Energy & Buildings 285 (2023) 112896

Contents lists available at ScienceDirect

Energy & Buildings

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

Impacts of building load dispersion level on its load forecasting accuracy: Data or algorithms? Importance of reliability and interpretability in machine learning

Maomao Hu [a,b], Bruce Stephen [c], Jethro Browell [d], Stephen Haben [e], David C.H. Wallom [a,*]

[a] Oxford e-Research Centre, Department of Engineering Science, University of Oxford, Oxford OX1 3QG, United Kingdom
[b] Department of Energy Science and Engineering, Stanford University, Stanford, CA 94305, United States
[c] Institute for Energy and Environment, Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow G1 1RD, United Kingdom
[d] School of Mathematics and Statistics, University of Glasgow, United Kingdom
[e] Mathematical Institute, University of Oxford, Oxford OX2 6GG, United Kingdom

ARTICLE INFO

Article history:
Received 13 December 2022
Revised 5 February 2023
Accepted 12 February 2023
Available online 15 February 2023

Keywords:
Building load forecasting
Machine learning
Deep learning
Long short term memory
Reliability and interpretability

ABSTRACT

Data-driven forecasting techniques have been widely used for building load forecasting due to their accuracy and wide availability of operational data. Recent advances have been underpinned by the increased capability of machine learning (ML) algorithms; however, most studies only tested ML techniques on a single or a small number of buildings over short periods, lacking reliable tests. Moreover, few studies focused on the effects of characteristics of building load profiles on forecast accuracy, lacking the interpretation of ML-based prediction results. In this study, we investigate the impacts of building load dispersion level on its best load forecasting accuracy, which is obtained by comparing the forecasting performances of 11 prediction models over 9 weeks for 56 British non-domestic buildings. We find that conventional shallow ML models still outperform the increasingly popular deep learning models for time-series load forecasting, and ensemble learning can help improve forecast accuracy by integrating diverse individual models. We demonstrate that each building's best forecasting performance is largely influenced by the load dispersion level. In practice, the proposed dispersion metrics are recommended to quantify load dispersion levels before model development. For a building with a low dispersion level, the simple persistence model has satisfactory performance and could be directly used for design, control, and fault diagnosis of building energy systems for energy efficiency and energy flexibility.

1. Introduction

Building operations and construction account for 35 % of total final energy consumption and 38 % of total energy-related carbon dioxide (CO_2) emissions worldwide in 2019 [1]. Electricity consumption in building operations is responsible for nearly 55 % of global electricity consumption [2] and 59 % of electricity consumption in the UK [3]. To reduce energy consumption and CO_2 emissions, energy efficiency measures are needed for buildings throughout all phases, from design, through construction and operations to maintenance. Accurate building load forecasting is essential to support decisions that impact energy efficiency, including design [4], management & control [5,6], and fault detection & diagnosis (FDD) [7]. It can also assist in power management by improving understanding of the balance between demand and supply to improve the power reliability [8,9].

Existing approaches for building load forecasting can be classified into three categories: physics-based (white-box), purely data-driven (black-box), and hybrid (grey-box), i.e., partially data-driven approaches [6]. The development of physics-based models requires extensive domain expertise and in-depth understanding of building thermal dynamics and building energy systems. Detailed information about the building to be studied is needed for model inputs, such as thermophysical and geometric parameters of building envelopes, which are difficult to obtain even for newly constructed buildings [5]. Physics-based modelling becomes more labour-intensive, computationally inefficient, and time-consuming when the scale and complexity of buildings increase [10]. With the wide availability of operational data in today's buildings, data-driven modelling approaches, especially black-box models, have attracted increasing research interests in the field of load forecasting for

[*] Corresponding author.
E-mail address: david.wallom@oerc.ox.ac.uk (D.C.H. Wallom).

https://doi.org/10.1016/j.enbuild.2023.112896
0378-7788/© 2023 The Author(s). Published by Elsevier B.V.

buildings [7]. Two major advantages of black-box models are their simplicity and ease of automation. In particular, the process of black-box modelling doesn't require any physical building information. Instead, they use statistical tools and ML techniques to produce predictions based on historical operational data, which can be fully automated and readily implemented for engineering use [11].

