Sustainable Cities and Society 49 (2019) 101587

Contents lists available at ScienceDirect

Sustainable Cities and Society

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

Data-driven building load profiling and energy management

Jin Zhu[a], Yingjun Shen[b], Zhe Song[b,*], Dequn Zhou[a], Zijun Zhang[c], Andrew Kusiak[d]

[a] College of Economics and Management, Nanjing University of Aeronautics and Astronautics, No. 29 Jiangjun Avenue, Nanjing, 211106, China
[b] School of Management, Nanjing University, 22 Hankou Road, Nanjing, 210093, China
[c] School of Data Science, YEUNG-P7318, City University of Hong Kong, Hong Kong
[d] Department of Industrial and Systems Engineering, 4627 Seamans Center for the Engineering Arts and Sciences, The University of Iowa, Iowa City, IA, 52242, United States

ARTICLE INFO

Keywords:
Buildingload profiles
Energy management
Anomaly detection
Machine learning
Prediction model
Datadriven

ABSTRACT

Commercial buildings consume a lot of energy and contribute a significant part of greenhouse gas emission. Many energy-saving or green-building initiatives were compromised by equipment and human-related faults under the umbrella of poor facility management. Data-driven building energy management is a cost-effective approach to improve energy efficiency of commercial buildings, and gains more and more popularity worldwide with the deployment of smart metering systems. This paper developed a systematic process of using smart metering data to quantify building daily load profiles (i.e. energy consumption patterns) with a set of statistics, e.g. base load, peak load, rising time and so on. Then prediction models of these building load statistics are constructed from historical training data consisting of energy consumption, environment and holiday information. At last residuals of the prediction models are analyzed to form statistical control charts. As a result anomaly energy consumption could be detected by comparing the predicted statistics and observed ones, which will help building managers to locate problems just in time. The effectiveness of the proposed solution is verified through real-world data analysis and computational studies.

1. Introduction and literature review

Commercial and residential buildings account for roughly 60% of the world's electricity consumption (Araya, Grolinger, ElYamany, Capretz, & Bitsuamlak, 2017). In U.S.A, about 40% percent of the energy is consumed by buildings (Chen & Wen, 2017). Although China is a developing country, its building energy consumption (account for 20% of China's energy consumption in 2015) is steadily increasing in recent 10 years due to real estate sector's barbarian growth. Energy saving and green building initiatives could run through the whole life cycle of a building, including design, construction, operations and maintenance. Although a building could be designed and constructed in a green and energy-efficient fashion, a significant portion of energy could be wasted if the energy management isn't properly executed during building operation. Unexpected equipment or human-related faults, such as malfunctioning sensors, changed control objectives, fluctuating environment, inexperienced crew and so on, all could create black holes of energy consumption. For example, highrise floodlighting could be shining during daytime for several months without being noticed. Because of malfunctioning converters, water pumps of high-end office buildings are running up and down during midnight when nobody is working.

The aforementioned energy consumption black holes are nibbling away at green energy-efficient building initiatives, and impose additional strains on facility management crews. With the development of smart metering and building automation technology, commercial buildings are equipped with various sensors (e.g. temperature sensors, power meters, flow meters) and generating rich data streams minute by minute. This building big data is stored continuously and could be analyzed to help facility management teams improve their operational performance and reduce building energy waste (Palensky & Dietrich, 2011; Wang, Chen, Hong, & Kang, In Press). Intelligent energy management is an integral part of smart cities (Kylili & Fokaides, 2015).

Recent advances in Big Data Analytics scatter across various subjects and disciplines. Various data-driven methods are developed recently to analyze big historical data, which emerge from healthcare to innovation, from energy systems to smart manufacturing (Kusiak, 2015, 2016; Kusiak, 2017). Descriptive analytics encompasses the set of techniques that describes what has happened in the past (Camm et al., 2015). Examples are data queries, data comparison reports, descriptive

* Corresponding author.
E-mail addresses: zhujin1981@nuaa.edu.cn (J. Zhu), 2723502573@qq.com (Y. Shen), zsong1@nju.edu.cn (Z. Song), dqzhou@nuaa.edu.cn (D. Zhou), zijzhang@cityu.edu.hk (Z. Zhang), andrew-kusiak@uiowa.edu (A. Kusiak).

https://doi.org/10.1016/j.scs.2019.101587
Received 14 November 2018; Received in revised form 12 March 2019; Accepted 1 May 2019
Available online 29 May 2019

statistics, data visualization, and basic what-if spreadsheet models. KPIs of building energy performance could be developed to alert the crew to take necessary actions (Miller, Nagy, & Schlueter, 2018).

Predictive analytics consists of techniques that use models constructed from past data to predict the future or ascertain the impact of one variable on another. For example, data mining algorithms could be used to forecast building energy consumption based on its historical operational data and environmental data (Amasyali & El-Gohary, 2018; Deng, Fannon, & Eckelman, 2018; Edwards, New, & Parker, 2012; Robinson et al., 2017; Touzani, Granderson, & Fernandes, 2018; Wei et al., 2018). Prescriptive analytics differ from descriptive or predictive analytics in that prescriptive analytics indicate a best course of action to take; that is, the output of a prescriptive model is a best decision. For example HVAC system could be optimized with data mining and computational intelligence algorithms (He, Zhang, & Kusiak, 2014; Lin, Afshari, & Azar, 2018; Tang, Kusiak, & Wei, 2014; Zeng, Zhang, & Kusiak, 2015).

Among various data-driven methods for improving building energy efficiency, monitoring the energy consumption with the aim of identifying abnormal patterns is promising and cost-effective. Once identified, this abnormal consumption behavior can be reported to building managers, who can subsequently perform appropriate energy-saving procedures (Araya et al., 2017; Chou, Telaga, Chong, & Gibson, 2017). Recent researches about building energy consumption anomaly detection could be divided into two categories: one is called point anomaly detection; the other is context anomaly detection. The key idea behind this point anomaly detection is using historical energy consumption time series data to build consumption prediction models. Then energy consumption is predicted periodically, and anomalies are identified by comparing whether or not the actual reading deviated significantly from the predicted value. Context anomaly detection utilizes more information other than energy consumption time series to define anomalies. Building construction and material information, local environment, operational rules, domain expertise and so on, all could be integrated to decide whether recent energy consumptions are out of control.

Point anomaly detection for building energy consumptions usually involves two steps: the first step is to collect and preprocess normal building energy consumption time series data with clustering or Fourier transformation algorithms, and then normal energy consumption models are built based on this training data with descriptive or predictive analytics, such as neural networks auto regressive models. The second step is to compare the smart-meter measured energy consumption data with the model predicted one. If these two show significant difference, it is concluded that there is an anomaly (Chou & Telaga, 2014; Chou et al., 2017).

Context anomaly detection methods are diverse and use various categories of statistical learning algorithms. One **typical** way is to transform domain expertise, building operational knowledge, major energy consumption equipment's operating characteristics, and so on, into "If…Then…" rules. These rules will be deployed to monitor the building energy consumption (Peña, Biscarri, Guerrero, Monedero, & León, 2016). Once the "If" condition is satisfied based on the measured data, this rule is triggered and corresponding anomaly is reported. The other **classical** way of context anomaly detection is to transform the energy consumption time series with Symbolic Aggregate Approximation or other sliding window pattern recognition methods (Araya et al., 2017; Chen & Wen, 2017; Capozzoli, Piscitelli, Brandi, Grassi, & Chicco, 2018; Miller, Nagy, & Schlueter, 2015). After transformation, nearest neighborhood, clustering and classification algorithms could be used to determine which patterns are normal and which ones are abnormal.

Other existing researches of context anomaly detection are based on multivariate linear regression and Principal Component analysis. Zoritaa, Fernández-Tempranob, García-Escuderob, and Duque-Pereza (2016) presented a multivariate linear regression model with climatic data, building construction characteristics and activities performed in the building to predict the monthly energy consumption and compare with the observed consumption. Kapetanakis, Mangina, Ridouane, Kouramas, and Finn (2015) used correlation analysis to quantify the relationship between building thermal load and input variables, indicating that ambient temperature and relative humidity are the predominant variables that should be considered into the predictive model. Ploennigs, Chen, Schumann, and Brady (2013) presents a diagnosis method based on the hierarchy of the building's sub-meters and on generalized additive models

Our contribution is the combination of point anomaly detection and context anomaly detection. We don't predict energy consumption and compare predictions with actual readings; instead we predict a set of statistics extracted from daily load time series. Prediction models are trained with supervised statistical learning algorithms on historical energy consumption data as well as other related information, such as weather conditions, holidays and so on. At last, residuals of the prediction model are analyzed with statistical control chart theory (Kusiak, Zheng, & Song, 2009; Long, Wang, Zhang, Song, & Xu, 2015; Montgomery, 2005), upper bounds of the energy consumption anomaly monitoring are determined.

The proposed energy anomaly detection framework has the advantage of utilizing existing context information to preprocess the load time series into a set of statistics, and has the freedom of choosing related predictors (i.e. features or attributes) and best statistical learning algorithms to build prediction models. Statistical control chart theory is very mature and has been widely tested in engineering practices, which could enhance the anomaly monitoring reliability by formally considering residual characteristics of prediction models. The proposed method is the key technology for intelligent energy management for smart cities (Fig. 1).

The remaining sections of the paper are organized as follows: Section 2 provides the background information of load profiling and algorithms to calculate some daily load statistics. Section 3 describes related work of building prediction models. Sections 4 discussed how to analyze the model residuals and formulate corresponding statistical control charts. Section 5 presents the experimental results and discussion, and finally Section 6 concludes the paper.

2. Data-driven building load profiling

For a commercial building equipped with smart metering and building automation (BA) systems, a building can be described by an attribute vector, $\boldsymbol{x} = (x_1, x_2, x_3, \ldots)$, where x_1, x_2, x_3, etc., are attributes, more specifically, a set of relative parameters, such as the building load, outside temperature, outside humidity, HVAC air flow, water pump on/off, etc. x_{it} is the measured value of the i^{th} attribute at time stamp t. These parameters are monitored by the smart metering and BA systems. Their values are continuously recorded in a database with a sampling frequency equals to 5-min or 10-min (Table 1).

Commercial building energy consumption patterns are generally following the daily activities and weather conditions. For building operations management purpose, daily energy consumption patterns are very useful and easy to interpret. Thus this paper is focused on developing algorithms to extract statistics from daily building energy consumption time series. However, the proposed approach is still applicable to other specific time windows.

Let x_1 be a building load variable (e.g. measured by a smart meter with engineering unit kilowatt), x_{1t} will be the load value recorded at time stamp t. For a single day, the time series of a building load is represented as $x_{11}, x_{12}, \ldots, x_{1t}, \ldots, x_{1T}$. If the sampling frequency is 10 min, initial time stamp 1 corresponds to 00:00 AM, the second time stamp corresponds to 00:10 AM, and the last one T corresponds 11:50 PM. There will be total 24*6 = 144 samples for a single day if there are no missing values or damaged readings.

Fig. 2 shows the time series of a 5-star hotel and an office building. It is obvious to see some patterns from the plots.

