analysis plays a key role in determining the prediction accuracy of the model, especially for which input features should be used in the data-driven black-box model. Too few features used in the model would cause an under-fitting problem, while too many features would result in an over-fitting problem [52]. This is because the model becomes more complex as the number of input features increase. When the model complexity rises, bias reduces and variance increases. If there is not enough information from the input features, the model's bias cannot be learned effectively, leading to a high bias value. As a result, the model accuracy would be poor, with an under-fitting problem. If there are too many input features, however, the model becomes too complex. Although the bias is very small, the variance would be too large, which could also reduce the model's accuracy. Thus, there is an optimum number/range for selecting input features, model complexity, and model accuracy (seen Fig. 16).

To eliminate the effects caused by data volume on the feature analysis, 50%, 75%, and 100% of the input data were used to conduct the feature importance analysis. Tables 3–5 show the feature analysis results for the TRN-based model, with different numbers of input features. Several things could be found by comparing the results in them. First, the volume size of the input data had almost no effect on the feature analysis. Second, when there was only one input feature, T_{db} was the most important; it made the model have R^2 above 0.89, *RMSE* smaller than 0.03, and *MAE* smaller than 0.06. Third, T_{db} and *HOD* were the two

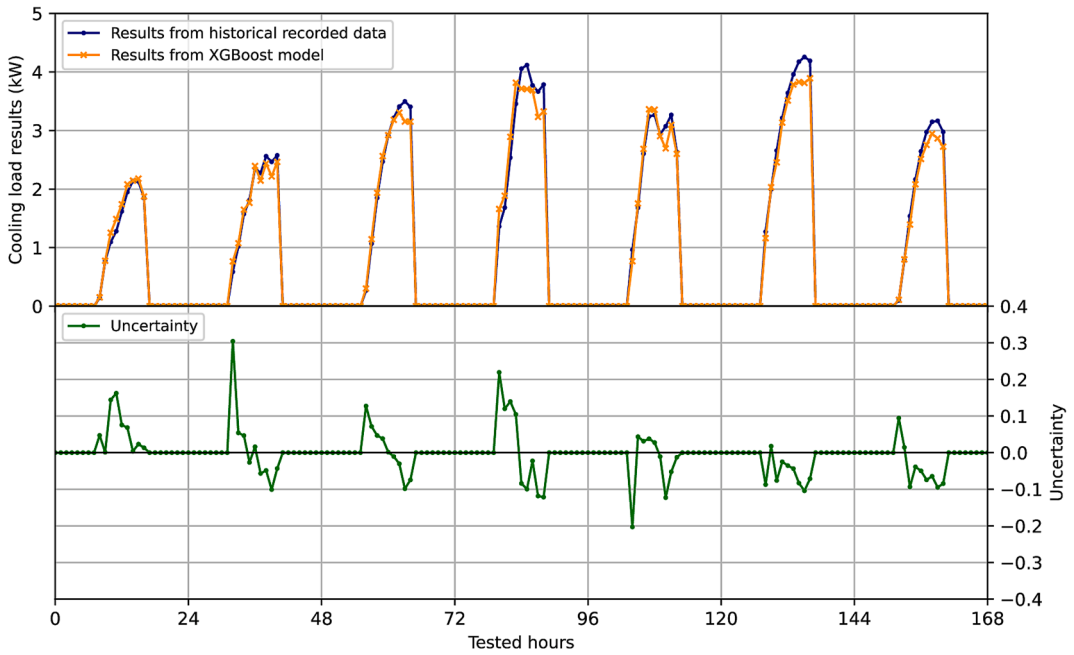

Fig. 12. Comparison between historical recorded cooling loads and predict cooling loads using XGBoost model.