In addition to the increased data availability in buildings, the rise of artificial intelligence and machine learning (ML) techniques is another driving force behind the success of the data-driven building load forecasting [7,12,13]. In response, a considerable amount of literature has been published on ML-based building load forecasting over the last decade. In general, the ML techniques for building load forecasting can be categorized into *conventional ML techniques* [7,10,12] and *deep learning techniques* [14,15]. A deep learning architecture consists of a series of simple learning modules. In each module, the input is transformed in a linear or nonlinear manner to improve the selectivity and invariance of the representation [16]. In contrast, conventional ML techniques adopt 'shallow' architectures and input data are only transformed once or twice [14].

1.1. Conventional ML techniques

Artificial Neural Networks (ANNs), Support Vector Regression (SVR), and Decision Trees (DTs) are common and powerful conventional ML approaches that have been widely used in the area of building load forecasting due to their capability of handling complex and nonlinear relationships [17]. ANNs and SVR were reported to represent 47 % and 25 % of the total studies on data-driven building load forecasting, respectively [11]. Seyedzadeh et al. [18] and Ahmad et al. [17] have comprehensively reviewed the applications of ANNs and SVR for building electricity use forecasting. Several studies have focused on comparing the prediction performance of the SVR model with different ANN models [19–23], and in general SVR model was found to have a higher prediction accuracy than ANN models.

DTs use a tree-like structure to classify historical data into various target classes (i.e., classification problems) or continuous values (i.e., regression problems). One of the most popular decision tree-based techniques for building load forecasting is random forest (RF) [24]. Its prediction performance has been compared with other algorithms, including ANNs [25,26] and SVR [24,27,28]. In recent years, there has been an increasing interest in the combination of DTs and gradient boosting techniques, including extreme gradient boosting trees (XGB, first proposed in 2016) [29], light gradient boosting machine (LightGBM, first proposed in 2017) [30], and categorical boosting (CatBoost, first proposed in 2017) [31]. In the ASHRAE Great Energy Predictor III competition [32], gradient boosted decision trees outperformed other prediction algorithms and dominated the energy prediction competition.

1.2. Deep learning techniques

As defined by LeCun et al. in 2015 [16], "deep learning is a technique which allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction." As opposed to conventional ANNs ('shallow' neural networks), which typically have 1 hidden layer together with 1 input layer and 1 output layer, deep learning models could have a number of hidden layers, where each layer transforms the representation from one level (starting with the initial raw input) to a higher and more abstract level. In the area of deep learning, recurrent neural networks (RNNs) and convolutional neural networks (CNNs) are two of the most widely used deep neural networks. RNNs have been successfully applied to natural language processing and speech recognition due to their ability to handle sequential inputs, while CNNs gain many successes in the field of computer vision due to their good performance in feature extraction.

Among different RNN architectures, the long short-term memory (LSTM) network [33] has been recognized as the most successful since it can learn long-term temporal correlations compared with conventional RNNs. Several studies have applied LSTM to predict the building energy consumption [34–36]. Gao et al. [37] even integrated the LSTM model into transfer learning frameworks to increase the prediction performance for buildings with poor information. To integrate the ability of CNNs in terms of feature extraction, some studies have attempted to combine CNNs with LSTM, forming hybrid CNN-LSTM models [15,38,39]. Fan et al. developed a transfer learning methodology based on the CNN-LSTM model structure to improve the load forecast accuracy for buildings with limited observation data [40].

