https://doi.org/10.1016/j.apenergy.2017.03.064 This is the Pre-Published Version.

A Short-term Building Cooling Load Prediction Method Using Deep Learning Algorithms

Cheng Fan[a], Fu Xiao[b,*], Yang Zhao[c]

[a]Department of Construction Management and Real Estate, Shenzhen University, Shenzhen, China

[b]Department of Building Services Engineering, The Hong Kong Polytechnic University, Kowloon, Hong Kong, China

[c]Institute of Refrigeration and Cryogenics, Zhejiang University, Hangzhou, China

[*]E-mail: linda.xiao@polyu.edu.hk; Tel.: (852) 27664194

Abstract

Short-term building cooling load prediction is the essentialfoundation for manybuilding energy managementtasks, such as fault detection and diagnosis, demand-side management and control optimization.Conventional methods, which heavily rely on physical principles,have limited power in practiceas their performance is subjectto many physical assumptions. By contrast, data-driven methods have gained hugeinterests due to their flexibility in model development and the rich dataavailable in modern buildings. The rapid development in data sciencehas provided advanced data analytics to tackle prediction problems in a more convenient, efficient and effective way.

This paper investigates the potential of one of the most promisingtechniquesin advanced data analytics, i.e., deep learning, in predicting 24-hour ahead building

cooling load profiles.Deep learning refers to a collection of machine learning algorithms whichare powerful in revealing nonlinear and complex patterns inbig data. Deep learning can be used either in a supervised manner todevelop prediction models with given inputs and output (i.e., cooling load), or in an unsupervised manner to extract meaningfulfeatures from raw data as model inputs. This study exploits the potential of deep learning in both manners, and compares its performance incooling load predictionwith typical feature extraction methods and popular prediction techniques in the building field. The results show that deep learning can enhance the performance of building cooling load prediction, especially when used in an unsupervised manner for constructing high-level features as model inputs.Using the features extracted by unsupervised deep learning as inputs for cooling load prediction can evidently enhance the prediction performance. The findings are enlightening and could bring more flexible and effective solutions for building energy predictions.

Keywords: Building cooling load; Building energy prediction;Deep learning; Data mining; Big data.

1. Introduction

The building sector has become the largest energy consumer worldwide, accounting for 32% of global final energy consumption and one third of the Green House Gas emissions [1].Compared to the transportation and industry sectors, the energy saving potential in buildings is much more significant and could reach 30-80% using currently available building technologies [2].Among various building services

systems, the Heating, Ventilation and Air-Conditioning (HVAC) system is responsible for the largest proportion of building energy consumption (e.g., around 50% in U.S.) and has the largest energy saving potential (e.g., 15-30% for commercial buildings) [3, 4]. As a result, the current energy conservation measures in building operations mainly focus onthe HVAC system. Reliable prediction of short-term (i.e., with a prediction horizon of shorterthan 1-week) cooling load profile istheessentialfoundation for manybuilding energy managementtasks[4, 10], includingoptimal controland fault detection and diagnosis (FDD) strategies [5-7]. Ben-Nakhi and Mahmoud adopted artificial neural networks to predict next-day cooling load for optimizing the HVAC thermal energy storage system operation [6].It was shown that optimal control strategies canincrease the operating flexibilities while reducing the operating costs. Lu et al. utilizedartificial intelligence for building cooling load predictions with the aim ofoptimizing HVAC system operations [7]. Energy-efficient operations were achieved by optimizing the set points of chilled water supply temperature, chilled water pump head andsupply air pressure in duct networks. Shan et al. developed a robust chiller sequencing control strategy relying onbuilding cooling load predictions [8]. The strategy was validatedand could achieve 3% energy saving compared to conventional strategies. Predicted cooling load has been used either directly or indirectly as an indicator for FDD. As examples, previous studies have used cooling load for detecting and diagnosing the low delta-T syndrome in chilling system [9], reducing energy consumption in air-handling units [10], and detectingabnormal energy use at the building-level [11].Building cooling load

prediction is also critical to building demand-side management. A large number of studies have been carried out to investigate the most cost-effective demand response measures (e.g., load shifting) considering the interactions between buildings and smart-grids [5]. An essential assumption of thesestudies is that reliable predictions of short-term building cooling load profiles are available to use.

Existing methodsfor short-term cooling load prediction can generally be classified into two types, i.e., physical-modelbasedmethods and data-driven methods. Physical-model based methodsrely onphysical principles and detailed information on building and itssystems to characterizebuilding thermal behaviors. The models developed are usually referred as white-box models.Admittedly, they cancapture the actual building thermal response to various influential factors, such as outdoor and indoor environment. However, it requires a large amount of detailed building information (e.g., information on building envelop and the selection of building equipment) and the model performance may not be consistent if assumptions of physical principles are not fulfilled [12].

The other type of prediction methods, i.e., data-driven methods, mainly relies on building operationaldata to discover the relationship betweenbuilding cooling load and relevantvariables (e.g., the outdoor temperature and relative humidity, and indoor occupancy). The models developed in such a mannerare known as either grey-box or black-box models[13, 14]. The main advantage of data-driven models, especially black-box models, is that the modeling process is more efficient and flexible.The use of advanced data analytics, such as machine learning and artificial

intelligence,enables data-driven models to achieve high accuracyanddiscover potentially useful yet previously unknown relationships with efficient computation. The performance of data-driven methods is mainly affected by two factors, i.e., the prediction techniques used for model developmentand the features used as model inputs. Previous research showed that the prediction techniques from the field of machine learning and artificial intelligence, such as support vector regression [16, 17] and artificial neural networks [18, 19], worked very well in building energyprediction. Various studies have also shown that nonlinear techniquescould achieve more accurate results compared with linear ones, e.g., multiple linear regression and autoregressive moving average [20, 21]. Regarding to the model inputs, previous studies mainly relied on engineering knowledge or simple statistical methods (e.g., correlation coefficient)to select model inputs or develop features as model inputs. For instance, engineering knowledge tells that the building cooling load is closely related to the outdoor weather condition and indoor occupancy. Therefore, outdoor dry-bulb temperature, relative humidity and solar irradiationas well as the indoor occupancy schedule (e.g.,*Day of the week*, *Hour* and *Minute*) were typically selected as model inputs [15, 22].Some studies also used historical data as model inputs considering the building thermal capacity[20, 23]. Using original historical data, such the outdoor temperature and humidityat previous time steps,as model inputs is generally not recommended, as it may substantially increase the number of model inputs, making prediction models more complicated and computationally expensive. Feature extraction, which transforms raw data intoa compact yet information-preserving form,

can beappliedto develop features as model inputs.Three types of feature extraction methods have been found in previous studies, i.e., engineering, statistical and structural feature extraction [22-27]. Engineering features are constructed based on engineering knowledge and experience, e.g., using the data at previous one-hour as model inputs [23]. Statistical features are constructedusingsummarizing statistics, e.g., minimum,maximum and mean values of the measurementsover a period of time [22, 24]. Structural features represent the structural or temporal relationships within the data over a period of time, e.g., the cut-off lag of autocorrelation function or the dominant frequencies in the time-series data [25-27].

The data-driven approacheshavegained increasing popularityin the building field, as more and more building operational data are available in modern Building Automation System (BAS). The rapid development in big data analytics offers opportunities for the effective use of big BAS data. One prominent and promising example is deep learning, which has gained huge success in the field of pattern recognition [28, 29]. Deep learning refers to a collection of machine learning algorithms which adopts a ‘deep’ model architecture for knowledge discovery. In other words, the input data will be transformed in either a linear or a nonlinear manner multiple times before deriving the output. By contrast, conventional machine learning algorithms are ‘shallow’ and input data only undergo one or two rounds of transformation. Deep learning can be used either in a supervised mannerfordeveloping a prediction model or in an unsupervised manner for extracting meaningful features from raw data. The formerworks on two clearly defined data setsasthe input set

(denoted as *X*) and the output set (denoted as *Y*), while the latter works on the input data set (*X)*alone and aims to extract high-level abstractions of *X*. Deep learning has demonstrated its power in various applications, such as speech recognition and visual object detection [28, 29]; however, it's potential in building cooling load prediction is still unknown.To fill this research gap, this study systematically investigatesthe potential of deep learning in building cooling load predictionand detailed comparisons with existing analyticsare given.

The paper is organized as follows: Section 2 presents the research outline and a brief introduction of data analytics used.Section 3 describes the modeling process using the data retrieved from an educational building. The performance in terms of prediction accuracy and computation loadis compared and discussed in Section 4. Conclusions are drawn in Section 5.