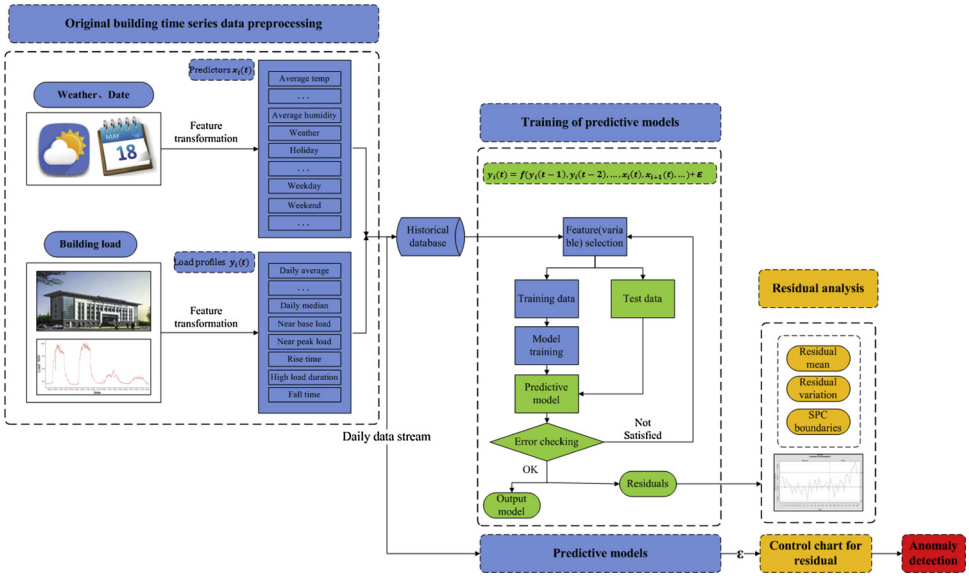


Fig. 1. Process chart of the proposed framework for building energy management.
(a) Office building's load time series of two consecutive days.
(b) 5-star hotel building's load time series of two consecutive days.

Not following previous studies of transforming time series with Symbolic Aggregate Approximation, we quantify the daily load time series with a set of meaningful statistics. For example, the daily average load, the minimum and maximum load, the median load and so on, all could be the useful statistics in later modeling and analysis.

Besides these common descriptive statistics, other meaningful statistics could be defined to describe the load profiles. Near-Base Load, Near-Peak Load, High-Load Duration, Rise Time, Fall Time, these statistics are related with load time series shape and time (Mathieu, Price, Kiliccote, & Piette, 2011). These statistics are useful for building energy management.

Given a building's load observations $x_{11}, x_{12}, \ldots, x_{1t}, \ldots, x_{1T}$ of a single day, **Near-Base Load** y_1 is equal to the 2.5th percentile of the aforementioned daily time series. In other words, y_1 is the load value such that approximately 5% of the observations $(x_{11}, x_{12}, \ldots, x_{1t}, \ldots, x_{1T})$ are at or below this value and approximately 95% of them are above it. Similarly **Near-Peak Load** y_2 is equal to the 95th percentile of the daily time series.

Rise Time y_3 is calculated in three steps:

1 Find a set, called **Lower_Than_Base_Load**, composed of observations x_{1t} from the daily time series $(x_{11}, x_{12}, \ldots, x_{1t}, \ldots, x_{1T})$, satisfying the following condition, $x_{1t} < y_1$, and t is the time stamp between 1 and T. Let t_1 be the smallest (earliest) time stamp in set **Lower_Than_Base_Load**;
2 Find a set, called **Closer_To_Peak**, composed of observations x_{1t} from the time series $(x_{11}, x_{12}, \ldots, x_{1t}, \ldots, x_{1T})$, satisfying the following condition, $x_{1t} > (y_2 - y_1)/2$. Let t_2 be the largest time stamp in set **Closer_To_Peak**;
3 **If** $t_2 > t_1$ **Then** find t_3 which is the smallest time stamp in **Closer_To_Peak**, and at the same time is greater than t_1, find t_4 be the largest time stamp in **Lower_Than_Base_Load** and smaller than t_3, **Rise Time** $y_3 = t_3 - t_4$; **Else Rise Time** = 0.

High-Load Duration y_4 is calculated in three steps:

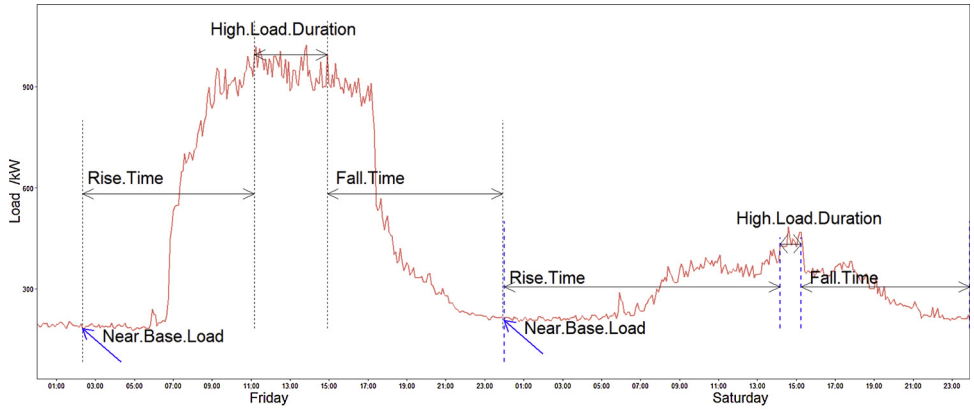
1 Find a set, called **Closer_To_Peak**, composed of observations x_{1t} from the time series $(x_{11}, x_{12}, \ldots, x_{1t}, \ldots, x_{1T})$, satisfying the following condition, $x_{1t} > (y_2 - y_1)/2$;
2 Count the number of observations in this set **Closer_To_Peak**, let L be the number;
3 **High-Load Duration** is equal to $(L + 1)$ times the sampling time interval (e.g. 5 min).

Fall Time y_5 is calculated in three steps:

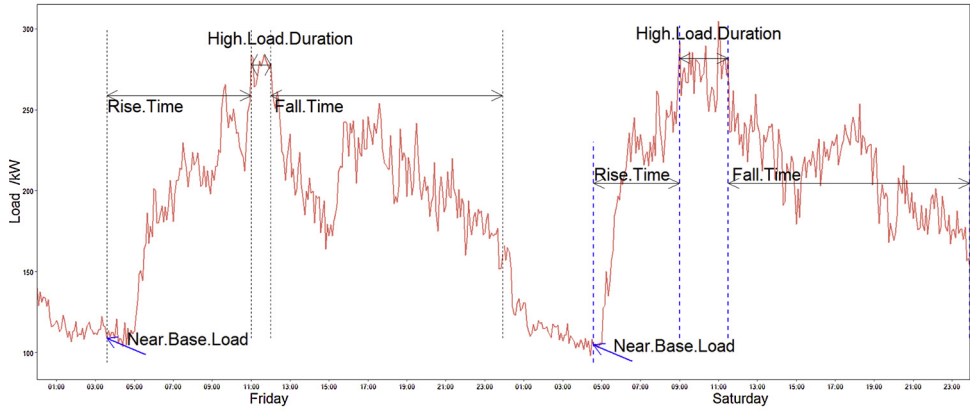
1 Find a set, called **Lower_Than_Base_Load**, composed of observations x_{1t} from the daily time series $(x_{11}, x_{12}, \ldots, x_{1t}, \ldots, x_{1T})$, satisfying the following condition, $x_{1t} < y_1$, and t is the time stamp between 1 and T. Let t_1 be the largest time stamp in set

Table 1
Small sample of original building load time series recorded every 5 min, including weather information.

Time Stamp	Load(kW)	Outside Temperature(°C)	Humidity(%)	Weather Condition
01-01-2016 00:00	207.64	5	100	31
01-01-2016 00:05	197.4	5	100	31
01-01-2016 00:10	190.58	5	100	31
01-01-2016 00:15	192.74	5	100	31
01-01-2016 00:20	195.76	5	100	31
01-01-2016 00:25	202.06	5	100	31
01-01-2016 00:30	192.96	5	100	31
01-01-2016 00:35	197.2	5	99	31
01-01-2016 00:40	189.72	5	96	31
01-01-2016 00:45	187.82	5	96	31
01-01-2016 00:50	188.92	5	96	31
01-01-2016 00:55	190.72	5	96	31



(a) Office building's load time series of two consecutive days



(b) 5-star hotel building's load time series of two consecutive days

Figure 2. Typical commercial building load time series

Fig. 2. Typical commercial building load time series.

Lower_Than_Base_Load;

2 Find a set, called **Closer_To_Peak**, composed of observations x_{1t} from the time series (x_{11}, x_{12}, ...,x_{1t}, ...,x_{1T}), satisfying the following condition, $x_{1t} > (y_2 - y_1)/2$. Let t_2 be the largest time stamp in set **Closer_To_Peak**;

3 **If** $t_1 > t_2$ **Then** find t_3 which is the smallest time stamp in **Lower_Than_Base_Load**, and at the same time is greater than t_2, **Fall Time** $y_5 = t_3 - t_2$;

4 **Else Fall Time** $y_5 = T - t_2$.

Besides the building load profiles, we also preprocess the daily temperature, humidity time series into corresponding daily statistics, such as average temperature, average humidity, median temperature, median humidity and so on. After the preprocessing, a new historical training/test data table is obtained. Basically the original time series data of a building is preprocessed into a set of daily statistics, which is composed of two parts. One part is about building load profiles, extracted from the building load time series. The other part is about the everyday weather condition, temperature (mean, mode, median, min, max), humidity (mean, mode, median, min, max) and holiday information (weekday/weekend, national holidays) (Table 2). Weather condition code could be found in the appendix.

3. Prediction modeling of building load statistics

Once the building load time series are preprocessed into statistics, prediction models can be built by considering weather, temperature, humidity, holidays and other related information. Predictive models are usually composed of an output variable y_i, and a set of related predictors x, x$\in \{x_2, ...,x_p\}$, i.e. $y_i = f_i(\mathrm{x}) + \varepsilon_i$, where ε_i is the residual for prediction model $f_i(\cdot)$. Since the output variable is a daily time series, following the autoregressive formulation, at current time stamp t (t represent a specific date, from now on $t = 1$ means the first day, 2 means the second day, etc), the prediction model is usually written as $y_i(t) = f_i(\mathrm{x}(t), y_i(t-1), y_i(t-2), y_i(t-3)...) + \varepsilon_i$, where output's past values are used as predictors. Let $\boldsymbol{y_i}(t\_)$ be a set of predictors which is composed of past values of the output variable $y_i(t)$, $\boldsymbol{y_i}(t\_) \in \{y_i(t-1), y_i(t-2), ...,y_i(t-d)\}$, d is an integer specifying the maximum number of time steps looking backward, $y_i(t) = f_i(\mathrm{x}(t), \boldsymbol{y_i}(t\_)) + \varepsilon_i$.

Table 2
Building daily load profiles and corresponding weather, holiday information.

	Variable	Description	Type
Daily Load Profles	y_1	Near Base Load	Numeric
	y_2	Near Peak Load	Numeric
	y_3	Rise Time	Numeric
	$y4$	High Load Duration	Numeric
	y_5	Fall Time	Numeric
	y_6	Average Load	Numeric
Temperature Statistics	x_2	**Temperature-Mean**	Numeric
Humidity Statistics	x_3	**Humidity-Mean**	Numeric
Weather& Holiday		Weekday	Categorical
		National Holiday	0 or 1
		Holiday	0 or 1
		Weather condition	Categorical

To build a predictive model, training data set and test data set are prepared for evaluating different learning algorithms and selecting

Table 3
Descriptions of two building time series data sets for computational study.