1.3. Interpretability of machine learning techniques

Although ML techniques have been widely used for building load forecasting and other fields, it is of great challenge for end users to understand and trust them due to their "black-boxness" [41,42]. Therefore, researchers have shown an increasing interest in the interpretability of machine learning in recent years. Interpretable ML techniques can be classified into ante-hoc and post-hoc approaches according to when the interpretable measures are adopted [43]. Ante-hoc and post-hoc interpretable techniques are applied during and after the model training process, respectively. For ante-hoc interpretable techniques in building load forecasting, Gao and Ruan [44] integrated the attention mechanism into encoder and decoder models based on LSTM to improve the interpretability of deep learning models. Test results for an office building showed that the features of daily maximum temperature, mean temperature, minimum temperature, and dew point temperature were the most important features in terms of the improvement of interpretability. Li and Xiao et al. [45] applied the attention mechanism to RNNs for a 24-hour ahead building cooling load prediction. Results showed that the RNNs with attention mechanisms can improve prediction accuracy and interpretability compared with the RNNs without attention mechanisms. For post-hoc interpretable techniques, Local Interpretable Model-agnostic Explanations (LIME) [46] and Shapley Additive exPlanations (SHAP) [47] are the most commonly-used model-agnostic tools to interpret individual predictions. Wastensteiner et al. [48] applied LIME to interpret the classification results of time-series building energy consumption data and evaluated the reliability of the interpretation. Zdravkovic et al. [49] employed LIME to generate the feature importances for the local forecasts for the district heating demand. Jin et al. [50] developed an interpretable building energy benchmarking framework based on the LIME method to make the results more understandable. Chang and Li. et al. [51] used the SHAP method to improve the interpretability of PV power generation models, including XGBoost, SVR, and LSTM, by measuring the feature importance. Results showed that global horizontal irradiance was the most influential feature based on SHAP values. Bellahsen and Dagdougui [52] employed SHAP to rank the importance of various features for the building-level and district-level electrical load predictions. Results indicated that the historical load right before the present forecasting time was the most important feature.

1.4. Research questions, objectives, and contributions

Although conventional ML techniques and deep learning techniques for building load forecasting have been extensively studied, two major issues remain to be addressed: 1) Inconclusive forecast evaluation and benchmarking involving ML techniques. Most existing studies tested the performance of prediction algorithms on a single or a small number of buildings over short periods. In the building load forecasting area, a technique is desired to have a reliable and robust performance under different weather conditions and occupants' behaviours for different buildings. Moreover, some studies claimed they developed a more accurate model over simple benchmarks but lacked justifying the increase in model complexity. 2) A lack of interpretation of the prediction results achieved by ML techniques from the perspective of the building load characteristics. The prior studies on interpretable ML techniques improved the interpretability by improving the structure of the individual models during the model training process [44,45] or measuring the feature importance after training [48–52]. Few studies have tried to explain the prediction performance of ML techniques based on the characteristics of building load profiles. In summary, the essential research questions of this study are:

- *What algorithms? Specifically, what algorithms are more likely to outperform others in the field of building load forecasting? Do deep learning models outperform conventional machine learning models? What model is the feasible model for benchmarking?*
- *Why that building? Specifically, why does Building X have higher forecast accuracy than Building Y? what is the major factor contributing to the high forecast accuracy of Building X? Data (*i.e., *load profile dispersion) or algorithms? Moreover, how we can quantify the influential load profile dispersion level?*

To bridge the research gaps and improve the reliability and interpretability of ML techniques, we conduct a comparative study on building load forecasting, in which 56 British non-domestic buildings with different primary usage types are used to test the performances of 11 different prediction models with different mechanisms over a time period of 9 weeks. Following the performance comparison, we investigate the influences of building load dispersion level on its best forecasting performance using correlation analysis. In summary, our paper makes the following contributions:

1) We propose novel metrics to quantify the dispersion level of building intraday load profiles based on a new load visualization method. The new load visualization method enables us to quickly capture some intuitive and useful insights, including intraday load shape, intraweek load discrepancy, and intraday load dispersion.
2) By a large-scale and long-period comparative study, we find that conventional ML models still outperform deep learning models, despite their increasing application for building load forecasting in recent years. Ensemble learning can help to improve the forecast accuracy and provide better forecasting performance than using individual base models alone. We also demonstrate that the naive persistent model (same-day-previous-week) is not naive at all but has close forecast accuracy with some ML models after a long period of testing. It is feasible to use it as the benchmarking model in the building load forecasting problem due to its simplicity and computational efficiency.
3) We find that the best achievable forecasting performance is largely influenced by its load dispersion level by correlation analysis. Buildings with higher dispersion levels are more likely to have lower forecast accuracy.

The paper is organized as follows: Section 3 presents the full methodology, beginning with an overview of the proposed approach (Section 3.1), development of load dispersion metrics based on a better load profile visualization method (Section 3.2), descriptions and choices of different categories of prediction techniques (Section 3.3), and correlation analysis to be used (Section 3.4). In Section 3, the datasets of target buildings are first introduced and visualized based on the proposed approach. The prediction performance comparison and correlation analysis are then performed. In Section 4, the in-depth discussion is presented based on the results in Section 3. Finally, concluding remarks are provided in Section 5.

2. Methodology

2.1. Overview of the proposed approach

As shown in Fig. 1, to investigate the effects of load dispersion level on load forecast accuracy, the proposed approach consists of three steps. In Step 1, a new approach to visualizing building load profiles is first proposed to gain some intuitive insights directly from the plots. Based on the newly proposed visualization method, we then develop new metrics to quantify the building intraday load dispersion for the correlation analysis. In Step 2, different categories of prediction techniques are developed and applied to a 24-hour ahead short-term load forecasting at time intervals of 30 mins, including naive persistence models, conventional ML models, deep learning models, and an ensemble model. In Step 3, the performances of these previously listed models are evaluated and compared. Finally, a correlation analysis is conducted to investigate the effects of building load dispersion level on its load forecast accuracy.

2.2. Load dispersion quantification based on a new visualization method

2.2.1. Load profile visualization

Data visualization is a crucial step of the data analytics process [53]. Its goal is to transform data into a visual context to provide the data users with a better understanding of information, such as clear patterns and relationships, trends, and outliers. In the domain of energy data analytics, initial data visualization before further data analytics such as energy forecasting is desirable and necessary to provide hidden insights into energy use behaviours and lifestyles of customers. In this study, we propose an approach to improving the visualisation of building electricity consumption data to gain some intuitive insights directly from the plots.

The time-series electricity consumption data of two example buildings (Building #1 and Building #2) are plotted against the time of the year in Fig. 2-a and Fig. 2-c, respectively. It can be found that both buildings have a dynamic energy consumption pattern throughout the whole year, and the patterns are different from each other. However, no other useful information such as frequent trends and patterns can be found from these representations. To explore the intraday load patterns, the electricity consumption data of Building #1 and Building #2 are plotted against the time of the day in Fig. 2-b and Fig. 2-d, respectively. By plotting intraday load profiles together, we can observe the load trend over a day. To make the intraday trend clearer, 5 different levels of percentiles (i.e., 5th, 25th, 50th, 75th, and 95th percentiles) are used to illustrate the distributions of observations. The 5th percentile is, for example, the score below which 5 % of the observations may be found. To show the load pattern discrepancy between weekdays and weekends, the average load profiles on weekdays and at weekends are plotted together with the percentiles. In summary, the following insights can be captured directly from the proposed plots (Fig. 2-b and Fig. 2-d):