2. Research Methodology

2.1 Research outline

Fig. 1 presents the general research outline. Feature extraction is firstly carried out to extract meaningfulfeatures as model inputs. Fourtypes of feature extraction methods highlightingthe unsupervised deep learning areused. Sevenprediction techniques featuring the supervised deep learning are adopted to develop prediction models based on different feature sets. The performance in terms of prediction accuracy and computation load is compared and discussed.This research is valuable for building professionals who need the 24-hour ahead building cooling load profiles for

evaluating building energy performance, developing online control, optimization and fault diagnosis strategies, developing operation strategies toward smart grid, and etc.It is impossible and unnecessary to introduce all the feature extraction and data analytic methods in detail here, as most of the methods have been studied in existing literature. These methods are used to compare with the deep learning method.The following subsections presents an overview onthe feature extraction methods and prediction techniques used in this study. A brief introduction on deep learning is provided in a late part.

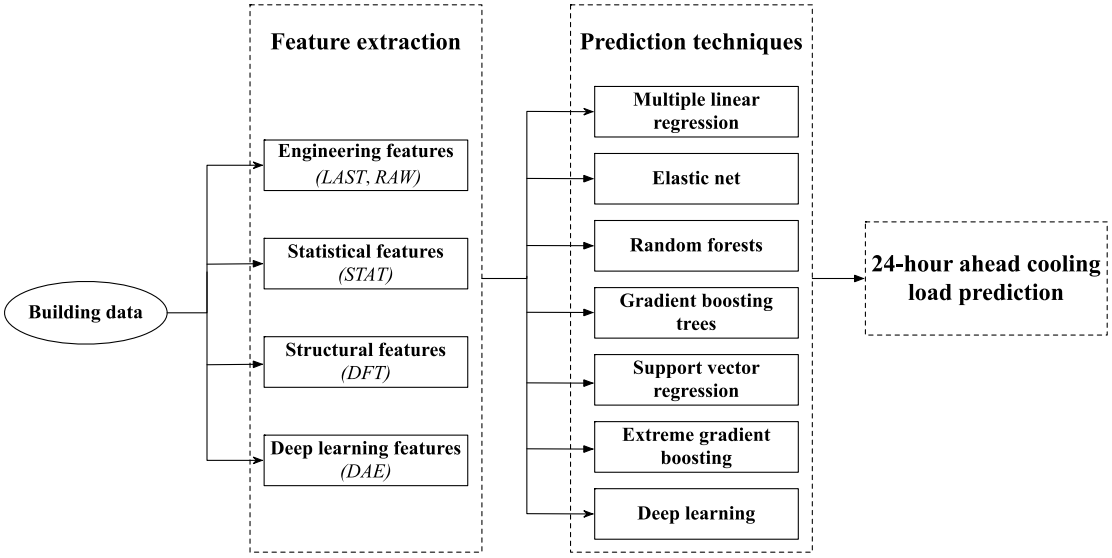

Fig. 1 Research outline

2.2 Feature extraction

The selection of the model inputs is usually the first yet the most important step in developing reliableprediction models. Feature extraction has been widely used to obtain useful and representative information from raw data as model inputs. The intuition behind is twofold. Firstly, taking the historical data in their original form

might result in a large number of model inputs and the information contained could be redundant.As a result, the risk of over-fitting is increased dramatically. Secondly, feature extraction helps to reduce the dimensionality of model inputs and thereby, reducing the computation load in model development.This paper examinesthe performance of four feature extraction methods. Three of them, i.e., engineering, statistical and structural feature extraction methods, have been used in previous studies and are selected for comparison purposes.

The engineering methods mainly rely on engineering expertise to select model inputs. A typical engineering method is to select the *k* most recent historical data as model inputs for building cooling load predictions. Considering that building operation presents great daily seasonality, this research adopts two engineering feature sets for comparison: the first consists of historical measurementsin previous 24-hours and the second only consider measurements in previoushour.

Statistical feature extraction methods calculate the summarizing statistics of a time series as features. This study selects four of the most commonly used summarizing statistics as statistical features, i.e., the minimum, maximum, mean and standard deviation of a time series.

Structural features represent the structural information of a time series. One popular approach is to transform the time series from time domain to frequency domain, and then exploring the data in frequency domain for feature extraction. In this study, the top-*k* dominant frequencies of a time series are extracted as structural features.

The fourth feature extraction method is the main focus of this study: unsupervised

deep learning modelsaredeveloped to extract high-level data abstractions as features. These features obtainedare in essence nonlinear combinations of raw input data. They are extracted in a purely data-driven fashion and can hardly be formulatedusingdomain knowledge.

This study adopts an iterative approach for predicting the 24-hour ahead building cooling loads. It requires the features to be updated along with the prediction process. The general idea is illustrated in Fig. 2. At the first iteration, feature extraction is performed based on the historical values in the past 24-hour. The features extracted are then fed to the prediction model to generate the predicted value at the next time step. At the second iteration, this predicted value is combined with part of the historical values to form the new inputs. The following feature extraction process will update the features, based on which new prediction is made. The process stops till all predictions of the next 24-hour are generated.

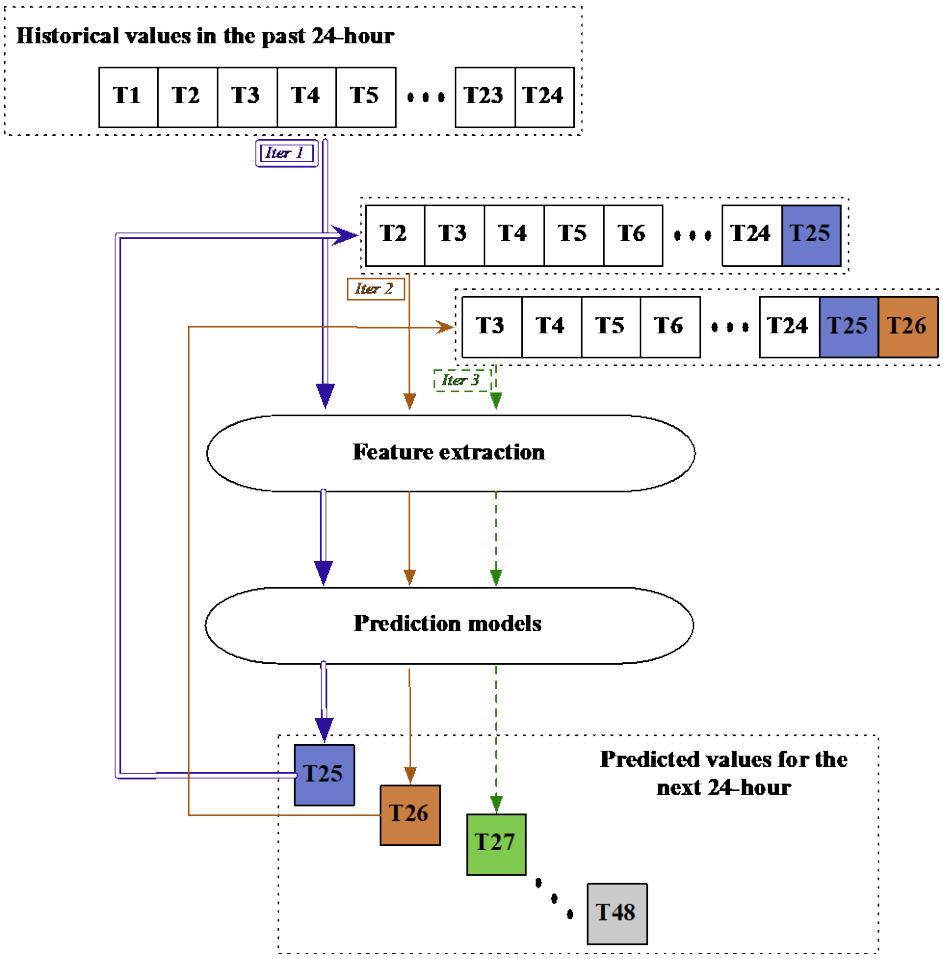

Fig. 2 Iterative process for 24-hour ahead building cooling load prediction

2.3 Prediction techniques

Seventypical prediction techniques are selected to develop prediction models, including multiple linear regression (MLR), elastic net (ELN), random forests (RF), gradient boosting machines (GBM), support vector regression (SVR), extreme gradient boosting trees (XGB) and the most commonly used architecture of supervised deep learning, i.e.,deep neural network (DNN). The first six techniques are selected based on their popularity in previous studiesand used to compare the performance of supervised deep learning.