Load & Weather time series	Mean	Max	Min	Median	SD	# of Observations	# of Missing values	Start time	End time	Sampling frequency(minutes)
Office Building, 2016	441.96364	1394.16	1.5	276.02	317.3806	105408	858	01-01-2016 00:00	31-12-2016 23:55	5
Office Building, 2017	506.32834	1425.72	1.1	331.44	342.25263	105120	1262	01-01-2017 00:00	31-12-2017 23:55	5
Office Building, 2018	528.72527	1493.68	173.76	335.38333	351.45481	56448	0	01-01-2018 00:00	15-07-2018 23:55	5
Hotel, 2016	267.74	1001.5	32.25	202.5	169.47	105408	2926	01-01-2016 00:00	31-12-2016 23:55	5
Hotel, 2017	291.71	1015	60	231.5	171.16	105120	380	01-01-2017 00:00	31-12-2017 23:55	5
Hotel, 2018	264.05	996	36	220	149.15	56448	1637	01-01-2018 00:00	15-07-2018 23:55	5
Humidity, 2016	77.77	100	20	81	18.91	105408	16517	01-01-2016 00:00	31-12-2016 23:55	5
Humidity, 2017	70.71	100	16	74	18.52	105120	300	01-01-2016 00:00	31-12-2016 23:55	5
Humidity, 2018	72.43	100	16	77	18.86	56448	1553	01-01-2016 00:00	31-12-2016 23:55	5
Temperature, 2016	16.36	38.42	−7	16.67	8.74	105408	16520	01-01-2016 00:00	31-12-2016 23:55	5
Temperature, 2017	18.12	41.67	−3.89	18.33	9.49	105120	300	01-01-2016 00:00	31-12-2016 23:55	5
Temperature, 2018	16	37.78	−5	17.22	9.92	56448	1553	01-01-2016 00:00	31-12-2016 23:55	5

Table 4
Number of training and testing daily profiles after preprocessing of the original time series data set.

Trainging and testing daily profiles	# of days	Missing daily profiles
Office Building, 2016	366	3
Office Building, 2017	365	0
Office Building, 2018	196	3
Hotel, 2016	366	0
Hotel, 2017	365	0
Hotel, 2018	196	0

appropriate predictors. Training data set is first used to test which data mining (machine learning) algorithms are good at building the model $f_i(\cdot)$, which predictors will be selected to build the model $f_i(\cdot)$. Usually cross-validation technique will be employed to estimate the model's performance during the training process.

Let $\hat{y}_i(t) = f_i(\mathrm{x}(t), \boldsymbol{y_i}(t\_))$, $\varepsilon_i = y_i(t) - \hat{y}_i(t)$, $y_i(t)$ is the observed value of variable y_i at time t, $\hat{y}_i(t)$ is the model predicted value. Two metrics are used to evaluate the prediction model $f_i(\cdot)$' s performance: MAE (Mean Absolute Error) and MAPE (Mean Absolute Percentage Error). Absolute Error $= |y_i(t) - \hat{y}_i(t)|$,
Absolute Percentage Error $= \left|\frac{y_i(t) - \hat{y}_i(t)}{y_i(t)}\right| \times 100\%$. Given a training data set with N samples, the MAE (Mean Absolute Error) is defined as $\mathrm{MAE} = \frac{1}{N}\sum_{t=1}^{N} |y_i(t) - \hat{y}_i(t)|$, MAPE (Mean Absolute Percentage Error) is defined as $\mathrm{MAPE} = \frac{1}{N}\sum_{t=1}^{N} \left|\frac{y_i(t) - \hat{y}_i(t)}{y_i(t)}\right| \times 100\%$.

Prediction modeling process usually involves two tasks: one is about selecting best data mining algorithms to extract best model from the training data set. The other is about selecting best predictors to build the model. Some benchmark models are usually prepared for comparisons. One popular benchmark model is called persistent model where $y_i(t) = f_i(y_i(t-1)) + \varepsilon_i$. Persistent model is a simplest forecasting model by using the nearest past value of y_i. Persistent model is used to justify the advanced forecasting models with sophisticated data mining algorithms.

Popular data mining algorithms used in building energy consumption/load prediction or fault detection are Linear Regression, Lasso Regression, Support Vector Machine/Regression, Classification and Regression Tree, Artificial Neural Networks, *K*-Nearest Neighbors, Random Forest, Gradient Boosting and so on (Ahmad, Mourshed, & Rezgui, 2017; Amasyali & El-Gohary, 2018; Araya et al., 2017; Deng et al., 2018; Wang, Wang, Zeng, Srinivasan, & Ahrentzen, 2018; Wei et al., 2018). All these data mining algorithms are good candidates for building the prediction model. Recent advances in deep learning algorithms are also worthy of investigation. However this paper is focused on the development of general framework for building load profiling and anomaly detection. Only a subset of these candidate algorithms will be tried in this paper due to the limited content. But we strongly believe that in real engineering applications, more tailored and sophisticated learning algorithms should be researched to further improve the prediction model's accuracy.

Besides the selection of appropriate data mining algorithms, predictor selection is another important part of the predictive modeling process. Theoretically, for each data mining algorithm, given a training data set, there will be a best set of predictors for building the prediction model. When the number of potential predictors is large, searching for the best set of predictors is time consuming. Domain expertise is usually used in speeding up the search process. For example, it obvious that temperature will be a very strong predictor for building energy consumption and load profiles. Other heuristics for searching the best set of predictors are genetic algorithm based random search, greedy stepwise predictor selection based on some information gain criteria and so on (Tan, Steinbach, & Kumar, 2006; Witten, Frank, Hall, & Pal, 2016).

In data mining area, "Divide and Conquer" strategy is good at

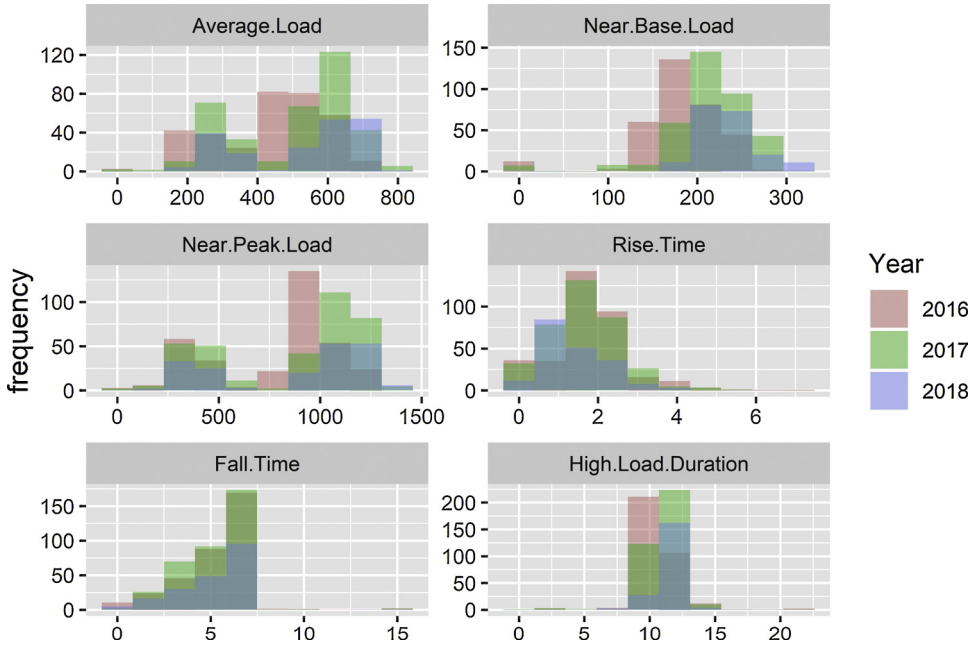


Fig. 3. Office building's load profiles histogram across three consecutive years.

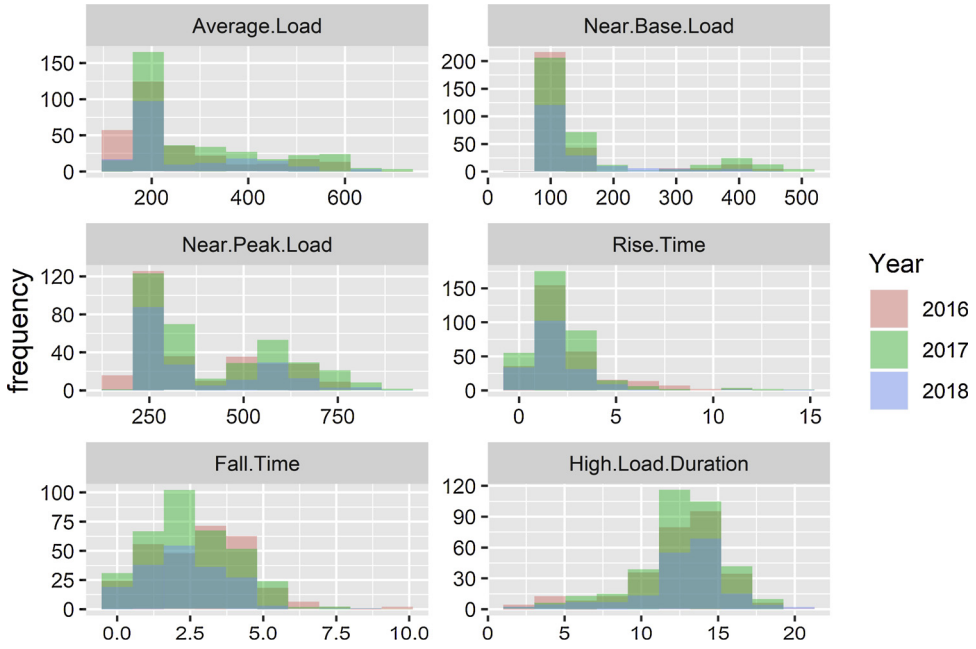


Fig. 4. Hotel's load profiles histogram across three consecutive years.

building more accurate and generalizable prediction models. As building load statistics are influenced by seasonal patterns and transitions, it is very reasonable to divide the training data set into three subsets according to the cooling, heating and transition seasons. Then for each subset training data, applying appropriate data mining algorithms to learn the prediction models. As a result, we would see three prediction models for cooling, heating and transition seasons respectively, which will have a better prediction performance than building only one prediction model.

Following the "Divide and Conquer" strategy, we could build separate prediction models for cooling Monday, cooling Tuesday, cooling Wednesday and so on. It is noteworthy that "Divide and Conquer" strategy may fail if there are not enough training data samples as you try to partition the training data set into too many subsets/categories. In this paper, due to the limited number of training samples (3 years, less than 1 thousand samples), we stick to three partitions: cooling, heating and transition seasons. However, in future engineering practices, we believe that the prediction model's accuracy could be further improved if we consider the weekday, weekend, national holiday and weather information into the partition.

4. Statistical control chart based on residual analysis

How to identify the abnormal energy consumption events is crucial to energy management of buildings. Previous researches (Chou & Telaga, 2014; Chou et al., 2017) presented a two-stage abnormal building energy consumption detection framework. Daily real-time consumption is predicted by using a hybrid neural net ARIMA(auto-

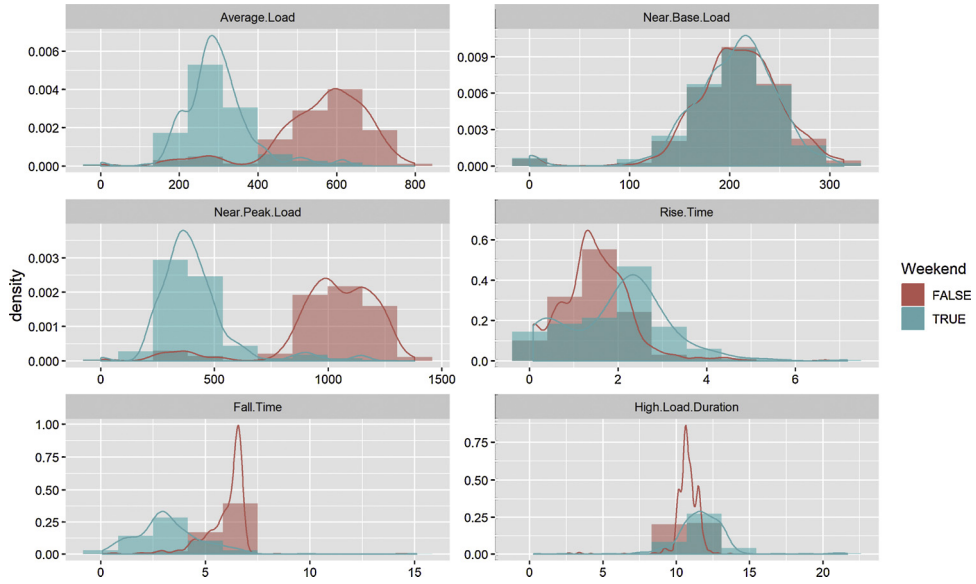


Fig. 5. Office building's load profiles histogram comparison between weekend and weekday.