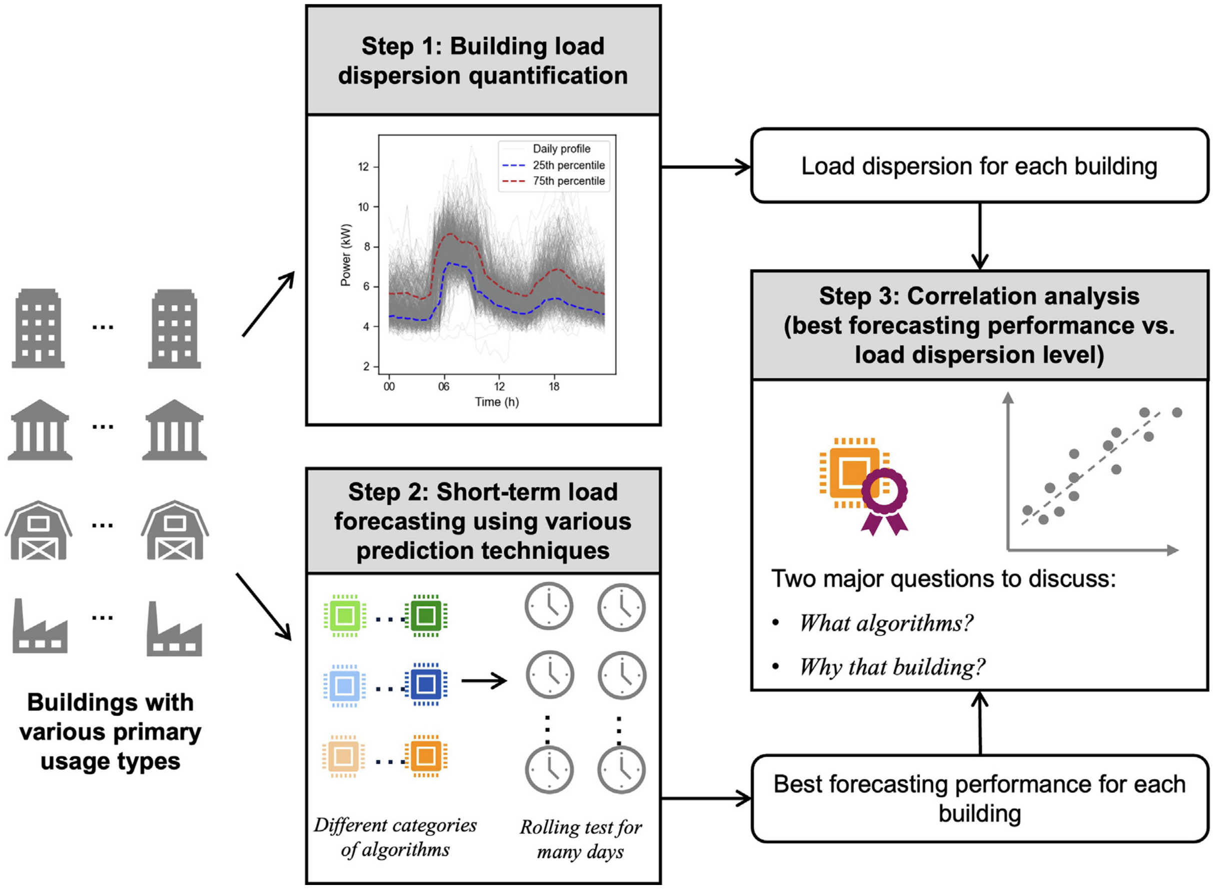

Step 1: Building load dispersion quantification
Daily profile
25th percentile
75th percentile
Power (kW)
Time (h)
00
06
12
18
2
4
6
8
10
12
Load dispersion for each building
Buildings with various primary usage types
Step 2: Short-term load forecasting using various prediction techniques
Different categories of algorithms
Rolling test for many days
Step 3: Correlation analysis (best forecasting performance vs. load dispersion level)
Two major questions to discuss:
• What algorithms?
• Why that building?
Best forecasting performance for each building

Fig. 1. Overview of the proposed approach.