The first two, i.e., MLR and ELN, are linear techniques and served as performance benchmark in this study. ELNis an extended version of MLR, with extra regularization termsintroduced to perform variable selection and weaken the influence of multicollinearity. It is expected to have better generalization performance than MLR when the number of inputs is large. Developing models based on these two linear techniques are computationally efficient and the resulting models are easy to interpret. Nevertheless, the prediction accuracy could be poor due to their incompetence ofmodelingnonlinearity.

The other predictiontechniques used are capable of capturing complex and nonlinear relationships. SVR, RFand GBM are powerful machine learning methods which have been successfully used for building energy predictions in previous studies [12, 34]. XGBisan improved version of gradient boosting with higher computation efficiency and better capabilities in tacklingthe over-fitting problem [31]. This study also investigates the potential of deep learningas a prediction technique in predicting 24-hour ahead building cooling load [29]. The resulting model is denoted as DNN in this study. It shares a similar representation with conventional neural networks, but with a more complex architecture and training schemes.

2.4 Deep Learning

As defined by LeCun et al., deep learning is a technique which allows computational models with multiple processing layers to learn representations of data with multiple levels of abstractions [29]. Oneway to understand deep learning models is to compare

them to artificial neural networks, but with more complex architecturesand training schemes.A typicalneural network usually has three layers, i.e., one input layer, one hidden layer and one output layer. By contrast, deep learning models could have a number of hidden layers and each layer may have different functions, e.g., performing nonlinear transformations or convolution operations. The training scheme is also different from training conventional neural networks. New model parameters, such as the dropout which specifies the proportion of neurons to be randomly ignored during model training, are introduced to ensure the robustness of deep learning models.

Deep learning can be used either in a supervised manner to develop a deep neural network model (DNN) for prediction or in an unsupervised manner to develop a deep auto-encoder model for feature extraction. The latter is of special interests as it tackles the most intrinsiclimitation of conventional machine learning techniques, i.e., the lack of ability to process data in their raw form [29]. In other words, to ensure the success of conventional machine learning techniques, great effort is needed to transform the raw data into meaningful features.Unsupervised deep learning has been widely and successfully used to perform the task of feature extraction in other fields [30]. By setting the output as the same as the input, a deep auto-encoder model tries to reconstruct the input by minimizing the reconstruction error. A common practice is to design the auto-encoder with a symmetric structure and the number of neurons at each layer is decreasing towards the middle hidden layer. The activations obtained at the middle hidden layer are taken as features, which in essence are nonlinear combinations of the raw input data. Compared to the three feature extraction methods

introduced in section 2.2, using unsupervised deep learning requires little domain expertise and is very promising to extract the best information-preserving features.

3. Case Study

3.1 Data description

The data to be analyzedin this research are retrieved from an educationalbuilding in Hong Kong. The building mainly consists of offices, classrooms and a computer data center. The gross floor area is around 11,000m^2 and 8,500m^2 are air-conditioned. One-year data in 2015 are collected with a collection interval of 30-minute. The variables included in this dataset contains five time variables (i.e., *Month*, *Day*, *Hour*, *Minute* and *Day type*), the outdoor temperature, the outdoor relative humidity, the supply and return chilled water temperature and the flow rate of the chilled water temperature. The building cooling load is calculated based on the latter three variables. In total, the dataset contains 15,792 observations. Table-1 presents the summary of numeric variables in the dataset.

Table-1 Summary on numeric variables in the dataset

Variables	Min	Mean	Median	Max
Outdoor temperature (ºC)	12.01	25.59	27.28	35.79
Outdoor RH (%)	20.35	82.10	83.96	98.41
Chilled water supply	6.25	8.87	8.71	23.92

temperature(°C)				
Chilled water return temperature(°C)	6.82	11.03	10.67	24.04
Chilled water flowrate(l/s)	0.33	34.81	29.05	168.80
Cooling Load (kW)	0.0	653.3	363.7	2474.0

3.2 Construction of feature setsas model inputs

Building cooling load is heavily influenced by two factors, i.e., building occupancyand outdoor condition. While the information of building occupancy is seldom available for direct use, it is possible to take into account theoccupancy influenceusing time variables, as the occupancy schedule for a specific functional building is usuallyfixed and correlated with time. Outdoor conditions can be well-described using outdoor dry-bulb temperature, outdoor relative humidity, wind direction and speed, outdoor luminance and etc. The outdoor temperature and relative humidity are the most correlated variables to building cooling load and they are readily available in modernbuilding automation systems. Therefore, the *BASIC* feature set contains all the five time variables(i.e., *Month*, *Day*, *Hour*, *Minute* and *Day type*), the outdoor temperature and the outdoor relative humidity at time *T*. These seven features are taken as model inputs to predict building cooling load at time *T*.

Considering that building cooling load presents daily seasonality, it is expected that the prediction accuracy could be improved usingmeasurements in previous 24-hour[22]. Compared to the *BASIC* feature set, additional information of building

cooling load, outdoor temperature and RH during the past 24-hour are added for analysis, either in their raw form or after feature extraction. If without feature extraction, each time series of building cooling load, outdoor temperature and outdoor RH during the past 24-hour will result in 48 more variables (due to a collection interval of30-minute). The resulting feature set therefore contains 151 (i.e., 144+7) variables and is denoted as the *RAW* feature dataset.

As introduced in section 2.2, four feature extraction methods are adopted in this study for comparison purposes. Feature extraction is performed onthe measurements in past 24-hours. More specifically, engineering expertise tells us that the measurements at previous time step (denoted as *T-1*) may have the largest impact to the building cooling load at time *T* [23]. Therefore, three more features representing the cooling load, outdoor temperature and RH at time *T-1* are added to the *BASIC* feature set to form a new feature set, denoted as *LAST*. Statistical features are extracted by calculating the minimum, maximum, mean and standard deviation of measurements in previous 24-hours. The resulting feature set is denoted as *STAT* and has 12 more features (four features for each of the three time series of cooling load, outdoor temperature and RH) compared to the *BASIC* feature set. The discrete Fourier transformationmethod is used to extract the structural features in past 24-hour measurements. After transforming the time series from time-domain to frequency-domain, the four most dominant frequencies are selected as features for each time series. The number of dominant frequencies is selected as four to make it a fair comparison with statistical features. Similar to the *STAT* feature set, the resulting

feature set, denoted as *DFT*, has 12 more features than the *BASIC* feature set.

Another feature set is constructed using unsupervised deep learning. A deep auto-encodermodel is developed for each of the three time series. An optimization process is performed to determine the optimal model architecture. As shown in Fig. 3, the deep auto-encoder model developed has a symmetric structure with five layers in total. The model uses a *tanh* activation function, i.e., $tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$. The first and last layers both have 48 neurons to represent the measurements in previous 24-hour. The middle layer (i.e., hidden layer 2) containsthe features extracted and the number of neurons is set as four in accordance with above-mentioned feature extraction methods. The optimized neuron number in the other two hidden layers is setas 25. The resulting feature set is denoted as *DAE*.

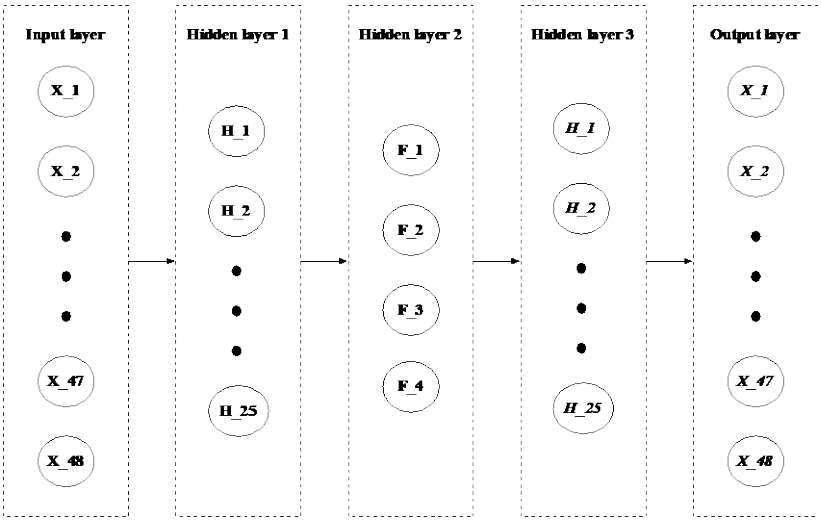

Fig. 3 Schematic of the deep auto-encoder model for feature extraction

3.3 Development of prediction models

The entiredataset is divided into training, validationand testing data with proportions of 70%, 15% and 15% respectively. The model parameters of each algorithm are optimized through cross-validation and parameter grid search. The parameters to be optimized for each prediction technique are briefly introduced as below.