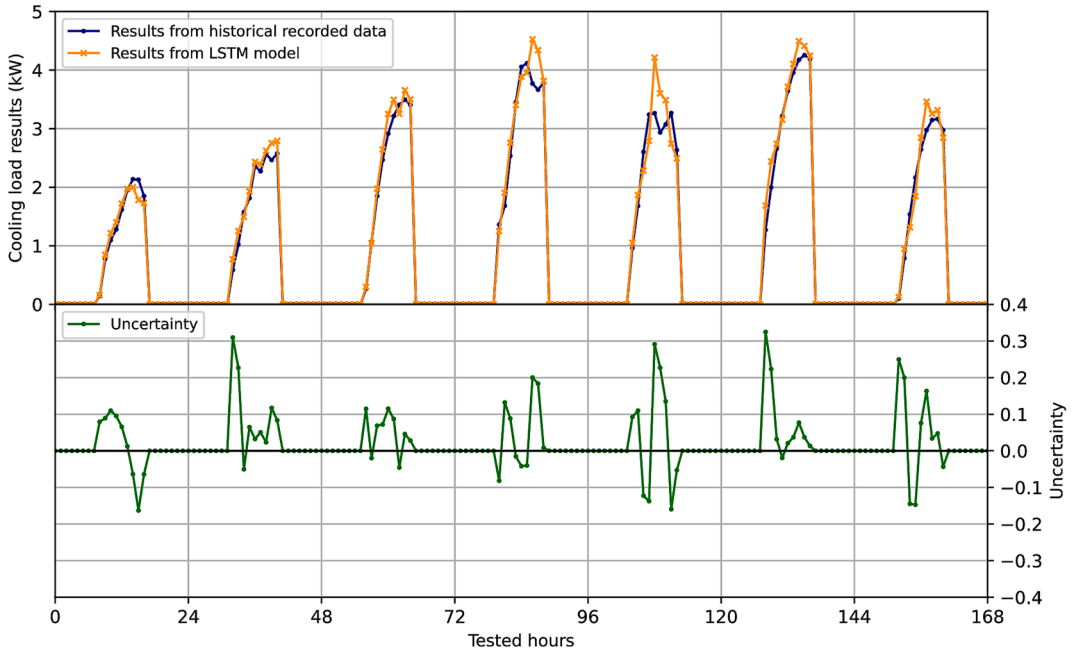

Fig. 13. Comparison between historical recorded cooling loads and predict cooling loads using LSTM model.

most important features when there were only two input features. T_{db} reflected the heat flux through the building walls from outside into the building and *HOD* helped the model to pattern the cooling load information caused by internal heat gains from people's activity and electrical equipment during the operation time of the cooling system. Fourth, *RH* and *DNI* were third- and fourth-most important features when there were three or four input features, because *RH* influenced the building's latent cooling load caused by ambient humidity, and *DNI* affected the building's solar heat gain through the windows. Last, *DOM*, *MOY*, and V_w had little influence on the performance of the TRN-based model, because their inclusion showed little improvement in the model's performance. Thus, the input features can be ranked in importance as follows: T_{db}, *HOD*, *RH*, and *DNI*.

Fig. 17 presents the plots for the predicted cooling loads and uncertainties of 10 typical summer days, with different numbers of input features ($[T_{db}]\rightarrow[T_{db}, HOD]\rightarrow[T_{db}, HOD, RH]\rightarrow[T_{db}, HOD, RH, DNI]$), the model's performance did not improve significantly (seen from Tables 2 and 3), but there was an obvious decrease in uncertainty of –50% to + 37% (for only one input feature), –26% to + 15% (two input features), –21% to + 14% (three input features), and –15% to + 16% (four input features). This implies that at least four features were required for the TRN-based model to have good performance (using the three evaluated metrics) and small uncertainties. This is because each of these four input features represents one part of the cooling load: T_{db} influences the heat flowing through the building walls, *HOD* affects the internal heat gains from

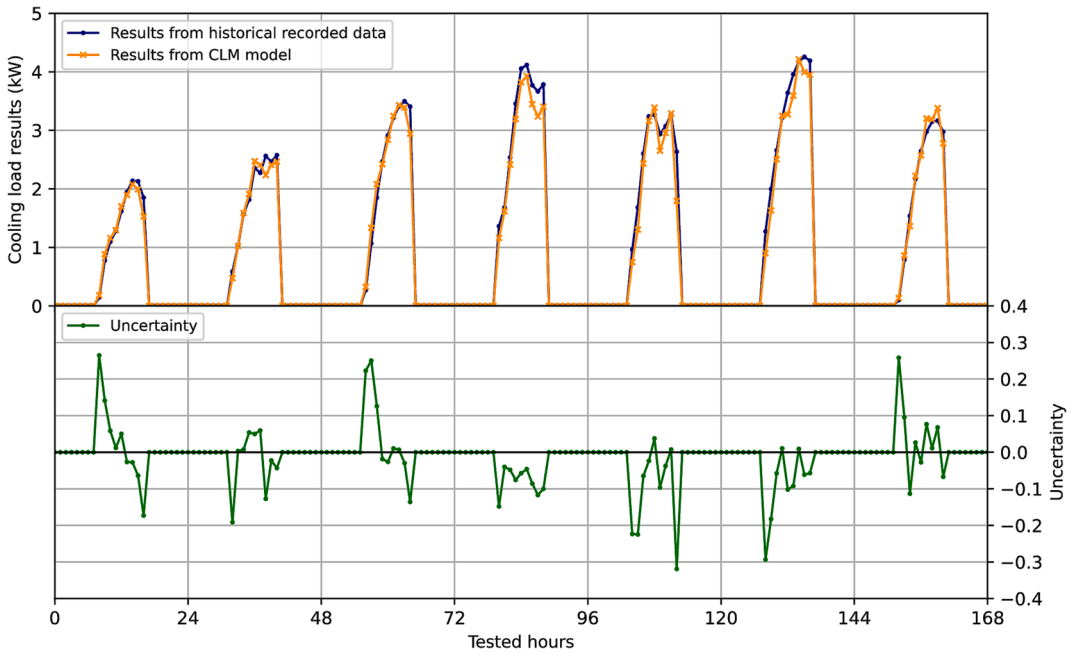

Fig. 14. Comparison between historical recorded cooling loads and predict cooling loads using CLM model.