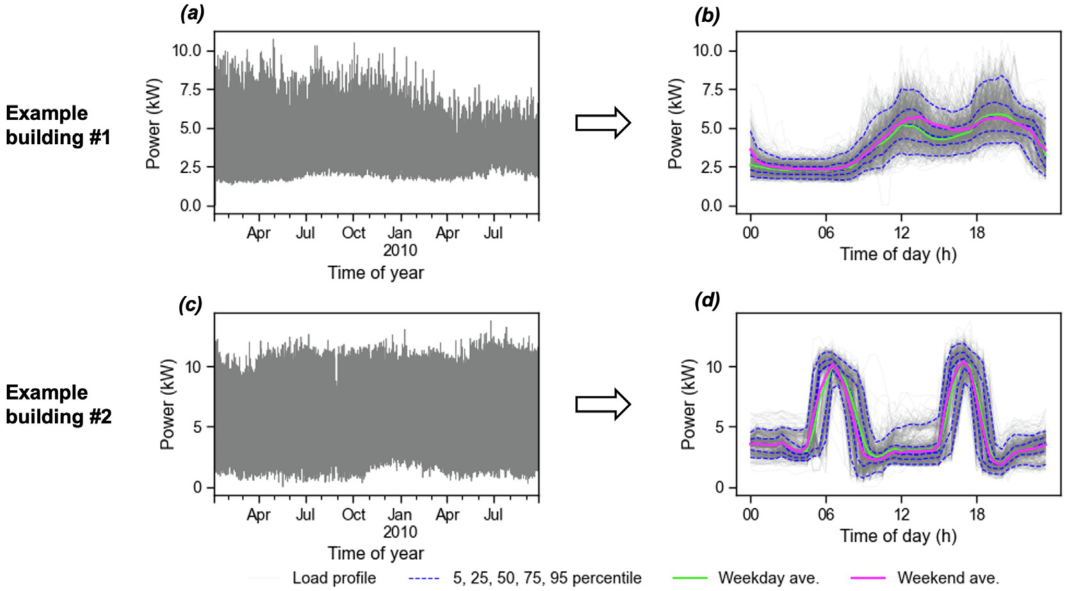


Fig. 2. Two examples of plotting building load profiles: (a and c) building load profile against the time of the year, and (b and d) building load profile against the time of the day with percentiles plotted.

1) ***Intraday load shape***: by looking at the percentiles, we can quickly and clearly identify the intraday trends and the load peaks during a day. For example, compared with Building #1 which has one noon peak and one evening peak, Building #2 has a morning peak and an evening peak.

2) ***Intraweek load discrepancy***: by comparing the average load profiles on weekdays and at weekends, we can know whether the building occupiers have similar energy use behaviours throughout the week. For example, Building #1 has a larger load pattern discrepancy between weekdays

and weekends than Building #2. This may also allow some inference to the types of occupants and their energy-related behaviours.

3) ***Intraday load dispersion***: Apart from the intraday load shape, the percentile set can also help indicate the dispersion level of the load profiles, i.e., widely or densely distributed. The load dispersion level has two dimensions: dispersion at a specific time (e.g., hourly and sub-hourly) and dispersion over a whole day. For the dispersion at a specific time, the load profiles of Building #1 at 18:00 are more dispersed than the profiles at 06:00. For the overall dispersion, Building #2 has a lower overall relative dispersion than Building #1, since the percentiles of Building #2 overlap and are indistinguishable during some periods, and they are less widespread than Building #1.

2.2.2. Quantifying load dispersion based on visualization

Based on the newly proposed visualization method for building energy load data, the intraday load dispersion level can be observed. To support correlation analysis, the dispersion level needs to be quantified. Two metrics are therefore proposed in this subsection to quantify the intraday dispersion level of building energy load.