Two parameters, i.e., *alpha* and *lambda*, are optimized for ELN models. The*alpha* ranges from zero to one and defines the combination of the Lasso and the Ridge regression methods. The *lambda* specifies the penalty strength for regularization terms and well tuning this parameter helps to avoid over-fitting. In this study, *alpha* is optimized over a sequence from zero to one with a decimal increment of 0.1, while lambda is optimized over a sequence from zero to one with a decimal increment of 0.01.

The support vector regression with a Gaussian radial basis function kernel is used in this study. Optimization is performed considering the complexity parameter *C* and smoothing parameter *sigma*. In general, a larger *C* tends to make the model more prone to over-fitting while a smaller*C* is more likely to cause under-fitting. The parameter *sigma* controls the shape of decision boundary. A larger *sigma* makes the decision boundary more flexible and smooth, while a smaller *sigma* makes the decision boundary more complicated and sharp.The candidate values of these two parameters take a form of 2^x, and *x* are integers ranging from 1 to 15 for *C* and -10 to 0 for *sigma*.

Random forests, gradient boosting trees and extreme gradient boosting are alldecision tree-based techniques. They all share two common model parameters, i.e., the number

of trees to grow and the depth of each individual tree. In general, the prediction accuracy and computation load both increase with the number of trees to grow. To make a fair comparison while guarantee the model performance, the total tree number is fixed as 1000 and the depth of each individual tree is set as four. The number of variables selected as candidates for tree splitting is another parameter needs to be optimized for random forests models.It is taken as $\frac{No.\ \ ofinputs}{3}$as recommended in [39]. The learning rate, which specifies how quickly a tree model adapts to the errors in previous iteration, is optimized for the latter two tree-based methods. Its candidate values range from 0.025 to 0.5 with a decimal increment of 0.025.

Compared to conventional neural networks, deep learning models have more hidden layers and the training scheme is more complicated. Four parameters of deep learning models are optimized in this study: (1) the number of hidden layers (ranging from 1 to 10 with an increment of 1); (2) the dropout ratio at the input layer (ranging from 0 to 0.3 with a decimal increment of 0.025); (3) the dropout ratio at the hidden layer (ranging from 0 to 0.3 with a decimal increment of 0.025); and (4) the activation function (including *Tanh*, *ReLU* and *sigmoid*). The dropout ratio at the input and hidden layers specifies the proportion of neurons to be randomly ignored during model training. It is a similar approach to bootstrap aggregation and has been proved to be an effective way to avoid over-fitting [32]. It should be mentioned that the number of neurons at each hidden layer is set as constant in this study to reduce the computation loadassociated with parameter optimization. The value is determined according to one of the rule of thumbs in neural network design, i.e.,

$\frac{No.\ of inputs + No.\ of outputs}{2}$ [33]. Further performance improvementmight be achievable if this parameter is optimized;however, the resulting computation load can be overwhelming.

The model optimization results are summarized in Table-2.One interesting finding is that the number of hidden layers optimized for DNN models is only 2, which means the optimal DNN model developed for building cooling load prediction does not actually need a ‘deep’ architecture. It indicates that given this dataset and this prediction problem, supervised deep learning exhibits limited advantages when compared to conventional artificial neural networks with a ‘shallow’ architecture. The potential reason behind is two-fold. Firstly, the increase in the number of hidden layers leads to a dramatic increase in the number ofmodel coefficients. To develop robust and reliable estimations of these model coefficients, ahuge amount of data is needed. This study adopts one-year data with a collection interval of 30-minute for analysis. The data amount may not be large enough to guaranteethe reliability of a ‘deeper’ model. Secondly, a neural network of two hidden layers was claimed to be sufficient to represent most of the functions in real world [40]. It is possible that two hidden layers are good enough tosolve the cooling load prediction problem.

Table-2 Parameter optimization results

Method	**Parameters**	**BASIC**	**RAW**	**LAST**	**STAT**	**DFT**	**DAE**
ELN	Alpha	0.20	0.90	0.70	0.40	0.30	0.30

	Lambda	0.19	0.02	0.19	0.04	0.10	0.19
GBM	Learning rate	0.05	0.01	0.01	0.05	0.05	0.05
SVR	*C*	8192	32768	2048	4096	4096	4096
	Sigma	0.125	0.008	0.063	0.125	0.063	0.063
XGB	Learning rate	0.1	0.1	0.1	0.1	0.1	0.1
DNN	Hidden layer	2	2	2	2	2	2
	Input dropout	0	0.05	0	0	0.05	0
	Hidden dropout	0.05	0	0	0.05	0	0
	Activation	ReLU	ReLU	ReLU	ReLU	ReLU	ReLU

4. Results and discussions

4.1 Predictionperformance

The prediction performance is evaluated using threemetrics (as defined in Equations 1 to 3, where y_k and $\widehat{y_k}$ are the actual and predicted values at time *k* respectively), includingthe mean absolute error (*MAE*), the root mean squared error (*RMSE*) and the coefficient of variation of the root mean squared error (*CV-RMSE*). *MAE* and *RMSE* are scale-dependent while *CV-RMSE* is scale-independent. Scale-dependent metrics provide readers a straightforward way to quantify the prediction error while scale-independent metrics are good for performance evaluation with othersimilarstudies. It is noted that some of the scale-independentmetrics, such as the mean absolute percentage error (*MAPE*), are not suitable for model evaluation, as theequation denominator (i.e., actual cooling loads) may be zero. By contrast,

CV-RMSE can overcome this problem, as the equation denominator is the mean of actual cooling loads over a time period.In addition, previous studies and guidelines have provided some benchmarks for model evaluation using *CV-RMSE* [35, 36]. It is specified that if the resulting *CV-RMSE* is below 30% when using hourly data, the model is calibrated and sufficiently close to physical reality for engineering purposes [37].

$$MAE = \frac{\sum_{k=1}^{n} |y_k - \widehat{y_k}|}{n} \qquad \text{(Equation 1)}$$

$$RMSE = \sqrt{\frac{\sum_{k=1}^{n} (y_k - \widehat{y_k})^2}{n}} \qquad \text{(Equation 2)}$$

$$CV - RMSE = \frac{\sqrt{\frac{\sum_{k=1}^{n} (y_k - \widehat{y_k})^2}{n}}}{\frac{\sum_{k=1}^{n} y_k}{n}} \qquad \text{(Equation 3)}$$

Table-3 summarizes the resulting *RMSE*, *CV-RMSE* and *MAE*. In terms of prediction techniques, MLR and ELN have the worst performance and the resulting *CV-RMSE*exceeds the 30% threshold most of the time. This is in accordance with expectation as these two are not capable of modelingcomplex and nonlinear relationships. Using nonlinear predictiontechniques can significantly boost the prediction accuracy and the resulting *CV-RMSE* can be well below the 30% threshold. In general, XGBmethod has a performance edge over the GBM, SVR and the DNN methods. The performance of RF is not as good as the above-mentioned four nonlinear methods. The best prediction performance is achieved when XGB models aredeveloped using the *DAE* feature set and the resulting *CV-RMSE*is 17.8%.

It is observed that linear and nonlinear techniques have their own best feature set for model development. Lineartechniques reach their best performance when the

*RAW*feature set is used, while nonlinear methods obtain the best performance using the *DAE* feature set. A possible explanation is that the features extracted by deep auto-coders are high-level nonlinear abstractions of the raw data. Therefore, nonlinear prediction methods are needed to fully release the power ofthese high-level features in predicting cooling loads.

The results show that the prediction performance can be greatly enhanced by introducing themeasurementsinthe past 24-hours in *RAW* form. However, the engineering features, statistical features and structural features extracteddo not necessarily enhance the performance when compared to using the *BASIC* feature set alone. It indicates that useful information is indeed hidden in the past 24-hours measurements, yet it is not a trivial task to discover useful information through conventional feature extraction methods. Conventional feature extractionmethods are highly dependent on domain expertise and may lead to completely different results for different buildings or different prediction techniques. By contrast, using unsupervised deep learning can almost alwaysguarantee a performance boost, especially when nonlinear prediction techniques are used. Such feature extraction method is purely data-driven and little human intervention is involved. It can be applied as a generic approach for extracting useful information in building data.