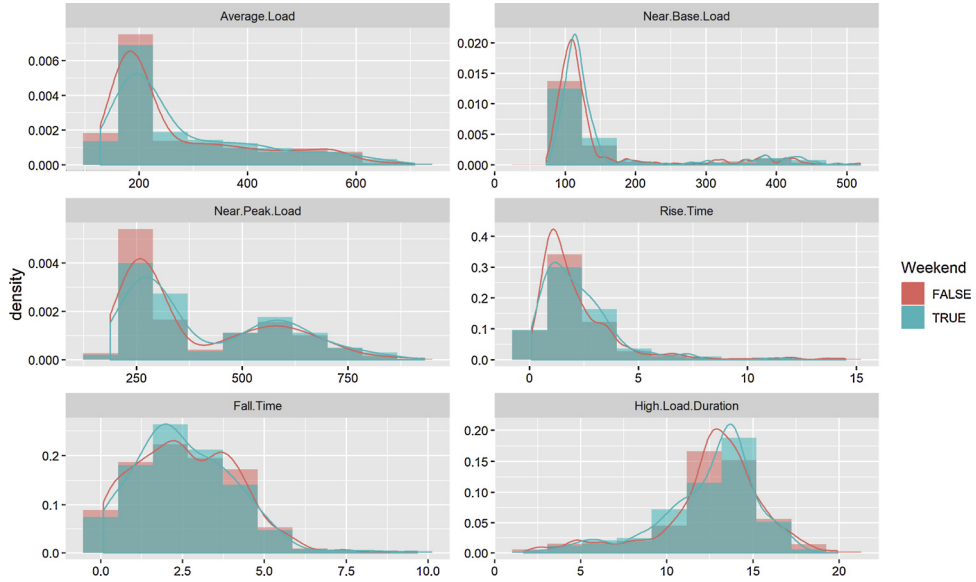


Fig. 6. Hotel's load profiles histogram comparison between weekend and weekday.

regressive integrated moving average) model of daily consumption. Anomalies are then identified by differences between real and predicted consumption by applying the two-sigma rule. This paper extends the idea of aforementioned research by employing the control chart technique and formally take the prediction model's residual into consideration.

Once a prediction model is finalized, its residuals are analyzed carefully and the boundaries of a control chart are determined. Given a training data set, a prediction model's performance can be measured by its training error (Absolute Percentage Error), the average and standard deviation of the training error can be calculated as following:

$$\mu_{\text{Train}} = \frac{1}{N_{\text{Train}}} \sum_{t=1}^{N_{\text{Train}}} \left| \frac{y_i(t) - \hat{y}_i(t)}{y_i(t)} \right|,$$

$$\sigma_{\text{Train}} = \sqrt{\frac{1}{N_{\text{Train}} - 1} \sum_{t=1}^{N_{\text{Train}}} \left(\left| \frac{y_i(t) - \hat{y}_i(t)}{y_i(t)} \right| - \mu_{\text{Train}} \right)^2}.$$

According to the $\bar{X}$ chart definition (Montgomery, 2005), the center line, the upper control limits can be estimated: $\text{UCL} = \mu_{\text{Train}} + \eta\sigma_{\text{Train}}$, $\text{CenterLine} = \mu_{\text{Train}}$, η is a constant and will be determined based on the training data and the sensitivity of identifying abnormal energy consumption events (e.g. false alarm rate).

Generally speaking, given a training data set, it is not easy to assure that there are no abnormal energy consumption events in it. In engineering practice, it is almost impossible for us to go through every line of the data set and check whether this sample is normal or abnormal. Especially when the building big data accumulates day after day and the facility management team doesn't have the necessary manpower to tag all abnormal events. As a result the control limits estimated from the training data set may not be sensitive to real abnormal events. Following the classical control chart technique, a Phase I process is necessary to preprocess the training data set and filter out potential outliers (Kang & Albin, 2000).

For example, typical value of η could be 3, and if one or more of training data set residuals fall outside the control limits then identify assignable causes if possible, and delete these points from the preliminary training data set. Re-train the prediction model with pre-determined data mining algorithm and predictors; recalculate the control limits based on new residuals. Repeat the process until all residuals are within the control limits. Then these control limits could be deployed

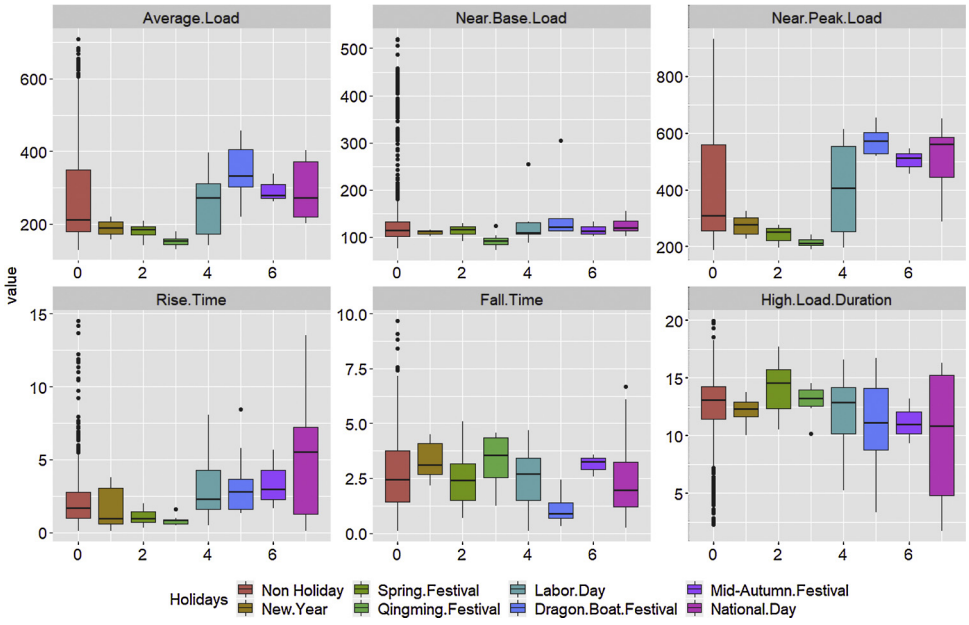


Fig. 7. Boxplot comparison of Office building's load profiles among holidays and non-holiday.

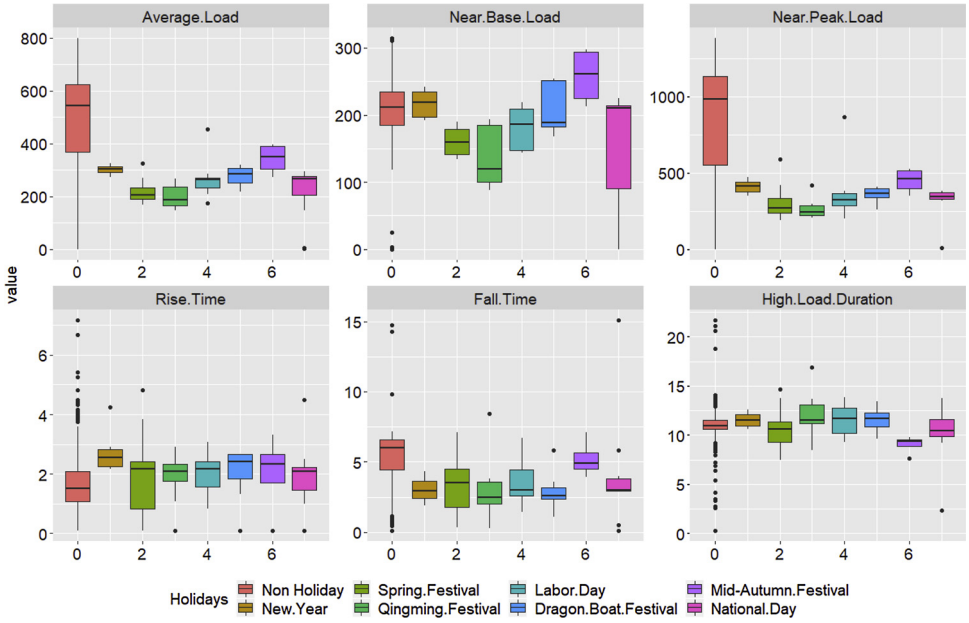


Fig. 8. Boxplot comparison of Hotel's load profiles among holidays and non-holiday.

for abnormal energy consumption detection.

5. Computational experiments

5.1. Descriptive analytics of the building time series data

Two buildings (one is a 5-star hotel, the other is an high-rise office building) both located in Dushu Lake district, Suzhou city, Jiangsu Province, are selected for our research and validate our proposed method. We installed sub-metering system and intelligent energy management platform in two buildings to collect the data.

The load time series data and related weather information are collected and stored in the energy management system. We collected three-year long time series for this computational study and validate our proposed framework. Table 3 lists the number of observations, number of missing values and some descriptive statistics of the original load, outside temperature and humidity time series downloaded from the energy management system. Missing values are caused majorly by sensor malfunction or network disconnection.

Due to some missing values in building time series data, after preprocessing the original time series data into daily profiles, we got 363 rows of data for 2016, 365 rows of data for 2017, and 193 rows of data for 2018 (see Table 4). Hotel time series data has better condition, and after preprocess, there are no missing daily profiles.

Figs. 3 and 4 lists the histograms of the daily load profiles for different years. Based on Fig. 4, hotel's daily load profiles don't show significant differences across different years. However, according to Fig. 3, some office building load profiles show lower values in 2016 compared with 2017-2018. For example histograms of the average load, the near base load and the near peak load of the office building are skewed toward the left. These two pictures show that the load profile of the two buildings didn't change too much in three years. If the change is

Table 5
Number of samples in the training (2017–2017) and testing (2018) data sets for **hotel** load profiles.

Hotel prediction modeling data set	# of Samples	# of Samples with missing y	# of Complete samples	# of cold season samples	# of heat season samples	# of transition season samples	# of Outliers deleted
Average Load 2016-2017	731	3	728	241	243	244	0
Average Load 2018	196	3	193	42	90	61	0
High Load Duration 2016-2017	731	70	661	215	224	222	0
High Load Duration 2018	196	16	180	37	85	58	0
Rise Time 2016-2017	731	70	661	201	223	220	17
Rise Time 2018	196	16	180	30	81	54	15
Fall Time 2016-2017	731	70	661	174	221	210	56
Fall Time 2018	196	16	180	27	85	53	15
Near Base Load 2016-2017	731	70	661	215	224	222	0
Near Base Load 2018	196	16	180	37	85	58	0
Near Peak Load 2016-2017	731	70	661	215	224	222	0
Near Peak Load 2018	196	16	180	37	85	58	0

Table 6
Number of samples in the training (2017-2017) and testing (2018) data sets for **office building** load profiles.