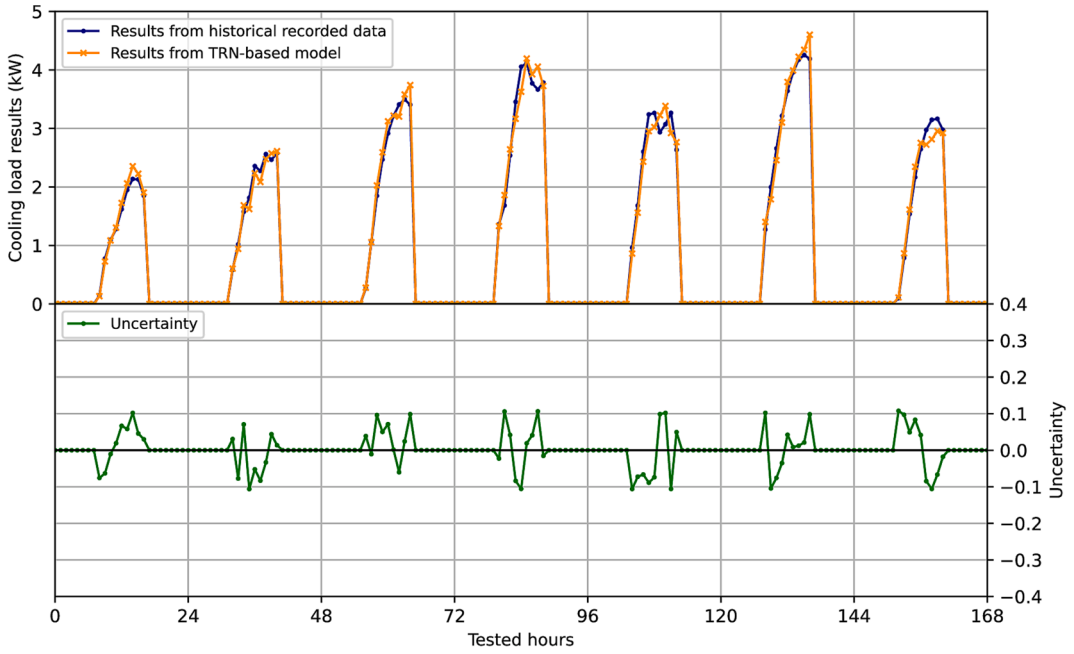

Fig. 15. Comparison between historical recorded cooling loads and predict cooling loads using TRN-based model.

people's activity and electrical equipment, *RH* affects the latent cooling loads, and *DNI* determines the heat going through the window.

5. Discussion

Compared to other comparative models, the Multi-head Attention mechanism of the TRN-based model can focus on information from different representation subspaces, allowing the model to obtain more data features under the same number of parameters. At the same time, the attention mechanism has good global dependency ability and can perform parallel calculations. The above reasons have significantly improved the prediction accuracy and uncertainty of the proposed model.

For the TRN-based model, the most important features are T_{db} and time-related features (HOD). Using these two features, the model can have an RMSE < 0.01, MAE < 0.05, and R^2 greater than 0.94. The uncertainty of the model results can also be maintained within the range of –26% to + 15%. Among these two features, T_{db} represents the heat flux through wall structures in the white- and/or gray-box models, which accounts for most of the cooling loads under this building's cooling load analysis, and time-related feature (HOD) patterns the internal heat gains because, in most cases, the occupant, lighting, and plug-load schedules are daily periodical behaviors. However, with a change in building type (residential, commercial, industrial, etc.), the importance of time-related features could change significantly, due to changes in internal heat gains in modern society [19]. Thus, when using data-driven black-

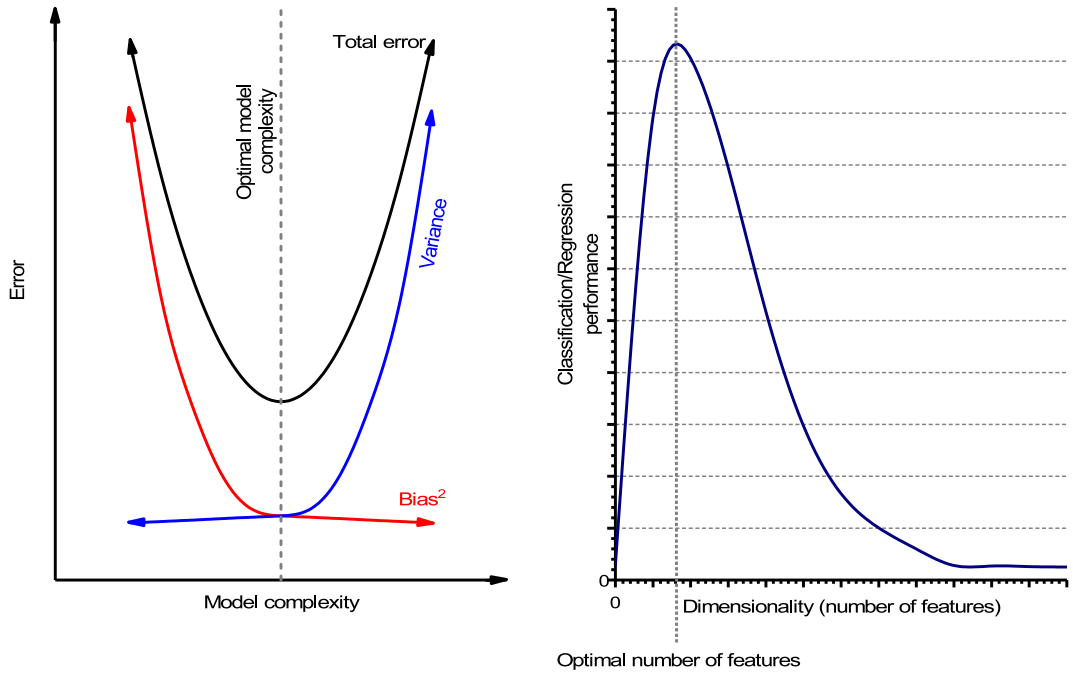

Fig. 16. Bias and variance contributing to total error (left)[47]; optimal number for models (right)[47].