Figs. 4-10 present the predicted and actual cooling load profiles on the 14 testing days, which are randomly selected from the testing data for visualization purposes. For the ease of interpretation, only the predictions using the *BASIC*, *RAW* and *DAE* feature sets are shown and compared with the actual building cooling loads. It is evident that

the prediction performance of linear techniques is quite poor and cooling load predictions are likely to be negative whentesting days are in cold seasons of Hong Kong (i.e., testing days 1, 2, 3, 4, 5, 13 and 14 are measured in January, February, March and December). Such modeling deficiency can be generally eliminated when using nonlinear prediction techniques.

Table-3 24-hour ahead prediction accuracy on testing data

Method	**Metrics**	**BASIC**	**RAW**	**LAST**	**STAT**	**DFT**	**DAE**
MLR	RMSE	286.8	166.3	244.8	315.1	347.3	216.8
	CV-RMSE	47.9%	27.8%	40.9%	52.6%	58.0%	36.2%
	MAE	230.4	117.8	202.1	242.8	280.2	177.2
ELN	RMSE	286.5	167.9	260.0	312.3	294.5	214.3
	CV-RMSE	47.8%	28.0%	43.4%	52.1%	49.2%	35.8%
	MAE	230.4	118.9	212.0	241.8	233.3	175.8
RF	RMSE	168.7	189.0	316.4	235.1	215.2	130.9
	CV-RMSE	28.2%	31.6%	52.8%	39.3%	35.9%	21.9%
	MAE	114.0	118.4	216.6	179.2	151.5	85.6
GBM	RMSE	136.8	146.3	178.4	148.5	133.2	117.8
	CV-RMSE	22.8%	24.4%	29.8%	24.8%	22.2%	19.7%
	MAE	94.2	102.5	127.7	108.1	94.0	83.9
SVR	RMSE	143.5	137.8	153.8	132.7	197.0	113.8

	CV-RMSE	24.0%	23.0%	25.7%	22.2%	32.9%	19.0%
	MAE	109.1	98.5	109.0	98.6	153.3	85.4
XGB	RMSE	129.0	116.6	227.4	148.0	122.6	106.5
	CV-RMSE	21.5%	19.5%	38.0%	24.7%	20.5%	17.8%
	MAE	85.8	82.1	139.4	104.1	83.6	71.6
DNN	RMSE	175.7	131.4	162.5	129.6	159.3	123.5
	CV-RMSE	29.3%	21.9%	27.1%	21.6%	26.6%	20.9%
	MAE	111.9	90.2	109.4	87.6	106.0	100.5

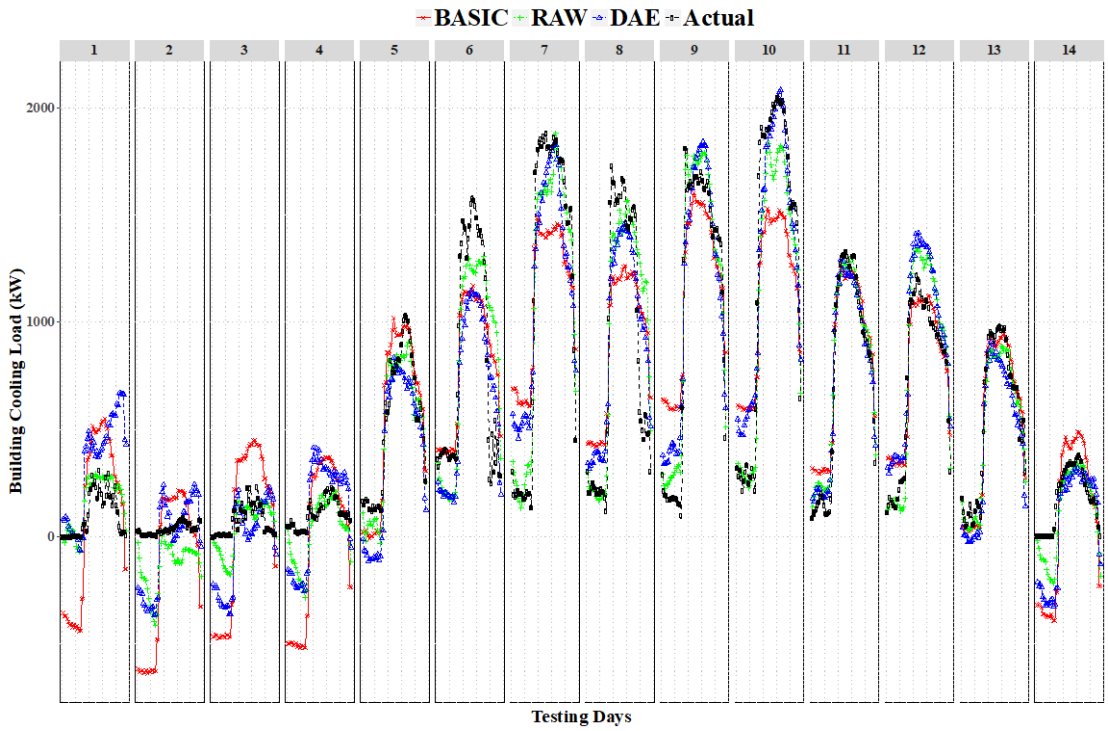

Fig. 4 Prediction performance using multiple linear regression (MLR)

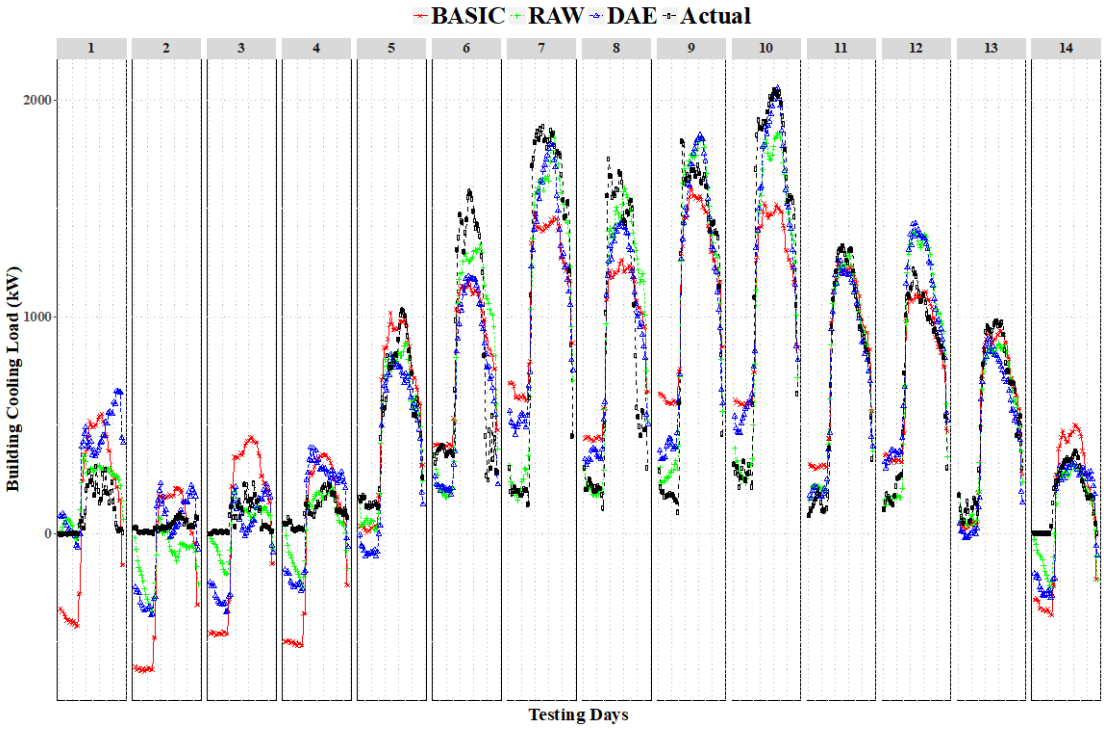

Fig. 5 Prediction performance using elastic net (ELN)