Office building prediction modeling data set	# of Samples	# of Samples with missing y	# of Complete samples	# of cold season samples	# of heat season samples	# of transition season samples	# of Outliers deleted
Average Load 2016-2017	731	0	731	243	242	237	9
Average Load 2018	196	0	196	45	90	61	0
High Load Duration 2016-2017	731	0	731	244	242	242	3
High Load Duration 2018	196	0	196	45	90	61	0
Rise Time 2016-2017	731	2	729	223	212	222	72
Rise Time 2018	196	0	196	43	83	58	12
Fall Time 2016-2017	731	0	731	244	241	241	5
Fall Time 2018	196	0	196	45	90	61	0
Near Base Load 2016-2017	731	0	731	240	236	226	29
Near Base Load 2018	196	0	196	45	90	61	0
Near Peak Load 2016-2017	731	0	731	243	242	237	9
Near Peak Load 2018	196	0	196	45	90	61	0

obvious, the characteristics are not predictable. These two pictures show that it is feasible to use the data of the past two years to model and predict the load profiles of the next year.

Figs. 5 and 6 try to discover the energy consumption differences between weekdays (red) and weekends (green). According to Fig. 5, office building shows significant lower energy consumption during weekends. Histograms of the average load and the near peak load are skewed toward the left in weekends, which conforms to common sense. The rise time of weekends is higher than that of week days. The reason is that most energy-consuming equipments of the office building are turned on between 7 AM and 9 AM during weekdays. On the contrary, the fall time of weekends is lower than that of week days. The explanation is less energy-consuming equipments are turned on during weekends and thus it takes less time to turn off. Fig. 6 shows that hotel energy consumption patterns don't show significant differences between weekdays and weekends, which conforms to common sense and actual hotel operations.

Figs. 7 and 8 try to discover the energy consumption differences among holidays and non-holiday through boxplots. In China, we usually have 7 major holidays, which are New Year (January 1), Spring Festival (usually 7 consecutive days in February, e.g. from February 15 to 21, 2018), Qingming Festival (usually 3 consecutive days in April, e.g. from April 5 to 7, 2018), Labor Day (usually 3 consecutive days around May 1, 2018), Dragon Boat Festival (usually 3 consecutive days in June, e.g. from June 16 to 18, 2018), Mid-Autumn Festival ((usually 3 consecutive days in September, e.g. from September 22 to 24, 2018), National Day (usually 7 consecutive days in October, e.g. from October 1 to 7, 2018). The patterns for office building are obvious (see Fig. 7) as the energy consumption is lower during these holidays. From Fig. 8, we can see that Labor Day, Dragon Boat Festival, Mid-Autumn Festival and National Day have higher energy consumption than other holidays and non-holidays. The reason is that people are willing to spend time traveling and living in the hotel during these holidays. Other holidays, such as Spring Festival, people are more willing to spend time with families and relatives in their hometowns.

Based on these descriptive analytics, we can conclude that weekend and holiday information are important factors for office building load profile modeling. Hotel energy consumption patterns are not sensitive to the weekdays and weekends changeover. Holiday information does have an impact on the hotel energy consumption patterns, which may be considered during the modeling process. However, due to limited sample sizes, in this paper we are not going to incorporate these factors into our prediction models. Future researches could be expected to follow this direction and build more complex prediction models by considering this valuable information.

5.2. Prediction modeling: predictors selection/algorithms selection

Tables 5 and 6 list the training (2016–2017) and testing (2018) data sets for building prediction models of the daily load profiles. For example, in Table 5, data set "Rise Time 2016–2017" originally has total 731 rows of data (# of Samples). But due to some missing values in the original time series, calculating the "Rise Time" may not be feasible, thus lead to 70 rows of data without the "Rise Time" (# of Samples with missing y"). After removing these samples without y, "Rise Time 2016–2017" data set is further divided into three subsets based on the three operating seasons. The sample size is further reduced by deleting some obvious outliers (# of Outliers deleted). After these preprocessing steps, training and testing data set are ready for further data mining and prediction modeling.

In this paper computational study of prediction modeling involves two parts: one is about selecting the best predictors and data mining algorithms. Seven data mining algorithms: Linear Regression, LASSO, SVM, CART, Neural Networks, kNN and Random Forests are tried with

Table 7
Best selected prediction models of **hotel load profiles** based on the 2016–2017 training data, after different prediction algorithms with different predictor combinations (5-fold cross-validation).

Data Set	Load Profiles	Best Prediction Algrithms	MAE	MAPE	MAE_SD	MAPE_SD	Settings	Predictors
Not Divide	Near Base Load	SVM2	17.66	10.48	28.60	11.66	default	$y_1(t\text{-}1)$, $y_1(t\text{-}2)$, $x_2(t)$, $x_3(t)$
	Near Peak Load	SVM12	45.06	11.42	44.85	11.25	default	$y_2(t\text{-}1)$, $y_2(t\text{-}2)$, $y_2(t\text{-}3)$, $y_2(t\text{-}4)$, $y_2(t\text{-}5)$, $y_2(t\text{-}6)$, $y_2(t\text{-}7)$,$x_2(t)$, $x_3(t)$
	Rise Time	SVM12	0.97	53.48	1.18	57.87	default	$y_3(t\text{-}1)$, $y_3(t\text{-}2)$, $y_3(t\text{-}3)$, $y_3(t\text{-}4)$, $y_3(t\text{-}5)$, $y_3(t\text{-}6)$, $y_3(t\text{-}7)$,$x_2(t)$, $x_3(t)$
	High Load Duration	Random Forests9	1.93	21.06	1.71	38.76	default	$y_4(t\text{-}1)$, $y_4(t\text{-}2)$, $y_4(t\text{-}3)$, $y_4(t\text{-}4)$, $y_4(t\text{-}5)$, $y_4(t\text{-}6)$,$y_4(t\text{-}7)$, $x_2(t)$, $x_3(t)$
	Fall Time	SVM6	1.16	59.84	0.96	83.97	default	$y_5(t\text{-}1)$, $y_5(t\text{-}6)$, $x_2(t)$, $x_3(t)$
	Average Load	Random Forests6	23.85	9.06	22.25	7.84	default	$y_6(t\text{-}1)$, $y_6(t\text{-}6)$, $x_2(t)$, $x_3(t)$
Cooling	Near Base Load	SVM2	35.49	14.97	44.53	16.55	default	$y_1(t\text{-}1)$, $y_1(t\text{-}2)$, $x_2(t)$, $x_3(t)$
	Near Peak Load	Linear regression1	55.67	9.18	39.83	6.76	default	$y_2(t\text{-}1)$,$x_2(t)$, $x_3(t)$
	Rise Time	SVM12	0.93	46.54	1.10	56.78	default	$y_3(t\text{-}1)$, $y_3(t\text{-}2)$, $y_3(t\text{-}3)$, $y_3(t\text{-}4)$, $y_3(t\text{-}5)$, $y_3(t\text{-}6)$, $y_3(t\text{-}7)$,$x_2(t)$, $x_3(t)$
	High Load Duration	Random Forests6	2.06	18.73	1.63	24.78	default	$y_4(t\text{-}1)$, $y_4(t\text{-}6)$, $x_2(t)$, $x_3(t)$
	Fall Time	SVM12	0.93	56.83	0.89	58.66	default	$y_5(t\text{-}1)$, $y_5(t\text{-}2)$, $y_5(t\text{-}3)$, $y_5(t\text{-}4)$, $y_5(t\text{-}5)$, $y_5(t\text{-}6)$, $y_5(t\text{-}7)$, $x_2(t)$, $x_3(t)$
	Average Load	LASSO	33.92	8.16	28.22	7.55	s = 0.028812	$y_6(t\text{-}1)$, $y_6(t\text{-}4)$, $y_6(t\text{-}5)$, $x_2(t)$, $x_3(t)$
Heating	Near Base Load	SVM7	7.17	6.70	6.47	5.96	default	$y_1(t\text{-}1)$, $y_1(t\text{-}7)$, $x_2(t)$, $x_3(t)$
	Near Peak Load	Linear regression6	25.03	9.41	18.19	6.96	default	$y_2(t\text{-}1)$, $y_2(t\text{-}6)$,$x_2(t)$, $x_3(t)$
	Rise Time	LASSO	0.69	47.81	0.65	41.18	s = 0.244463	$y_3(t\text{-}6)$,$x_2(t)$, $x_3(t)$
	High Load Duration	LASSO	1.26	10.78	1.12	14.49	s = 0.149867	$y_4(t\text{-}2)$, $y_4(t\text{-}3)$, $y_4(t\text{-}4)$, $y_4(t\text{-}6)$, $x_3(t)$
	Fall Time	LASSO	1.04	42.72	0.83	68.73	s = 0.331071	$y_5(t\text{-}1)$, $y_5(t\text{-}3)$, $x_3(t)$
	Average Load	LASSO	13.37	7.16	10.03	5.35	s = 0.151638	$y_6(t\text{-}1)$, $y_6(t\text{-}2)$
Transition	Near Base Load	SVM7	8.32	7.70	7.29	6.22	default	$y_1(t\text{-}1)$, $y_1(t\text{-}7)$, $x_2(t)$, $x_3(t)$
	Near Peak Load	SVM4	48.14	14.53	48.30	13.99	default	$y_2(t\text{-}1)$, $y_2(t\text{-}4)$,$x_2(t)$, $x_3(t)$
	Rise Time	SVM5	1.39	60.82	1.89	66.15	default	y3(t-1), y3(t-5), x2(t), x3(t)
	High Load Duration	CART9	2.41	30.26	2.47	46.86	default	$y_4(t\text{-}1)$, $y_4(t\text{-}2)$, $y_4(t\text{-}3)$, $y_4(t\text{-}4)$, $x_2(t)$, $x_3(t)$
	Fall Time	LASSO	1.26	58.02	1.08	74.53	s = 0.093248	$y_5(t\text{-}2)$, $y_5(t\text{-}3)$, $y_5(t\text{-}4)$, $y_5(t\text{-}5)$, $y_5(t\text{-}6)$, $y_5(t\text{-}7)$
	Average Load	SVM10	24.88	11.28	24.81	10.10	default	$y_6(t\text{-}1)$, $y_6(t\text{-}2)$, $y_6(t\text{-}3)$, $y_6(t\text{-}4)$, $y_6(t\text{-}5)$, $x_2(t)$, $x_3(t)$

Table 8
Best selected prediction model of the **hotel** average load and its performance.

Average load (y6), Random Forest6, trained on 2016-2017 data set						
Test data set	MAE	MAPE	MAE_SD	MAPE_SD	Settings	Predictors
2016-2017	11.37	4.32	10.56	3.79	default	$y_6(t\text{-}1)$, $y_6(t\text{-}6)$, $x_2(t)$, $x_3(t)$
2018	11.07	4.23	12.05	4.18	default	$y_6(t\text{-}1)$, $y_6(t\text{-}6)$, $x_2(t)$, $x_3(t)$

Table 10
Best selected prediction model of the hotel **Near Base Load** and its performance.

Near Base Load (y_1), SVM2, trained on 2016-2017 data set						
Test data set	MAE	MAPE	MAE_SD	MAPE_SD	Settings	Predictors
2016-2017	15.53	9.31	27.21	10.30	default	$y_1(t\text{-}1)$, $y_1(t\text{-}2)$, $x_2(t)$, $x_3(t)$
2018	17.54	9.73	31.68	10.97	default	$y_1(t\text{-}1)$, $y_1(t\text{-}2)$, $x_2(t)$, $x_3(t)$

different combinations of predictors. Persistent model is used as a benchmark to justify these more advanced prediction models. The other is about studying the effectiveness of the "Divide-and-Conquer" strategy. In Table 7, we tried different combination of prediction algorithms and predictors. Table 7 shows best selected prediction models of hotel load profiles based on the 2016–2017 training data (5-fold cross-validation), which is selected by trying different prediction algorithms with different predictor combinations. Table 7 also shows the computational experiments of predicting the hotel load profiles by dividing the training data into three seasons: cooling, heating and transition. For example, in cooling season, LASSO algorithm with a tuned parameter $s = 0.028812$ and predictors $y_6(t\text{-}1)$, $y_6(t\text{-}4)$, $y_6(t\text{-}5)$, $x_2(t)$ and $x_3(t)$ achieved the best prediction performance of average load; in heating season, LASSO algorithm with a tuned parameter $s = 0.151638$ and predictors $y_6(t\text{-}1)$ and $y_6(t\text{-}2)$ achieved the best prediction performance of average load; in transition season, SVM algorithm with default setting and predictors $y_6(t\text{-}1)$, $y_6(t\text{-}2)$, $y_6(t\text{-}3)$, $y_6(t\text{-}4)$,

Table 9
Best selected prediction models of the **hotel** average load for three different seasons, trained on the 2016–2017 data sets, and their performances on test data sets.