Fig. 3 illustrates the scheme of quantifying the load dispersions at a specific time and over the whole day. The load dispersion at a specific time t can be determined using the coefficient of variation, CV_t, and the coefficient of quartile variation, CQV_t, as shown in Eq. (1) and Eq.(2), respectively. CV_t is a standardized measure of the dispersion of a probability distribution; CQV_t is a measure of relative dispersion that is based on interquartile range, i.e., $Q_{3,t} - Q_{1,t}$. For the load dispersion over a whole day, it can be described by $CV_{overall}$ in Eq. (3) and $CQV_{overall}$ in Eq. (4), respectively, in which N is the number of the time slots within one day. Note that CV is useful for comparing datasets with different units or scales of measurement, while CQV is more useful when comparing datasets with extreme outliers.

$$CV_t = \frac{\sigma_t}{\mu_t} \tag{1}$$

$$CQV_t = \frac{Q_{3,t} - Q_{1,t}}{Q_{3,t} + Q_{1,t}} \tag{2}$$

$$CV_{overall} = \frac{1}{N}\sum_{t=0}^{N} CV_t \tag{3}$$

$$CQV_{overall} = \frac{1}{N}\sum_{t=0}^{N} CQV_t \tag{4}$$

2.3. Prediction techniques

2.3.1. Individual models

Different categories of prediction techniques are analysed in this study to predict building load profiles, including two naive persistence models, five conventional ML models (ANNs, SVR, RF, XGB, and LightGBM), and three deep learning models (classic LSTM model, LSTM encoder-decoder model, and hybrid CNN-LSTM model). In addition, ensemble learning techniques are used to integrate the performances of the individual base models. The detailed description of the prediction models is as follows:

- **Naive models**. Two naive models are used as benchmarking models: 1) to use the load profile of the previous day as the forecasted profile (noted as 'previous-day'), and 2) to use the load profile of the same day in the previous week as the forecasted profile (noted as 'same-day-previous-week'). It is worth mentioning that although the two naive models are simple, it