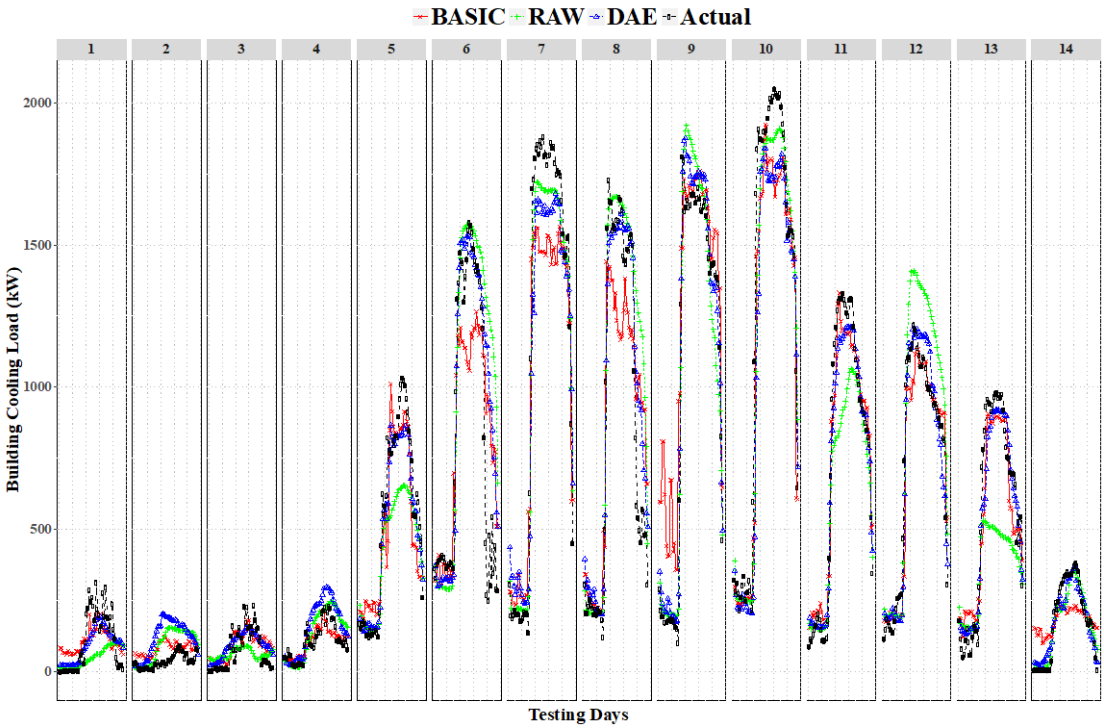

Fig. 6 Prediction performance using random forests (RF)

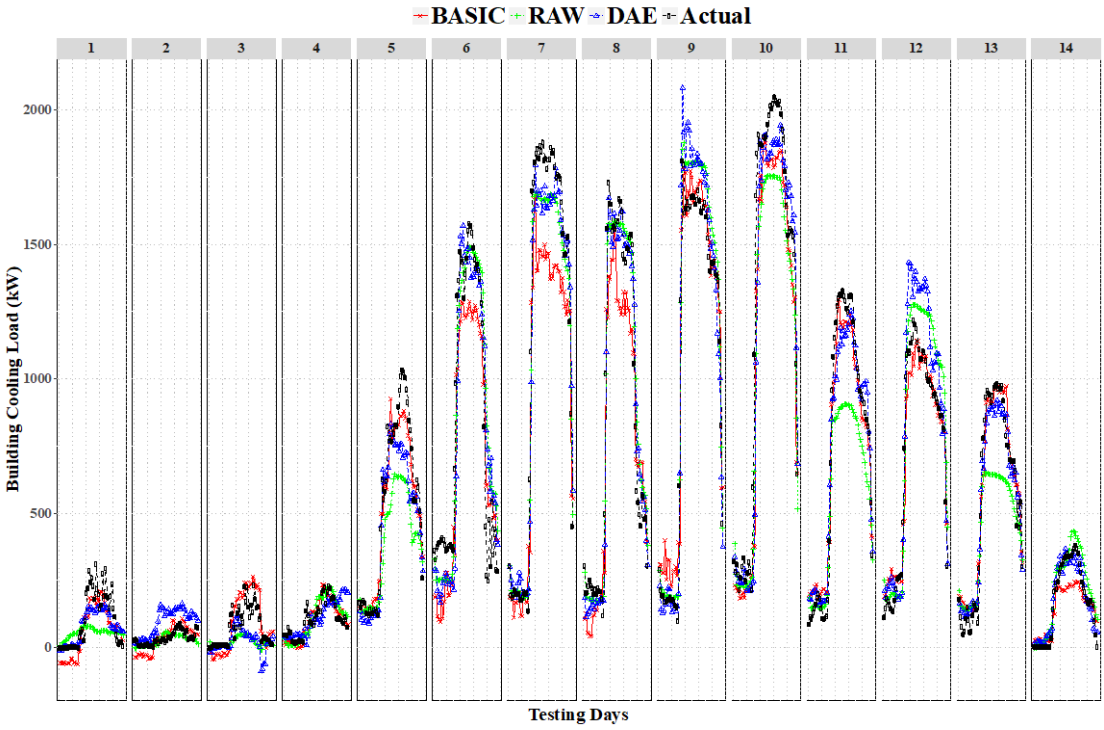

Fig. 7 Prediction performance using gradient boosting machines (GBM)

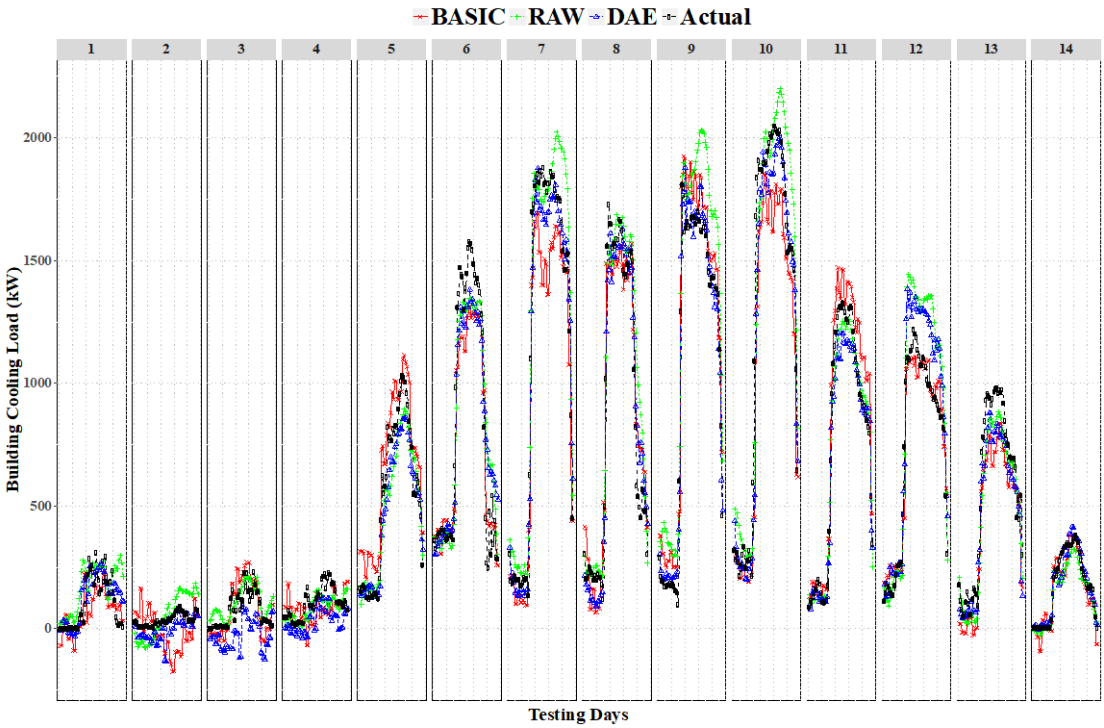

Fig. 8 Prediction performance using support vector regression (SVR)