Test data set	Season	Selected model	MAE	MAPE	MAE_SD	MAPE_SD	Settings	Predictors
2016–2017	Cooling	LASSO	34.25	8.45	26.64	7.74	s = 0.028812	$y_6(t\text{-}1)$, $y_6(t\text{-}4)$, $y_6(t\text{-}5)$, $x_2(t)$, $x_3(t)$
	Heating	LASSO	14.23	7.79	9.69	5.61	s = 0.151638	$y_6(t\text{-}1)$, $y_6(t\text{-}2)$
	Transition	SVM10	17.73	8.12	19.78	8.09	default	$y_6(t\text{-}1)$, $y_6(t\text{-}2)$,$y_6(t\text{-}3)$,$y_6(t\text{-}4)$, $y_6(t\text{-}5)$, $x_2(t)$, $x_3(t)$
2018	Cooling	LASSO	31.09	7.47	27.19	7.96	s = 0.028812	$y_6(t\text{-}1)$, $y_6(t\text{-}4)$, $y_6(t\text{-}5)$, $x_2(t)$, $x_3(t)$
	Heating	LASSO	13.1	7.03	8.31	4.53	s = 0.151638	$y_6(t\text{-}1)$, $y_6(t\text{-}2)$
	Transition	SVM10	21.48	8.82	22.58	8.57	default	$y_6(t\text{-}1)$, $y_6(t\text{-}2)$,$y_6(t\text{-}3)$,$y_6(t\text{-}4)$, $y_6(t\text{-}5)$, $x_2(t)$, $x_3(t)$

Table 11
Best selected prediction models of the hotel **Near Base Load** for three different seasons, trained on the 2016–2017 data sets, and their performances on test data sets.

Test data set	Season	Selected model	MAE	MAPE	MAE_SD	MAPE_SD	Settings	Predictors
2016–2017	Cooling	SVM2	30.77	12.78	42.38	15.96	default	$y_1(t\text{-}1)$, $y_1(t\text{-}2)$, $x_2(t)$, $x_3(t)$
	Heating	SVM7	5.87	5.42	6.43	5.77	default	$y_1(t\text{-}1)$, $y_1(t\text{-}7)$, $x_2(t)$, $x_3(t)$
	Transition	SVM7	6.65	6.09	7.35	6.06	default	$y_1(t\text{-}1)$, $y_1(t\text{-}7)$, $x_2(t)$, $x_3(t)$
2018	Cooling	SVM2	28.61	12.47	37.05	12.50	default	$y_1(t\text{-}1)$, $y_1(t\text{-}2)$, $x_2(t)$, $x_3(t)$
	Heating	SVM7	4.56	4.15	4.86	4.38	default	$y_1(t\text{-}1)$, $y_1(t\text{-}7)$, $x_2(t)$, $x_3(t)$
	Transition	SVM7	20.58	12.29	37.22	13.38	default	$y_1(t\text{-}1)$, $y_1(t\text{-}7)$, $x_2(t)$, $x_3(t)$

Table 12
Best selected prediction model of the hotel **Near Peak Load** and its performance.

Near Peak Load (y_2), SVM12, trained on 2016-2017 data set

Test data set	MAE	MAPE	MAE_SD	MAPE_SD	Settings	Predictors
2016-2017	35.60	9.15	36.04	8.64	default	$y_2(t\text{-}1)$, $y_2(t\text{-}2)$, $y_2(t\text{-}3)$, $y_2(t\text{-}4)$, $y_2(t\text{-}5)$, $y_2(t\text{-}6)$, $y_2(t\text{-}7)$, $x_2(t)$, $x_3(t)$
2018	23.47	6.64	31.58	7.12	default	$y_2(t\text{-}1)$, $y_2(t\text{-}2)$, $y_2(t\text{-}3)$, $y_2(t\text{-}4)$, $y_2(t\text{-}5)$, $y_2(t\text{-}6)$, $y_2(t\text{-}7)$, $x_2(t)$, $x_3(t)$

Table 13
Best selected prediction models of the hotel **Near Peak Load** for three different seasons, trained on the 2016–2017 data sets, and their performances on test data sets.

Test data set	Season	Selected model	MAE	MAPE	MAE_SD	MAPE_SD	Settings	Predictors
2016–2017	Cooling	LASSO	34.25	8.45	26.64	7.74	s = 0.028812	$y_6(t\text{-}1)$, $y_6(t\text{-}4)$, $y_6(t\text{-}5)$, $x_2(t)$, $x_3(t)$
	Heating	LASSO	14.23	7.79	9.69	5.61	s = 0.151638	$y_6(t\text{-}1)$, $y_6(t\text{-}2)$
	Transition	SVM10	17.73	8.12	19.78	8.09	default	$y_6(t\text{-}1)$, $y_6(t\text{-}2)$,$y_6(t\text{-}3)$,$y_6(t\text{-}4)$, $y_6(t\text{-}5)$, $x_2(t)$, $x_3(t)$
2018	Cooling	LASSO	31.09	7.47	27.19	7.96	s = 0.028812	$y_6(t\text{-}1)$, $y_6(t\text{-}4)$, $y_6(t\text{-}5)$, $x_2(t)$, $x_3(t)$
	Heating	LASSO	13.1	7.03	8.31	4.53	s = 0.151638	$y_6(t\text{-}1)$, $y_6(t\text{-}2)$
	Transition	SVM10	21.48	8.82	22.58	8.57	default	$y_6(t\text{-}1)$, $y_6(t\text{-}2)$,$y_6(t\text{-}3)$,$y_6(t\text{-}4)$, $y_6(t\text{-}5)$, $x_2(t)$, $x_3(t)$

Table 14
Best selected prediction model of the hotel **Rise Time** and its performance.

Rise Time (y_3), SVM12, trained on 2016-2017 data set

Test data set	MAE	MAPE	MAE_SD	MAPE_SD	Settings	Predictors
2016-2017	0.70	38.31	0.97	43.61	default	$y_3(t\text{-}1)$, $y_3(t\text{-}2)$, $y_3(t\text{-}3)$, $y_3(t\text{-}4)$, $y_3(t\text{-}5)$, $y_3(t\text{-}6)$, $y_3(t\text{-}7)$, $x_{2(}t)$, $x_3(t)$
2018	0.61	28.57	1.10	31.73	default	$y_3(t\text{-}1)$, $y_3(t\text{-}2)$, $y_3(t\text{-}3)$, $y_3(t\text{-}4)$, $y_3(t\text{-}5)$, $y_3(t\text{-}6)$, $y_3(t\text{-}7)$, $x_2(t)$, $x_3(t)$

Table 15
Best selected prediction models of the hotel **Rise Time** for three different seasons, trained on the 2016–2017 data sets, and their performances on test data sets.

Test data set	Season	Selected model	MAE	MAPE	MAE_SD	MAPE_SD	Settings	Predictors
2016-2017	Cooling	SVM12	0.61	30.75	0.91	47.14	default	$y_3(t\text{-}1)$, $y_3(t\text{-}2)$, $y_3(t\text{-}3)$, $y_3(t\text{-}4)$, $y_3(t\text{-}5)$, $y_3(t\text{-}6)$, $y_3(t\text{-}7)$, $x_{2(}t)$, $x_3(t)$
	Heating	Lasso	0.74	62.85	0.55	55.62	s = 0.244463	$y_3(t\text{-}6)$,$x_{2(}t)$, $x_3(t)$
	Transition	SVM5	1.14	46.89	1.80	50.80	default	$y_3(t\text{-}1)$, $y_3(t\text{-}5)$, $x_{2(}t)$, $x_3(t)$
2018	Cooling	SVM12	0.48	29.32	0.49	40.68	default	$y_3(t\text{-}1)$, $y_3(t\text{-}2)$, $y_3(t\text{-}3)$, $y_3(t\text{-}4)$, $y_3(t\text{-}5)$, $y_3(t\text{-}6)$, $y_3(t\text{-}7)$, $x_{2(}t)$, $x_3(t)$
	Heating	Lasso	0.64	46.42	0.48	43.49	s = 0.244463	$y_3(t\text{-}6)$,$x_{2(}t)$, $x_3(t)$
	Transition	SVM5	1.19	72.42	1.81	101.81	default	$y_3(t\text{-}1)$, $y_3(t\text{-}5)$, $x_{2(}t)$, $x_3(t)$

Table 16
Best selected prediction model of the hotel **High Load Duration** and its performance.

High Load Duration (y_4), random Forest9, trained on 2016-2017 data set

Test data set	MAE	MAPE	MAE_SD	MAPE_SD	Settings	Predictors
2016-2017	0.86	9.41	0.79	18.56	default	$y_4(t\text{-}1)$, $y_4(t\text{-}2)$, $y_4(t\text{-}3)$, $y_4(t\text{-}4)$, $x_2(t)$, $x_3(t)$
2018	0.87	10.86	0.89	23.91	default	$y_4(t\text{-}1)$, $y_4(t\text{-}2)$, $y_4(t\text{-}3)$, $y_4(t\text{-}4)$, $x_2(t)$, $x_3(t)$

$y_6(t\text{-}5)$, $x_2(t)$ and $x_3(t)$ achieved the best prediction performance of average load.

Finally random forests algorithm with predictors $y_6(t\text{-}1)$, $y_6(t\text{-}2)$, $x_2(t)$ and $x_3(t)$ achieved the best prediction performance (5-fold cross-validation). Then we fix the random forest algorithm and the selected predictors, re-train the model based on the total 2016–2017 data set, and test the trained model on the 2018 data set. Table 8 shows the trained random forest model's prediction performance of hotel building on 2016–2017 and 2018 data set. "MAE_SD" stands for the Standard Deviation of the Absolute Error, "MAPE_SD" stands for the Standard

Table 17
Best selected prediction models of the hotel **High Load Duration** for three different seasons, trained on the 2016–2017 data sets, and their performances on test data sets.

Test data set	Season	Selected model	MAE	MAPE	MAE_SD	MAPE_SD	Settings	Predictors
2016–2017	Cooling	Random Forests6	1.02	9.48	0.87	15.04	default	$y_4(t-1)$, $y_4(t-6)$, $x_2(t)$, $x_3(t)$
	Heating	LASSO	1.32	11.61	1.21	15.64	s = 0.149867	$y_4(t-2)$, $y_4(t-3)$, $y_4(t-4)$, $x_3(t)$
	Transition	CART9	1.71	20.95	1.73	33.04	default	$y_4(t-1)$, $y_4(t-2)$, $y_4(t-3)$, $y_4(t-4)$, $x_2(t)$, $x_3(t)$
2018	Cooling	Random Forests6	1.3	11.98	0.87	12.30	default	$y_4(t-1)$, $y_4(t-6)$, $x_2(t)$, $x_3(t)$
	Heating	LASSO	1.08	8.52	0.89	7.93	s = 0.149867	$y_4(t-2)$, $y_4(t-3)$, $y_4(t-4)$, $x_3(t)$
	Transition	CART9	2.34	33.89	1.86	51.71	default	$y_4(t-1)$, $y_4(t-2)$, $y_4(t-3)$, $y_4(t-4)$, $x_2(t)$, $x_3(t)$

Table 18
Best selected prediction model of the hotel **Fall Time** and its performance.