Table 3
TRN-based model results for different feature selection scenarios using 100% data.

Results for model with one feature						
Selected feature(s)	For training dataset			For testing dataset		
	MAE	*RMSE*	R^2	*MAE*	*RMSE*	R^2
$\boldsymbol{T_{db}}$[a]	**0.06**	**0.02**	**0.89**	**0.07**	**0.03**	**0.86**
HOD	0.09	0.04	0.83	0.09	0.04	0.83
DNI	0.1	0.04	0.79	0.1	0.05	0.76
DOM	0.12	0.05	0.75	0.11	0.06	0.73
RH	0.1	0.07	0.69	0.1	0.06	0.71
MOY	0.11	0.08	0.65	0.1	0.07	0.67
V_w	0.14	0.09	0.56	0.13	0.09	0.59
Results for model with two features						
$\boldsymbol{T_{db}, HOD}$[a]	**0.05**	**0.01**	**0.94**	**0.05**	**0.02**	**0.93**
T_{db}, RH	0.04	0.02	0.93	0.06	0.03	0.86
T_{db}, V_w	0.04	0.02	0.92	0.05	0.03	0.88
T_{db}, DOM	0.04	0.02	0.92	0.05	0.03	0.87
T_{db}, DNI	0.04	0.02	0.91	0.05	0.03	0.86
T_{db}, MOY	0.04	0.02	0.89	0.05	0.03	0.85
Results for model with three features						
$\boldsymbol{T_{db}, HOD, RH}$[a]	**0.03**	**0.01**	**0.98**	**0.03**	**0.01**	**0.95**
T_{db}, HOD, DNI	0.02	0.01	0.98	0.03	0.01	0.95
T_{db}, HOD, V_w	0.03	0.01	0.97	0.04	0.01	0.96
T_{db}, HOD, DOM	0.03	0.01	0.97	0.03	0.01	0.96
T_{db}, HOD, MOY	0.04	0.01	0.96	0.04	0.01	0.96
Results for model with more than three features						
Whentherearemorethanthreefeatures	0.02–0.04	0.01	0.96–0.98	0.03–0.04	0.01	0.95–0.96

[a]The bold part indicates the best result when inputting different features.

box models to predict the building thermal load, time-related features should be analyzed according to the building type.

Compared with T_{db} and time-related features (HOD), RH and DNI make limited contributions to the model's accuracy, even though these two features have important influences on the building thermal load. There are two reasons why these features contribute little to this particular building. First, RH mainly affects the latent cooling load inside the building, through filtration and fresh air ventilation. In this building, however, the amount of filtration and fresh air ventilation is quite small (<10% of the total sending cooled air into the building [40]). In addition, if the ambient humidity is equal to or less than the setting point of humidity inside the building, it would not contribute to the latent cooling load. Considering the weather in Sydney, Australia, those times when the ambient humidity is higher than the setting point of humidity accounts for just 12% of the total cooling time (seen in Section 3.1, Fig. 5). Therefore, RH is not important in this building. Second, DNI

Table 4
TRN-based model results for different feature selection scenarios using 75% data.

Selected feature(s)	For training dataset			For testing dataset		
	MAE	*RMSE*	R^2	*MAE*	*RMSE*	R^2
Results for model with one feature						
$\boldsymbol{T_{db}}$ [a]	**0.05**	**0.03**	**0.9**	**0.07**	**0.04**	**0.85**
HOD	0.09	0.04	0.87	0.09	0.04	0.88
DNI	0.11	0.07	0.75	0.12	0.09	0.71
RH	0.12	0.08	0.69	0.12	0.11	0.62
MOY	0.14	0.09	0.67	0.13	0.1	0.65
V_w	0.14	0.11	0.57	0.14	0.14	0.54
DOM	0.14	0.11	0.57	0.13	0.14	0.52
Results for model with two features						
$\boldsymbol{T_{db}, HOD}$ [a]	**0.05**	**0.02**	**0.94**	**0.06**	**0.02**	**0.93**
T_{db}, MOY	0.04	0.02	0.94	0.06	0.03	0.89
T_{db}, DNI	0.04	0.02	0.93	0.07	0.04	0.86
T_{db}, RH	0.06	0.02	0.92	0.07	0.03	0.9
T_{db}, DOM	0.05	0.03	0.9	0.07	0.04	0.88
T_{db}, V_w	0.06	0.03	0.88	0.07	0.05	0.84
Results for model with three features						
$\boldsymbol{T_{db}, HOD, RH}$ [a]	**0.03**	**0.01**	**0.98**	**0.04**	**0.01**	**0.97**
T_{db}, HOD, DNI	0.03	0.01	0.98	0.04	0.01	0.96
T_{db}, HOD, V_w	0.04	0.01	0.96	0.05	0.02	0.92
T_{db}, HOD, MOY	0.05	0.01	0.96	0.06	0.02	0.94
T_{db}, HOD, DOM	0.04	0.01	0.96	0.05	0.01	0.95
Results for model with more than three features						
Whentherearemorethanthreefeatures	0.02–0.04	0.01	0.96–0.98	0.03–0.04	0.01	0.94–0.97

[a] The bold part indicates the best result when inputting different features.