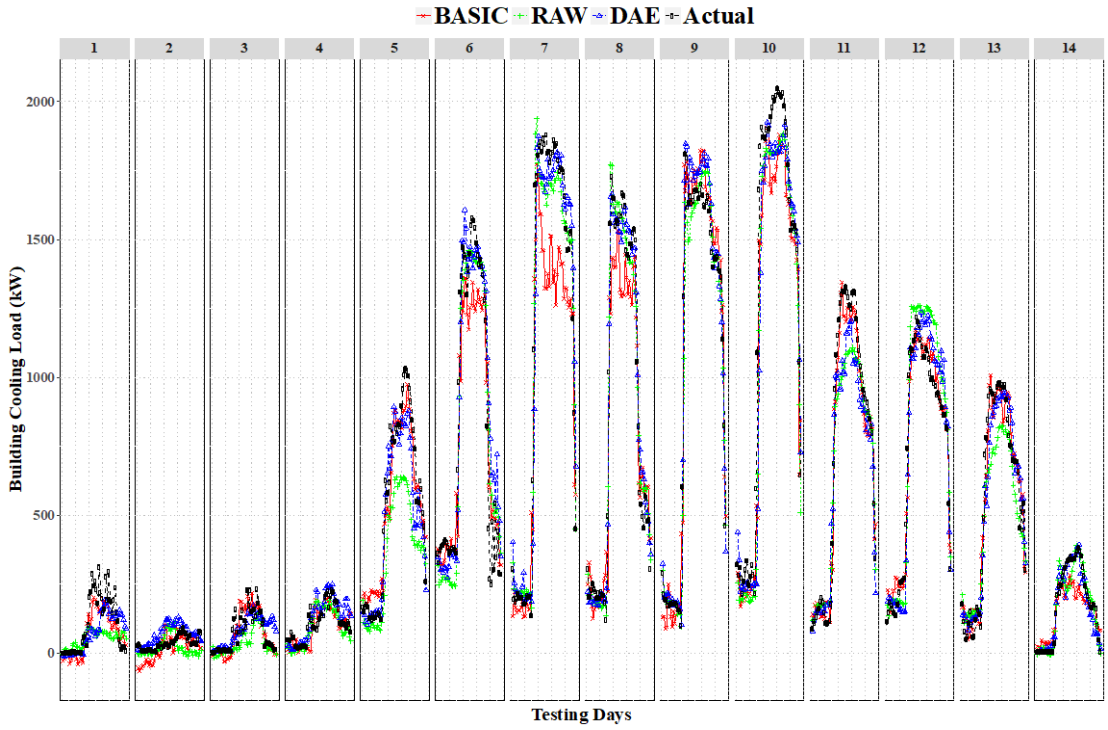

Fig. 9 Prediction performance using extreme gradient boosting (XGB)

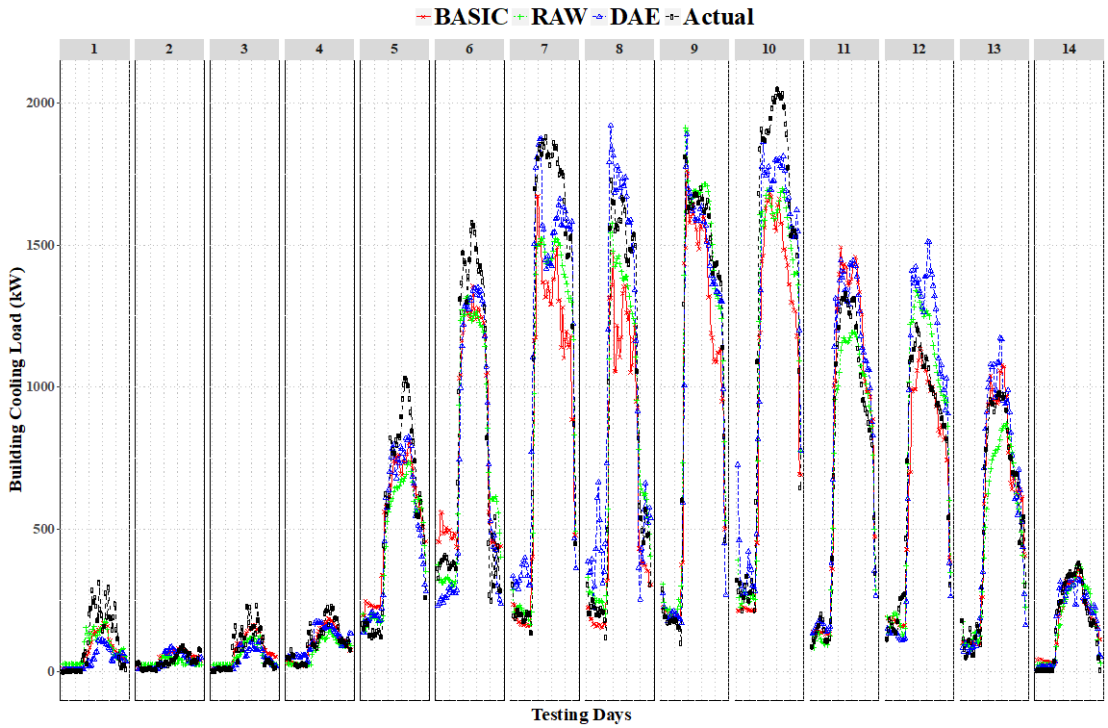

Fig. 10 Prediction performance using deep neural networks (DNN)

4.2 Computation load

The computation load is evaluated from two perspectives, i.e., (1) the computation time spent in model development using the training data set; (2) the computation time spent in generating 24-hour cooling load profilesfor 14 days. The computation tool used for this research is a MacBook Pro with a processor of 2.5GHz Intel Core i7. All the computation is performed using the open source software *R* [38].

As summarized in Table-4, linear prediction techniques account for the least computation time for model development. By contrast, nonlinear techniques spend much more time. The XGB method is the most efficient nonlinear method. Itscomputation time is at least four times faster than the other four nonlinear methods, except when the *RAW* feature set is used. The RF and DNN methods are the most computational expensive ones.

The computation time generally increases with feature numbers. As shown in Table-4, the model development time reaches maximum when the *RAW* feature set is used. It contains 151 features, which is much larger than feature numbers in other feature sets (i.e., 7 for *BASIC*, 10 for *LAST*, 19 for *STAT*, *DFT* and *DAE*). In addition, when a feature is categorical rather than numerical, the one-hot-encoding method is usually performed fordata preprocessing. In such a case, a categorical variable with *k* unique values will be transformed into *k* columns containing either 0 or 1 for value indication. This explains why the computation time using *DFT* featuresare much larger than that using *STAT* and *AE* features, even though all these three feature sets have the same feature number, i.e., 19.

Table-5 summarizes the computation time when predicting the fourteen 24-hour cooling load profiles. Except for the models using the *BASIC* feature set, all the other models adopts an iterative process to make predictions (as shown in Fig. 2), i.e., the cooling load prediction at time T is used to predict cooling load at later time. Similar to the computation time used for model development, the RF and DNN require the largest time for prediction. The models based on *DAE* features result in the largest prediction time, as deep auto-encoders spend more time in updating features.

Table-4 Computation time for model development

Time (s)	**MLR**	**ELN**	**RF**	**GBM**	**SVR**	**XGB**	**DNN**
BASIC	0.10	0.05	210.94	73.80	85.64	10.20	206.74
RAW	0.61	0.36	489.34	279.53	151.05	203.66	258.76
LAST	0.10	0.05	193.60	80.01	25.40	14.20	218.93
STAT	0.12	0.07	214.24	89.53	96.91	39.36	208.86
DFT	0.60	0.31	449.80	208.97	127.63	18.24	196.64
DAE	0.12	0.06	199.20	90.72	91.61	36.46	218.82

Table-5 Computation time for predicting 24-hour profilesfor 2-week data

Time (s)	**MLR**	**ELN**	**RF**	**GBM**	**SVR**	**XGB**	**DNN**
BASIC	0.005	0.002	0.209	0.022	0.253	0.093	0.016
RAW	8.988	3.126	33.553	3.917	21.794	3.463	13.864

LAST	5.120	5.120	35.414	1.769	8.274	1.678	11.897
STAT	4.208	1.822	35.017	2.335	9.524	2.175	11.242
DFT	8.782	2.881	35.990	3.654	21.280	3.229	11.769
DAE	221.795	221.134	232.615	227.491	226.377	236.169	229.402

5. Conclusions

This paper systematically investigates the potential of deep learning in building cooling load prediction from two perspectives, i.e., extracting meaningful features and developing prediction models.The deep learning-based method is compared with the state-of-the-art prediction techniques and existing feature extraction methods used in the building field in terms of accuracy and computation efficiency.To the best of the authors' knowledge, this is the first attempt in the building fieldto examine the potential of deep learning in building energy prediction. The research results show that deep learning-based methodsdo help to enhance the prediction performance.

In terms of prediction techniques, the research results show that nonlinear prediction techniques perform much better than the linear ones. The extreme gradient boosting (XGB) method shows its superiority in prediction when compared with others.The best prediction performance is obtained by adopting XGB models using features extracted byunsupervised deep learning models (i.e., deep auto-encoders). Supervised deep learning models do not show evident advantagesin developing cooling load prediction models. The hidden layer numberoptimized shows that supervised deep learning models do not actually need a 'deep' architecture and a 'shallow' architecture

with two hidden layers works well. The potential reasons are two-fold:(1) the data used in this study is not large enough to ensure the reliable training of a ‘deeper’ model; (2) two hidden layers are sufficient to describe the underlying patterns of building cooling load in this data set. Nevertheless, the newly developed training techniques associated with supervised deep learning, including using the dropout method for model regularization and the *ReLU* as the activation function, do help to enhance the prediction performance of neural networks.