Fall Time (y_5), SVM6, trained on 2016-2017 data set						
Test data set	MAE	MAPE	MAE_SD	MAPE_SD	Settings	Predictors
2016-2017	1.00	51.80	0.93	81.75	default	$y_5(t-1)$, $y_5(t-6)$, $x_2(t)$, $x_3(t)$
2018	0.80	47.12	0.87	77.40	default	$y_5(t-1)$, $y_5(t-6)$, $x_2(t)$, $x_3(t)$

Deviation of the Absolute Percentage Error. Table 8 also proves that the prediction model of the hotel average load trained on the historical data could be applicable to the next year's daily average load prediction with satisfactory errors.

Table 9 summarized the best selected prediction models' performance of hotel average load on 2016–2017 and 2018 data set. By comparing Tables 8 and 9, we can see that the "Divide-and-Conquer" strategy didn't improve the prediction performance. One potential reason is that when the training sample size is limited, this strategy is not always effective. Tables 10–19 summarized the prediction models for different building profiles. It is easy to see that the "Divide-and-Conquer" strategy improves the prediction performance significantly in some scenarios. But most of the computational experiments show that training prediction models from the whole 2016–2017 data set provides stable and better prediction performance. Thus in the following computational study of control chart construction and outlier detection, we will train the prediction model based on the 2016–2017 data set without "Divide-and-Conquer". The training residuals will be analyzed to determine appropriate control chart boundaries.

Through the computational study of prediction modeling, we can argue that best prediction algorithm and predictors are determined by the data. Sometimes simple prediction algorithms, such as linear regression, *k*NN or CART could achieve similar or even best prediction accuracy. More machine learning techniques such ensemble learning (Araya et al., 2017), could be applied to further improve the prediction performance.

5.3. Residual analysis and control chart boundaries

To demonstrate the construction of control chart and selection of appropriate upper control limits, we analyzed training errors of the **average load** and **near base load**. The average load and near base load are two important statistics for building energy management and easy to interpret. Higher average load and near base load indicate more energy consumption. Monitoring the average load and near based load will provide the building facility management team with just-in-time information to locate potential abnormal energy consumptions and save energy.

Recall that the selected prediction models are trained on the 2016–2017 data sets. The center line (CenterLine) and upper control limit (UCL) are calculated according to the definitions in Section 4. Fig. 9 shows the CenterLine and UCLs for the hotel average load. The bottom line is the CenterLine of the control chart, the top line is the UCL with $\eta = 3$. The middle two lines are the UCLs with $\eta = 2$ and $\eta = 1$ respectively. It is easy to see that with the increase of η (1–3), less outliers are filtered and the control chart is less sensitive for outlier detection. When $\eta = 3$, there are 12 outliers above the line CenterLine $+3^*\sigma$. When $\eta = 2$, there are 29 outliers above the line CenterLine $+2^*\sigma$. When $\eta = 1$, there are 90 outliers above the line CenterLine $+1^*\sigma$. Not all outliers could be classified as abnormal energy consumption. For energy saving purpose, we are more interested in those outliers with excessive energy consumption. In other words, we try to find which date is using more energy than expected. Table 20 listed 12 average load outliers identified by the control chart with $\eta = 3$. Among these outliers, only one outlier (2016/10/20) is using more energy than expected, i.e. the predicted average load is significantly smaller than the observed average load, which may indicate an abnormal energy consumption and energy saving opportunity.

Similarly, when $\eta = 2$, among these 29 average load outliers, 3 outliers are potential candidates of excessive energy consumption. When $\eta = 1$, among these 90 average load outliers, 25 outliers are potential candidates of excessive energy consumption. Selection of the appropriate value η will be determined by verifying these potential candidates. If all the 25 outliers are genuine abnormal energy consumptions, $\eta = 1$ is a good choice to set up the upper control limit. If some of the 25 outliers are not genuine abnormal energy consumptions, $\eta = 1$ may be too sensitive and will cause false alarms. Thus using larger η could be an alternative. In this paper we will not show how to find the exact optimal η value. But the idea and process presented in this paper is clear enough for engineering practitioners to follow.

Fig. 10 shows that average load of 2016/10/20 is significantly higher than that of 2016/10/18 and 2016/10/19. However the outside

Table 19
Best selected prediction models of the hotel **Fall Time** for three different seasons, trained on the 2016–2017 data sets, and their performances on test data sets.

Test data set	Season	Selected model	MAE	MAPE	MAE_SD	MAPE_SD	Settings	Predictors
2016-2017	Cooling	SVM12	0.55	28.16	0.73	34.44	default	$y_5(t-1)$, $y_5(t-2)$, $y_5(t-3)$, $y_5(t-4)$, $y_5(t-5)$, $y_5(t-6)$, $y_5(t-7)$, $x_2(t)$, $x_3(t)$
	Heating	LASSO	1.07	48.02	0.86	77.07	s = 0.331071	$y_5(t-1)$, $x_3(t)$
	Transition	LASSO	1.27	76.33	0.91	109.47	s = 0.093248	$y_5(t-2)$, $y_5(t-3)$, $y_5(t-4)$, $y_5(t-5)$, $y_5(t-6)$, $y_5(t-7)$
2018	Cooling	SVM12	0.32	17.92	0.69	18.70	default	$y_5(t-1)$, $y_5(t-2)$, $y_5(t-3)$, $y_5(t-4)$, $y_5(t-5)$, $y_5(t-6)$, $y_5(t-7)$, $x_2(t)$, $x_3(t)$
	Heating	LASSSO	0.84	38.63	0.56	51.65	s = 0.331071	$y_5(t-1)$, $x_3(t)$
	Transition	LASSO	1.2	87.39	1.10	117.18	s = 0.093248	$y_5(t-2)$, $y_5(t-3)$, $y_5(t-4)$, $y_5(t-5)$, $y_5(t-6)$, $y_5(t-7)$

Prediction modeling result of the office building could be found in the appendix. Similar conclusions could be drawn from these tables.

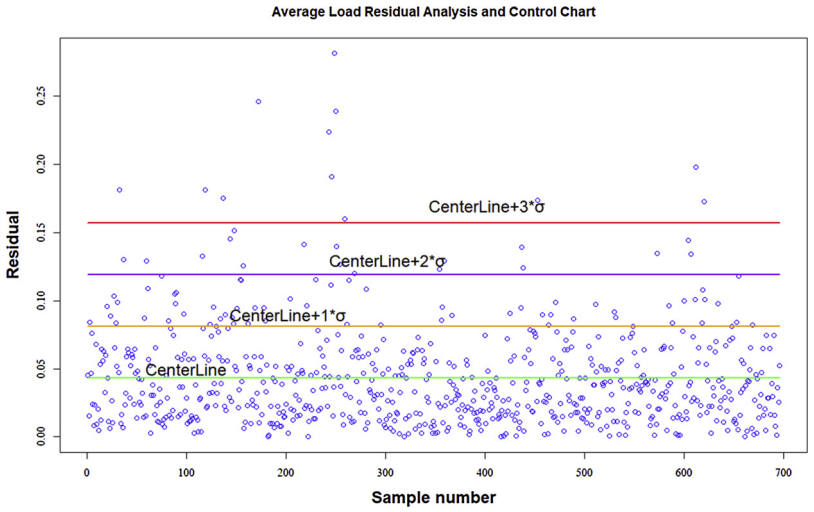


Fig. 9. Hotel average load residual analysis and corresponding control chart boundaries.

Table 20
Hotel average load control chart identified 12 outliers with $\eta = 3$.

Date	Observed Average Load	Predicted Average Load	Residual
08-02-2016	122.1	144.3	22.2
05-05-2016	149.8	177	27.2
23-05-2016	167.7	197.2	29.4
27-06-2016	231.4	288.3	57
01-10-2016	198.9	243.4	44.6
06-10-2016	233.7	278.4	44.7
10-10-2016	160.7	205.9	45.2
11-10-2016	154.6	191.6	37
20-10-2016	332.9	279.7	-53.2
02-05-2017	157.6	185	27.4
08-10-2017	200.6	240.3	39.7
16-10-2017	166.7	195.6	28.8

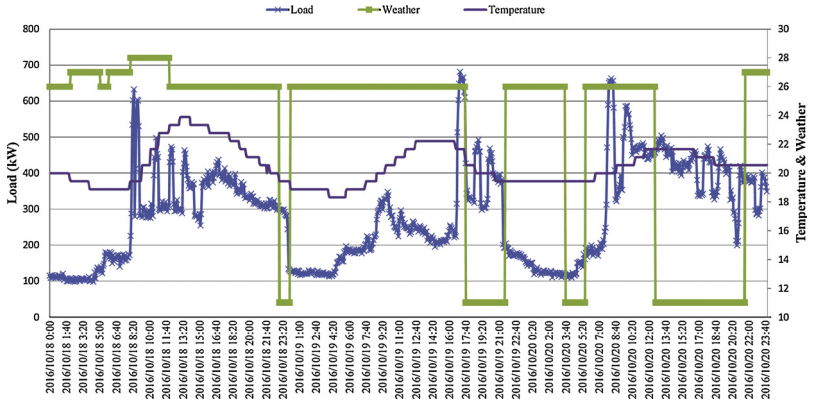


Fig. 10. Hotel's load, temperature and weather code time series from 2016/10/18 to 2016/10/20.

temperature and weather didn't vary too much across these three days, which indicated potential unnecessary excessive energy consumption. The average load outlier in 2016/10/20 was verified by the hotel energy management crew, where they found that some of the air conditioning equipments were turned on too early in the 7 o'clock in the morning.

Fig. 11 shows the CenterLine and UCLs for the hotel near base load control chart. The bottom line is the CenterLine, the top line is the UCL with $\eta = 3$. The middle two lines are the UCLs with $\eta = 2$ and $\eta = 1$ respectively. Based on the control chart, near base load outliers are detected and analyzed for energy saving purpose.

Table 21 listed 12 near base load outliers identified by the control chart with $\eta = 3$. Among these outliers, 10 outliers (highlighted in grey color) is using more energy than expected, i.e. the predicted near base load is significantly smaller than the observed near base load, which may indicate an abnormal energy consumption and energy saving opportunity. Near base load is usually caused by hotel equipments running between the midnight and early morning. Abnormal high near base load usually indicates abuse of some midnight-running equipments, such as water pumps, lighting, air conditioning and so on, which is required to be fixed and avoid energy losses. When $\eta = 2$, among these 19 near base load outliers, 14 outliers are potential candidates of

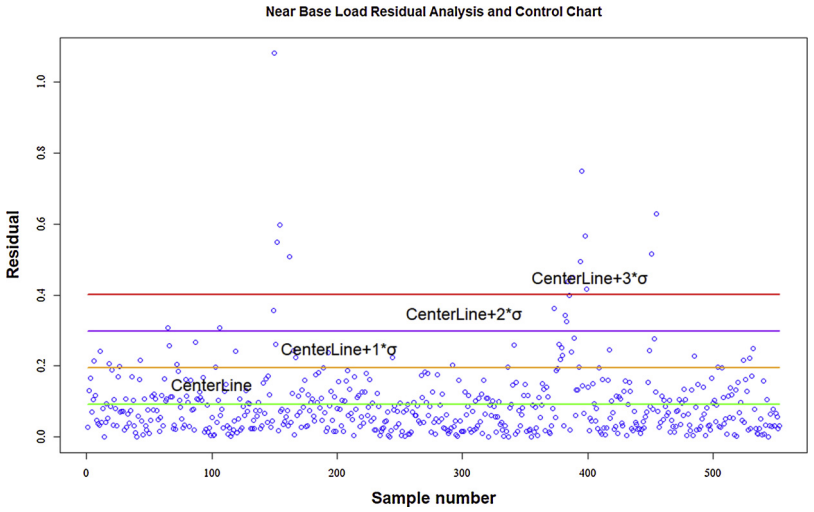


Fig. 11. Hotel near base load residual analysis and corresponding control chart boundaries.