Table 5
TRN-based model results for different feature selection scenarios using 50% data.

Selected feature(s)	For training dataset			For testing dataset		
	MAE	*RMSE*	R^2	*MAE*	*RMSE*	R^2
Results for model with one feature						
$\boldsymbol{T_{db}}$ [a]	**0.08**	**0.03**	**0.89**	**0.08**	**0.03**	**0.9**
DNI	0.07	0.03	0.86	0.09	0.05	0.82
V_w	0.09	0.04	0.83	0.1	0.05	0.82
HOD	0.08	0.04	0.83	0.09	0.04	0.84
RH	0.09	0.05	0.79	0.11	0.05	0.81
MOY	0.12	0.05	0.76	0.12	0.05	0.81
DOM	0.12	0.06	0.73	0.13	0.07	0.75
Results for model with two features						
$\boldsymbol{T_{db}, HOD}$ [a]	**0.04**	**0.01**	**0.97**	**0.04**	**0.01**	**0.97**
T_{db}, V_w	0.09	0.03	0.87	0.1	0.04	0.86
T_{db}, DNI	0.08	0.03	0.86	0.11	0.05	0.82
T_{db}, RH	0.09	0.03	0.85	0.12	0.05	0.81
T_{db}, MOY	0.09	0.04	0.8	0.1	0.06	0.78
T_{db}, DOM	0.09	0.05	0.79	0.1	0.06	0.79
Results for model with three features						
$\boldsymbol{T_{db}, HOD, RH}$ [a]	**0.04**	**0.01**	**0.98**	**0.05**	**0.01**	**0.95**
T_{db}, HOD, V_w	0.04	0.01	0.98	0.04	0.01	0.97
T_{db}, HOD, DOM	0.04	0.01	0.97	0.04	0.01	0.98
T_{db}, HOD, MOY	0.06	0.01	0.94	0.06	0.02	0.9
T_{db}, HOD, DNI	0.05	0.02	0.93	0.07	0.03	0.9
Results for model with more than three features						
Whentherearemore thanthreefeatures	0.02 - 0.05	0.01 - 0.02	0.97 - 0.98	0.03 - 0.04	0.01 - 0.02	0.96 - 0.98

[a]The bold part indicates the best result when inputting different features.

brings thermal load into buildings through window structures. In this building, however, the window area is small. Moreover, there are shading devices to prevent solar irradiance from entering the building and increasing the building thermal load. Therefore, DNI does not bring much thermal load into this building. It is crucial to include RH and DNI, however, if the building is located in a humid area with a high amount of filtration/ventilation, or if the building has a large area of window structures.

6. Conclusion

This study proposed a TRN-based model that was specifically designed to deal with time-series data (such as building cooling and heating load) with sequential information. The performance of the TRN-based model was then compared with state-of-art approaches from shallow-machine (XGBoost model), deep-learning models (LSTM model), and hybrid model (CLM model). The uncertainty of the four models was also compared to determine the difference between predicted cooling loads and historically recorded cooling loads. A comprehensive analysis of the input features was conducted to investigate the importance of each input feature. The conclusions can be summarized as follows:

1. The TRN-based model performs better than the XGBoost, LSTM and CLM models when dealing with time-series data (building cooling load) that contains sequential information. The model could have the *RMSE* of 0.01, *MAE* of 0.03, and R^2 of 0.98. The uncertainty of the predicted results was maintained within the range of ±11%;
2. The two most important features were T_{db} and time-related features (*HOD*). T_{db} reflected the heat flux through wall structures from outside the building to inside. The time-related feature reflected internal heat gains, such as people's activity and electrical equipment;
3. *RH* and *DNI* did not contribute much toward improving the model's performance (when using the selected evaluated metrics), but they