One interesting finding of this research is that significant improvementin prediction performance can beachieved when using features extracted by unsupervised deep learning models. The features extracted can almost always lead to the best prediction performance, no matter which nonlinear prediction technique is used. By contrast, the three other conventional feature extraction methods do not always guarantee the discovery of useful features and negative influence on prediction performance can be observed.Feature extraction is one of the most challenging tasks in predictive modeling, especially when the volume of data variables is large and domain expertise is lacking. The success of using unsupervised deep learning for feature extraction provides great flexibilities in practical applications. Itenables a purely data-driven approach to extract meaningful features and therefore, greatly reduces the complexity in building energy prediction.

The deep learning-based methods proposed in this study can achieve accurate and reliable prediction of 24-hour ahead building cooling load profiles. It is of high practical value, as the 24-hour ahead building cooling load profile is the foundationfor

many building operation management tasks. It can be used to develop demand-side management (DSM) programs, optimal control strategies, as well as fault detection and diagnosis methods. As examples, with the predicted 24-hour ahead cooling load profiles, chiller sequencing control schemes can be developed to maximize the energy efficiency of chiller plants.The deviation between the observed and predicted cooling load can be used as indicators for detecting anomalies in building operations.Control strategies can be developed to regulate the demand response operations and thereby, facilitate the interactions between building operations and smart-grid. Further studies using data from various sources will be valuable for testing the proposed methods and promoting their applications in practice.

Acknowledgements

The authors gratefully acknowledge the support of this research by the Research Grants Council (RGC) of the Hong Kong SAR (152181/14E), the Shenzhen University and the Hong Kong Polytechnic University.

References

[1] Urge-Vorsatz D, Cabeza LF, Serrano S, Barreneche C, Petrichenko K. Heating and cooling energy trends and drivers in buildings. Renew SustEnerg Rev2015; 41: 85-98.

[2] United Nations Environment Program, accessed on September 9, 2016, http://www.unep.org/sbci/AboutSBCI/Background.asp

[3] Perez-Lombard L, Ortiz J, Pout C. A review on buildings energy consumption

information.Energ Buildings2008; 40: 394-8.

[4] Li XW, Wen J. Review of building energy modeling for control and operation.Renew SustEnerg Rev2014; 37: 517-37.

[5] Xue X, Wang SW, Sun YJ, Xiao F. An interactive building power demand management strategy for facilitating smart grid optimization.ApplEnerg2014; 116: 297-310.

[6] Ben-Nakhi AE, Mahmoud MA. Cooling load prediction for buildings using general regression neural networks.Energ Convers Manage2004; 45: 2127-41.

[7] Lu L, Cai WJ, Xie LH, Li SJ, Soh YC. HVAC system optimization – in-building section.Energ Buildings2005; 37: 11-22.

[8] Shan K, Wang SW, Gao DC, Xiao F. Development and validation of an effective and robust chiller sequencing control strategy using data-driven models.AutomatConstr 2016; 65: 78-85.

[9] Gao DC, Wang SW, Shan K, Yan CC. A system-level fault detection and diagnosis method for low delta-T syndrome in the complex HVAC systems.ApplEnerg2016; 164: 1028-38.

[10] Kusiak A, Li MY. Cooling output optimization of an air handling unit.ApplEnerg2010; 87: 901-9.

[11] Wang HL, Xu P, Lu X, Yuan DK. Methodology of comprehensive building energy performance diagnosis for large commercial buildings at multiple levels.ApplEnerg2016; 169: 14-27.

[12] Zhao H, Magoules F. A review on the prediction of building energy

consumption.Renew SustEnerg Rev2012; 16: 3586-92.

[13] Zhou Q, Wang SW, Xu X, Xiao F. A grey-box model of next-day buildings thermal load prediction for energy efficient control.Int J Energ Res 2008; 32: 1418-31.

[14] Braun JE, Chaturvedi N. An inverse gray-box model for transient building load prediction. HVAC&R Res 2002; 8: 73-99.

[15] Benedetti M, Cesarotti V, Introna V, Serranti J. Energy consumption control automation using artificial neural networks and adaptive algorithms: Proposal of a new methodology and case study.ApplEnerg2016; 165: 60-71.

[16] Li Q, Meng QL, Cai JJ, Yoshino H, Mochida A. Applying support vector machine to predict hourly cooling load in the building. ApplEnerg2009; 86: 2249-56.

[17] Zhao HX, Magoules F. Parallel support vector machines applied to the prediction of multiple building energy consumption. J AlgorithmComputTechnol 2010; 4: 231-49.

[18] Neto AH, Fiorelli FAS, Comparison between detailed model simulation and artificial neural network for forecasting building energy consumption. Energ Buildings 2008; 40: 2169-76.

[19] Hou ZJ, Lian ZW, Yao Y, Yuan XJ. Cooling-load prediction by the combination of rough set theory and an artificial neural-network based on data fusion techniques.ApplEnerg2006; 83: 1033-46.

[20] Li Q, Meng Q, Cai JJ, Yoshino H, Mochida A. Predicting hourly cooling load in the building: A comparison of support vector machine and different artificial neural

networks.Energ Convers Manage 2009; 50: 90-6.

[21] Yang J, Rivard H, Zmeureanu R. On-line building energy prediction using adaptive artificial neural networks.Energ Buildings2005; 37: 1250-9.

[22] Cui C, Wu T, Hu MQ, Weir JD, Li XW. Short-term building energy model recommendation system: A meta-learning approach.ApplEnerg2016; 172: 251-63.

[23]Grolinger K, L'Heureux A, Capretz MAM, Seewald L. Energy forecasting for event venues: Big data and prediction accuracy.Energ Buildings2016; 112: 222-33.

[24] Lemke C, Gabrys B. Meta-learning for time series forecasting and forecast combination.Neurocomputing 2010; 73: 2006-16.

[25] Matijas M, Suykens JAK, Krajcar S. Load forecasting using a multivariate meta-learning system. Expert SystAppl 2013; 40: 4427-37.

[26] Brandt T, DeForest N, Stadler M, Neumann D. Power systems 2.0: Designing an energy information system for microgrid operation. In Proceedings of 2014 International Conference on Information Systems, December 14-17, 2014, Auckland, New Zealand.

[27] Wang Y, Chen QX, Kang CQ, Zhang MM, Wang K, Zhao Y. Load profiling and its application to demand response: A review. Tsinghua SciTechnol 2015; 20: 117-29.

[28] Schmidhuber J. Deep learning in neural networks: An overview. Neural Networks 2015; 61: 85-117.

[29] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature 2015; 321: 436-44.

[30] Langkvist M, Karlsson L, Loutfi A. A review of unsupervised feature learning and deep learning for time-series modeling. Pattern RecognLett2014; 42: 11-24.

[31] Chen TQ, Guestrin C. XGBoost: A scalable tree boosting system. KDD 2016, August 13-17, 2016, San Francisco, CA, USA.

[32] Goodfellow IJ, Warde-Farley D, Mirza M, Courville A, Bengio Y. Maxout networks. In Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013.

[33] Heaton J. Introduction to neural networks for JAVA. 2nd ed. Heaton Research Inc; 2008.

[34] Fan C, Xiao F, Wang SW. Development of prediction models for next-day building energy consumption and peak power demand using data mining techniques.ApplEnerg2014; 127: 1-10.

[35] American Society of Heating, Refrigerating and Air-conditioning Engineers (ASHRAE). ASHRAE guideline 14-2002: Measurement of energy demand and savings. 2002; Atlanta, GA, USA.

[36] Federal Energy Management Program (FEMP). M&V guidelines: Measurement and verification for federal energy projects version 3.0. 2008; U.S. Department of Energy. Washington, DC, USA.

[37] Reddy TA, Maor I, Panjapornpon C. Calibrating detailed building energy simulation programs with measured data-Part II: Application to three case study office buildings (RP-1051). HVAC&R Res 2007; 13:221-241.

[38] R Development Core Team. R: A language and environment for statistical computing. R Foundation for Statistical Computing.2008, Vienna, Austria, ISBN 3-900051-07-0, URL http://www.R-project.org.

[39] Liaw A, Wiener M. Classification and regression by randomForest.R News 2002; 2(3): 18-22.

[40] Heaton J. Introduction to neural networks for Java. 2nd ed. ; Heaton Research Inc; 2005.