Table 21
Hotel near base load control chart identified 12 outliers with $\eta = 3$.

Date	Near Base Load	Predicted Near Base Load	Residual
23-08-2016	135.7	282.3	146.6
25-08-2016	318	143.2	-174.8
27-08-2016	365.2	146.8	-218.4
04-09-2016	274.2	134.8	-139.4
24-06-2017	303.9	171	-132.8
28-06-2017	286.9	157.9	-129
04-07-2017	311.2	157.3	-153.8
05-07-2017	130	227.3	97.3
08-07-2017	425.7	184.6	-241.1
09-07-2017	382.4	222.9	-159.4
02-09-2017	283.7	137.6	-146.1
06-09-2017	370.2	137.8	-232.4

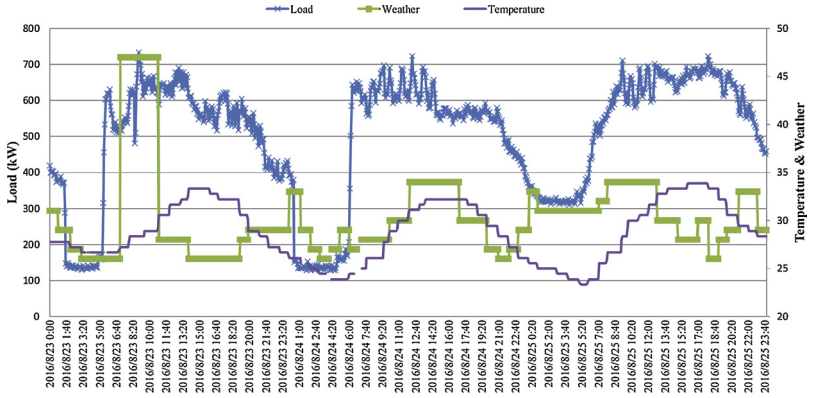


Fig. 12. Hotel's load, temperature and weather code time series from 2016/8/23 to 2016/8/25.

excessive energy consumption. When $\eta = 1$, among these 52 near base load outliers, 25 outliers are potential candidates of excessive energy consumption. Selection of the appropriate value η will be determined by verifying these potential candidates.

Fig. 12 shows that base load of 2016/8/25 is significantly higher than that of 2016/8/23 and 2016/8/24. However the outside temperature and weather didn't vary too much across these three days, which indicated potential unnecessary excessive energy consumption during the midnight and early morning. The near base load outlier in 2016/8/25 was verified by the hotel energy management crew, where they found that some of the air conditioning equipments and chillers weren't turned off or under low-load operation from 0 AM to 5 AM.

The above two control chart case studies proved that true abnormal energy consumption could be accurately identified with appropriate upper control limit. Similar control charts for near peak load, rise time and fall time can be constructed in the same way. These control charts will form a solid foundation to monitor building load profiles and provide energy-saving decision support for the energy management crew.

6. Conclusions

This paper presented a systematic way of quantifying daily building load patterns and identifying abnormal energy consumption. The proposed framework is composed of three major stages: the first step is to use simple and efficient algorithms to preprocess the high-frequency building load time series into a set of meaningful daily profiles. Secondly, prediction models of these profiles are built by selecting appropriate data mining algorithms and predictors. Prediction models are trained tested on historical data sets, and the best ones are selected based on predefined performance metrics. Thirdly, residuals of selected prediction models are analyzed by statistical quality control theory, and for each load profile, a control chart with appropriate upper control limit is constructed. Control charts are then deployed to monitor the daily load profiles and identity abnormal energy consumption, and help energy management crew locating energy saving opportunities. Computational experiments with real-world building load and weather data proves effectiveness of the proposed framework and algorithms. The identified abnormal energy consumption is further verified by field investigation. Our proposed framework is easy to implement in existing building energy management systems and doesn't require sophisticated sub-metering system.

However, due to limited paper space and time-span of building time series data as well as unavailability of tenant related information, this paper didn't discuss how to incorporate holiday, weather weekday/weekend and tenant information into the prediction modeling process. Future research could focus on this direction so that some false alarms of energy consumption caused by weather, holiday, unexpected arrival of many hotel guests could be reduced. Other machine learning techniques, such as ensemble, bagging and boosting could also be tried to improve the prediction model's stability and accuracy.

Acknowledgment

This research has been partially supported by the National Natural Science Foundation of China, Grant No. 71834003, 71573121 and 71001050 and Research Grants Council, University Grants Committee, Hong Kong, General Research Fund: 11272216.

Appendix A. Supplementary data

Supplementary material related to this article can be found, in the online version, at doi:https://doi.org/10.1016/j.scs.2019.101587.

References

Ahmad, M. W., Mourshed, M., & Rezgui, Y. (2017). Trees vs neurons: Comparison between random forest and ANN for high-resolution prediction of building energy consumption. *Energy and Buildings, 147*, 77–89.

Amasyali, K., & El-Gohary, N. M. (2018). A review of data-driven building energy consumption prediction studies. *Renewable and Sustainable Energy Reviews, 81*, 1192–1205.

Araya, D. B., Grolinger, K., ElYamany, H. F., Capretz, M. A. M., & Bitsuamlak, G. (2017). An ensemble learning framework for anomaly detection in building energy consumption. *Energy and Buildings, 144*, 191–206.

Capozzoli, A., Piscitelli, M. S., Brandi, S., Grassi, D., & Chicco, G. (2018). Automated load pattern learning and anomaly detection for enhancing energy management in smart buildings. *Energy, 152*, 336–352.

Camm, J. D., Cochran, J. J., Fry, M. J., Ohlmann, J. W., Anderson, D. R., Sweeney, D. J., et al. (2015). *Essentials of business analytics* (1st edition). Cengage Learning.

Chen, Y., & Wen, J. (2017). A whole building fault detection using weather based pattern matching and feature based PCA method. *2017 IEEE international conference on big data (big data 2017), December 11–14*.

Chou, J., Telaga, A. S., Chong, W. K., & Gibson, G. E., Jr (2017). Early-warning application for real-time detection of energy consumption anomalies in buildings. *Journal of Cleaner Production, 149*, 711–722.

Chou, J., & Telaga, A. S. (2014). Real-time detection of anomalous power consumption. *Renewable and Sustainable Energy Reviews, 33*, 400–411.

Deng, H., Fannon, D., & Eckelman, M. J. (2018). Predictive modeling for US commercial building energy use: A comparison of existing statistical and machine learning algorithms using CBECS microdata. *Energy and Buildings, 163*, 34–43.

Edwards, R. E., New, J., & Parker, L. E. (2012). Predicting future hourly residential electrical consumption: A machine learning case study. *Energy and Buildings, 49*, 591–603.

He, X., Zhang, Z., & Kusiak, A. (2014). Performance optimization of HVAC systems with computational intelligence algorithms. *Energy and Buildings, 81*(1), 371–380.

Kang, L., & Albin, S. (2000). On-line monitoring when the process yields a linear profile. *Journal of Quality Technology, 32*(4), 418–426.

Kapetanakis, D.-S., Mangina, E., Ridouane, E. H., Kouramas, K., & Finn, D. (2015). Selection of input variables for a thermal load prediction model. *Energy Procedia, 78*, 3001–3006.

Kusiak, A. (2015). Break through with big data. *Industrial Engineer, 47*(3), 38–42.

Kusiak, A. (2016). Share data on wind energy. *Nature, 529*(7584), 19–21.

Kusiak, A. (2017). Smart manufacturing must embrace big data. *Nature, 544*(7648), 23–25.

Kusiak, A., Zheng, H., & Song, Z. (2009). On-line monitoring of power curves. *Renewable Energy, 34*(6), 1487–1493.

Kylili, A., & Fokaides, P. A. (2015). European smart cities: The role of zero energy buildings. *Sustainable Cities and Society, 15*, 86–95.

Lin, M., Afshari, A., & Azar, E. (2018). A data-driven analysis of building energy use with emphasis on operation and maintenance: A case study from the UAE. *Journal of Cleaner Production, 192*, 169–178.

Long, H., Wang, L., Zhang, Z., Song, Z., & Xu, J. (2015). Data-driven wind turbine power generation performance monitoring. *IEEE Transactions on Industrial Electronics, 62*(10), 6627–6635.

Mathieu, J. L., Price, P. N., Kiliccote, S., & Piette, M. A. (2011). Quantifying changes in building electricity use, with application to demand response. *IEEE Transactions on Smart Grid, 2*(3), 507–518.

Miller, C., Nagy, Z., & Schlueter, A. (2018). A review of unsupervised statistical learning and visual analytics techniques applied to performance analysis of non-residential buildings. *Renewable and Sustainable Energy Reviews, 81*, 1365–1377.

Miller, C., Nagy, Z., & Schlueter, A. (2015). Automated daily pattern filtering of measured building performance data. *Automation in Construction, 49*, 1–17.

Montgomery, D. C. (2005). *Introduction to statistical quality control* (5th ed.). New York: John Wiley.

Palensky, P., & Dietrich, D. (2011). Demand side management: Demand response, intelligent energy systems, and smart loads. *IEEE Transactions on Industrial Informatics, 7*(3), 381–388.

Peña, M., Biscarri, F., Guerrero, J. I., Monedero, I., & León, C. (2016). Rule-based system to detect energy efficiency anomalies in smart buildings, a data mining approach. *Expert Systems with Applications, 56*, 242–255.

Ploennigs, J., Chen, B., Schumann, A., & Brady, N. (2013). *Exploiting generalized additive models for diagnosing abnormal energy use in buildings. 5th ACM workshop on embedded systems for energy-efficient buildings* pp.17:1–17:8.

Robinson, C., Dilkina, B., Hubbs, J., Zhang, W., Guhathakurta, S., Brown, M. A., et al. (2017). Machine learning approaches for estimating commercial building energy consumption. *Applied Energy, 208*, 889–904.

Tan, P. N., Steinbach, M., & Kumar, V. (2006). *Introduction to data mining.* Boston, MA: Pearson Education/Addison Wesley.

Tang, F., Kusiak, A., & Wei, X. (2014). Modeling and short-term prediction of HVAC system with a clustering algorithm. *Energy and Buildings, 82*(1), 310–321.

Touzani, S., Granderson, J., & Fernandes, S. (2018). Gradient boosting machine for modeling the energy consumption of commercial buildings. *Energy and Buildings, 158*, 1533–1543.

Wang, Y., Chen, Q., Hong, T., & Kang, C. (2019). Review of smart meter data analytics: Applications, methodologies and challenges. *IEEE transactions on smart grid, 10*(3), 3125–3148.

Wang, Z., Wang, Y., Zeng, R., Srinivasan, R. S., & Ahrentzen, S. (2018). Random forest based hourly building energy prediction. *Energy and Buildings, 171*, 11–25.

Wei, Y., Zhang, X., Shi, Y., Xia, L., Pan, S., Wu, J., et al. (2018). A review of data-driven approaches for prediction and classification of building energy consumption. *Renewable and Sustainable Energy Reviews, 82*, 1027–1047.

Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). *Data mining: Practical machine learning tools and techniques* (4th ed.). San Francisco, CA: Morgan Kaufmann.

Zeng, Y., Zhang, Z., & Kusiak, A. (2015). Predictive modeling and optimization of a multi-zone HVAC system with data mining and firefly algorithms. *Energy, 86*(1), 393–402.

Zoritaa, A. L., Fernández-Tempranob, M. A., García-Escuderob, L.-A., & Duque-Pereza, O. (2016). A statistical modeling approach to detect anomalies in energetic efficiency of buildings. *Energy and Buildings, 110*, 377–386.