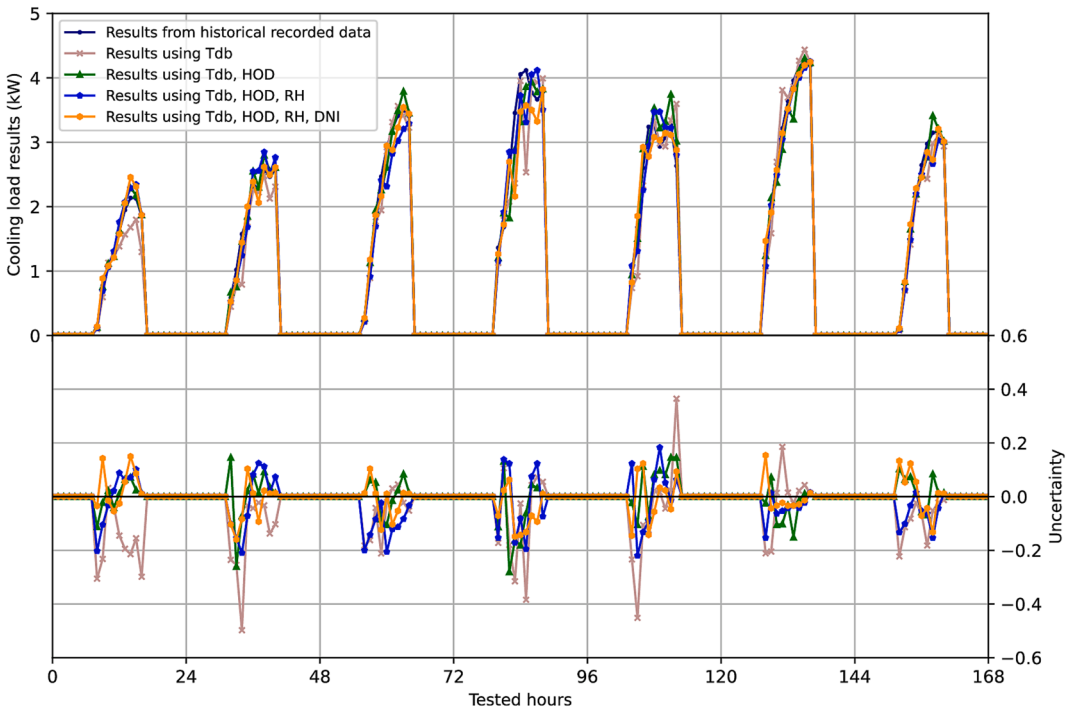

Fig. 17. Results for the TRN-based model with different numbers of input features.

helped to reduce the predicted result uncertainty for a single predicted point;

4. At least four features (T_{db},HOD,RH,DNI) were required for the TRN-based model to have good performance (using the three evaluated metrics) and small uncertainties. Each of these four input features represented one part of the cooling load: T_{db},HOD,RH, and DNI reflected the heat flux flowing through building walls, internal heat gains from people's activity and electrical equipment, latent cooling loads from external environment, and solar heat gain through windows, respectively.

This study is used for load forecasting of commercial buildings, in future work, as an expansion of the proposed methodology, we would like to further explore the following directions.

1) Research on different types of buildings.
2) Research on the impact of extreme climate on prediction results.

CRediT authorship contribution statement

Long Li: Methodology, Funding acquisition, Data curation, Writing – review & editing. **Xingyu Su:** Software, Writing – original draft. **Xianting Bi:** Software, Formal analysis. **Yueliang Lu:** Software. **Xuetao Sun:** Investigation, 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.

Data availability

The data that has been used is confidential.

Acknowledgements

The authors gratefully acknowledge the support of this research by the Shanxi Construction Engineering Group Co, China.

References

[1] British Petroleum, BP Energy Outlook: 2019 edition, https://www.bp.com/content/dam/bp/business-sites/en/global/corporate/pdfs/energy-economics/energy-outlook/bp-energy-outlook-2019.pdf [accessed 7 September 2020].
[2] D. Ürge-Vorsatz, L.F. Cabeza, S. Serrano, C. Barreneche, K. Petrichenko, Heating and cooling energy trends and drivers in buildings, Renew Sust. Energy Rev. 47 (2015) 85–98, https://doi.org/10.1016/j.rser.2014.08.045.
[3] C. Fan, F. Xiao, Y. Zhao, A short-term building cooling load prediction method using deep learning algorithms, Appl. Energy 195 (2017) 222–239, https://doi.org/10.1016/j.apenergy.2017.03.064.
[4] L. Pérez-Lombard, J. Ortiz, C. Pout, A review on buildings energy consumption information, Energy Build. 46 (2008) 454–458, https://doi.org/10.1016/j.enbuild.2007.03.007.
[5] W. Gang, S. Wang, G. Augenbroe, F. Xiao, Robust optimal design of district cooling systems and the impacts of uncertainty and reliability, Energy Build. 122 (2016) 11–22, https://doi.org/10.1016/j.enbuild.2016.04.012.
[6] Q. Cheng, S. Wang, C. Yan, F. Xiao, Probabilistic approach for uncertainty-based optimal design of chiller plants in buildings, Appl. Energy 185 (2017) 1613–1624, https://doi.org/10.1016/j.apenergy.2015.10.097.
[7] S. Imam, D.A. Coley, I. Walker, The building performance gap: are modellers literate? Build. Serv. Eng. Res. Technol. 44 (2017) 411–475, https://doi.org/10.1177/0144224476684647.
[8] X. Li, J. Wen, Review of building energy modeling for control and operation, Renew Sust. Energy Rev. 43 (2014) 517–543, https://doi.org/10.1016/j.rser.2014.05.056.
[9] X. Xu, S. Wang, A simplified dynamic model for existing buildings using CTF and thermal network models, Int. J. Therm. Sci. 47 (2008) 1249–1262, https://doi.org/10.1016/j.renene.2019.10.083.
[10] J. Yang, Z. Lin, H. Wu, Q. Chen, X. Xu, G. Huang, L. Fan, X. Shen, K. Gan, Inverse optimization of building thermal resistance and capacitance for minimizing air conditioning loads, Renew Energy 148 (2020) 975–986.
[11] J.E. Braun, N. Chaturvedi, An inverse gray-box model for transient building load prediction, HVAC&R Res. 8 (2002) 73–99, https://doi.org/10.1080/10789669.2002.10451290.
[12] Z. Wang, B. Lin, Y. Zhu, Modeling and measurement study on an intermittent heating system of a residence in Cambridgeshire, Build. Environ. 92 (2015) 440–446, https://doi.org/10.1016/j.buildenv.2015.05.014.
[13] Wang S, Xu X, Simplified building model for transient thermal performance estimation using GA-based parameter identification, Int J Therm Sci. 45 (2006) 479–38. doi: 10.1016/j.ijthermalsci.2005.06.009.

[14] Blum DH, Arendt K, Rivalin L, Piette MA, Wetter M, Veje CT, Practical factors of envelope model setup and their effects on the performance of model predictive control for building heating, ventilating, and air conditioning systems, Appl Energ. 242 (2019) 470–25. doi: 10.1016/j.apenergy.2018.11.093.
[15] T. Dewson, B. Day, A.D. Irving, Least squares parameter estimation of a reduced order thermal model of an experimental building, Build Env. 28 (1993) 127–143, https://doi.org/10.1016/0420-1383(93)90046-6.
[16] L. Duanmu, Z. Wang, Z.J. Zhai, X. Li, A simplified method to predict hourly building cooling load for urban energy planning, Energy Build. 58 (2013) 281–291, https://doi.org/10.1016/j.enbuild.2012.11.029.
[17] Y. Ji, P. Xu, P. Duan, X. Lu, Estimating hourly cooling load in commercial buildings using a thermal network model and electricity submetering data, Appl. Energy 169 (2016) 309–323, https://doi.org/10.1016/j.apenergy.2016.02.042.
[18] Z. Wang, T. Hong, M.A. Piette, Building thermal load prediction through shallow machine learning and deep learning, Appl Energ. 263 (2020), 114683, https://doi.org/10.1016/j.apenergy.2020.114683.
[19] Z. Wang, T. Hong, M.A. Piette, Data fusion in predicting internal heat gains for office buildings through a deep learning approach, Appl. Energy 246 (2019) 446–498, https://doi.org/10.1016/j.apenergy.2019.02.066.
[20] J.R. Forrester, W.J. Wepfer, Formulation of a load prediction algorithm for a large commercial building [accessed 7 September 2020], ASHRAE Trans. 90 (1984) 542–551, https://iifiir.org/en/fridoc/7461.
[21] H.X. Zhao, F. Magoulès, Parallel support vector machines applied to the prediction of multiple buildings energy consumption, J. Algorithm Comput. Technol. 4 (2010) 237–249, https://doi.org/10.1260/1748-3018.4.2.237.
[22] Y. Wei, L. Xia, S. Pan, J. Wu, X. Zhang, M. Han, W. Zhang, J. Xie, Q. Li, Prediction of occupancy level and energy consumption in office building using blind system identification and neural networks, Appl. Energy 240 (2019) 276–294.
[23] Y. Guo, J. Wang, H. Chen, G. Li, J. Liu, C. Xu, R. Huang, Y. Huang, Machine learning-based thermal response time ahead energy demand prediction for building heating systems, Appl. Energy 221 (2018) 16–27.
[24] Chou J-S, Bui D-K, Modeling heating and cooling loads by artificial intelligence for energy-efficient building design. Energy Build, 82 (2014) 443–46. doi: 10.1016/j.enbuild.2014.07.042.
[25] C. Fan, F. Xiao, S. Wang, Development of prediction models for next-day building energy consumption and peak power demand using data mining techniques, Appl. Energy 127 (2014) 1–10, https://doi.org/10.1016/j.apenergy.2014.04.016.
[26] R.E. Edwards, J. New, L.E. Parker, Predicting future hourly residential electrical consumption: a machine learning case study, Energy Build. 49 (2012) 591–603, https://doi.org/10.1016/j.enbuild.2012.03.010Get, rights and content.
[27] J. Massana, C. Pous, L. Burgas, J. Melendez, J. Colomer, Short-term load forecasting in a non-residential building contrasting models and attributes, Energy Build. 92 (2015) 382–390, https://doi.org/10.1016/j.enbuild.2015.02.007.
[28] Y. Li, Z. Tong, S. Tong, D. Westerdahl, A data-driven interval forecasting model for building energy prediction using attention-based LSTM and fuzzy information granulation, Sustain. Cities Soc. 76 (2022) 103481.
[29] Q. Li, Q. Meng, J. Cai, H. Yoshino, A. Mochida, Applying support vector machine to predict hourly cooling load in the building, Appl Energy 86 (2009) 2249–2256, https://doi.org/10.1016/j.apenergy.2008.11.041.
[30] J. Guo, S. Yun, Y. Meng, N. He, D. Ye, Z. Zhao, L. Jia, L. Yang, Prediction of heating and cooling loads based on light gradient boosting machine algorithms, Build. Environ. 236 (2023), 110252, https://doi.org/10.1016/j.buildenv.2023.110252.
[31] C. Miller, P. Arjunan, A. Kathirgamanathan, C. Fu, J. Roth, J.Y. Park, C. Balbach, K. Gowri, Z. Nagy, A.D. Fontanini, J. Haberl, The ASHRAE great energy predictor iii competition: overview and results, Sci. Technol. Built Environ. 26 (10) (2020) 1427–1447.
[32] Prokhorenkova, L., G. Gusev, A. Vorobev, A. V. Dorogush, A.Gulin, Cat-Boost: Unbiased boosting with categorical features, Advances in neural information processing systems (2019) 6638-3348. doi: 10.48550/arXiv.1706.09516.
[33] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, Tie-Yan Liu, LightGBM: a highly efficient gradient boosting decision tree, Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17) (2017) 3149–3157. https://dl.acm.org/doi/10.5555/3294996.3295074.
[34] Vaswani AS, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., Polosukhin, I, Attention is all you need, In Advances in Neural Information Processing Systems (2017) 5998–6008. doi: 10.48550/arXiv.1706.03762.
[35] Y. Li, J. Li, M. Zhang, Deep Transformer modeling via grouping skip connection for neural machine translation, Knowl.-Based Syst. 234 (2021), 107556, https://doi.org/10.1016/j.knosys. 2021.107556.
[36] Z.-M. Chen, Q. Cui, B. Zhao, R. Song, X. Zhang, O. Yoshie, SST: spatial and semantic transformers for multi-label image recognition, Trans. Image Processing 31 (2022) 2570–2583.
[37] Li M, Zhang S, C. Zorilă, R. Doddipatla, Transformer-Based Streaming ASR with Cumulative Attention, International Conference on Acoustics, Speech and Signal Processing (ICASSP) (2022) 8272-8276. doi: 10.1109/ICASSP43922.2022.9746693.
[38] B. Lim, S.Ö. Arık, N. Loeff, T. Pfister, Temporal fusion transformers for interpretable multi-horizon time series forecasting, Int. J. Forecast. 37 (4) (2021) 1748–1764.
[39] LI S, Jin X, Yao X, Zhou X, Chen W, Wang Y, Yan X, Enhancing the Locality and Breaking the Memory Bottleneck of Transformer on Time Series Forecasting, 33rd Conference on Neural Information Processing Systems. Vancouver, Canada. (2019) 1-11. doi: 10.48550/arXiv.1907.00235.
[40] H. Wu, G.Q. Shen, X. Lin, M. Li, C.Z. Li, A transformer-based deep learning model for recognizing communication-oriented entities from patents of ICT in construction, Autom. Constr. 125 (2021) 103608.
[41] Chen T, Carlos, G, XGBoost: A scalable tree boosting system, In: ACM SIGKDD. (2016) 785–94. doi: 10.1145/2945672.2945785.
[42] Natekin A, Knoll A, Gradient boosting machines, a tutorial, Front Neurorobotics. (2013) 7. doi: 10.4049/fnbot.2013.00021.
[43] A. Graves, J. Schmidhuber, Framewise phoneme classification with bidirectional LSTM and other neural network architectures, Neural Netw. 18 (2005) 602–610, https://doi.org/10.1016/j.neunet.2005.06.042.
[44] F. Dong, J. Yu, W. Quan, Y. Xiang, X. Li, F. Sun, Short-term building cooling load prediction model based on DwdAdam-ILSTM algorithm: a case study of a commercial building, Energy Build. 272 (2022), 112337, https://doi.org/10.1016/j.enbuild.2022.112337.
[45] Lin S, Combining a building integrated PVT system with a low temperature desiccant cooler to drive affordable solar cooling photovoltaics and renewable energy engineering, Faculty of Engineering, UNSW, https://wwwunsworksunsweduau/primo-explore/fulldisplay?docid=unsworks_54659&context=L&vid=UNSWORKS&lang=en_US&search_scope=unsworks_search_scope&adaptor=Local%20Search%20Engine&tab=default_tab&query=any [accessed 7 September 2020].
[46] Badr W. Why feature correlation matters … a lot!, https://towardsdatascience.com/why-feature-correlation-matters-a-lot-847e8ba445c4 [accessed 7 September 2020].
[47] Y. Liu, Y. Mu, K. Chen, Y. Li, J. Guo, Daily activity feature selection in smart homes based on Pearson correlation coefficient, Neural Process. Lett. 51 (2020) 1771–1787, https://doi.org/10.1007/s11063-019-10185-8.
[48] C. Cui, T. Wu, M. Hu, J.D. Weir, X. Li, Short-term building energy model recommendation system: a meta-learning approach, Appl. Energy 172 (2016) 251–263, https://doi.org/10.1016/j.apenergy.2016.03.112.
[49] Scikit-learn, scikit-learn: API reference, https://scikit-learn.org/stable/modules/classes.html [accessed 7 September 2020].
[50] XGBoost, XGBoost Documentation, https://xgboost.readthedocs.io/en/latest/ [accessed 7 September 2020].
[51] Keras, Keras: API reference, https://keras.io/api/ [accessed 7 September 2020].
[52] Computer Vision for Dummies, The curse of dimensionality in classification, https://www.visiondummy.com/2014/04/curse-dimensionality-affect-classification/ [accessed 7 September 2020].