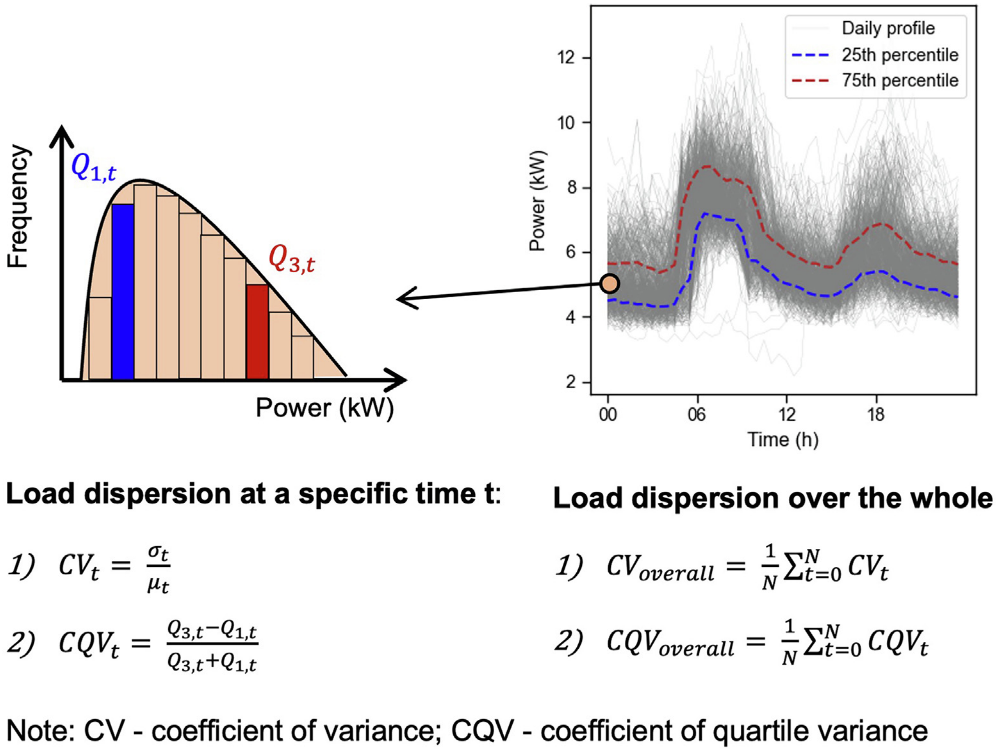

Frequency
$Q_{1,t}$
$Q_{3,t}$
Power (kW)
Daily profile
25th percentile
75th percentile
Power (kW)
12
10
8
6
4
2
00
06
12
18
Time (h)
Load dispersion at a specific time t:
1) $CV_t = \frac{\sigma_t}{\mu_t}$
2) $CQV_t = \frac{Q_{3,t}-Q_{1,t}}{Q_{3,t}+Q_{1,t}}$
Load dispersion over the whole day:
1) $CV_{overall} = \frac{1}{N}\sum_{t=0}^{N} CV_t$
2) $CQV_{overall} = \frac{1}{N}\sum_{t=0}^{N} CQV_t$
Note: CV - coefficient of variance; CQV - coefficient of quartile variance

Fig. 3. Scheme of quantifying load dispersions at a specific time and over the whole day. Each has two metrics: coefficient of variance (CV) and coefficient of quartile variance (CQV).

can be challenging for other prediction techniques to outperform them due to the daily and weekly time-dependent patterns [40].

- **ANNs** [54]. The simplest and most popular feedforward network consists of an input layer, a hidden layer, and an output layer. The training of ANNs is a process of determining the best weights on the inputs, and the backpropagation algorithm is the most common method for computing the error gradient for a feedforward network.
- **SVR** [55]. In contrast to conventional neural networks which are based on empirical risk minimization, SVM is based on the structural risk minimization principle, which aims to minimize an upper bound of the generalization error.
- **Decision tree-based techniques, including RF, XGB, and LightGBM**. RF, first proposed by Breiman in 2001 [56], is an ensemble learning technique consisting of a set of numerous regression trees. Compared with a single regression tree, RF can address instability issues by creating a diverse collection of trees, which are trained through bagging and random input variable selection [24]. Both XGB [29] and LightGBM [30] are extensions to gradient boosting frameworks based on decision trees. The major difference between XGB and LightGBM is that XGB uses the horizontal layer-wise tree growth strategy, and LightGBM uses the vertical leaf-wise tree growth strategy.
- **Classic LSTM model**. The LSTM architecture, an advanced version of RNNs, was first proposed by Hochreiter and Schmidhuber [33] and has received enormous attention due to its competence in processing long sequences than traditional RNNs. As shown in Fig. 4-b, a complete LTSM block consists of a forget gate (f_t), an input gate (i_t), an update gate (u_t), an output gate (o_t) and a memory unit called a cell. The operations of all nodes at time step t can be found in [36,37].
- **LSTM encoder-decoder model.** The encoder-decoder model is also known as the sequence-to-sequence model, which has been successfully used to translate one sequence into another in the field of machine translation. In this study, the LSTM technique is integrated into the encoder-decoder model framework to form the LSTM encoder-decoder model. As shown in Fig. 4-c, the LSTM encoder-decoder model consists of two components: LSTM encoder and LSTM decoder.
- **Hybrid CNN-LSTM model**. CNN is an effective technique for automatic feature extraction, which has been successfully applied to text, speech, and image recognition. To combine the advantages of CNN and LSTM, a hybrid CNN-LSTM model is used in this study to predict energy load profiles. As shown in Fig. 4-d, the architecture of a typical CNN consists of a convolution layer and a pooling layer. A pooling layer is employed to reduce the number of parameters and network computation costs by calculating the maximum value of a given area in a feature map. A flattening layer is added to transform the data into the format accepted by the LSTM layer.

Hyperparameter tuning plays a significant role in the forecast accuracy for both deep learning models and conventional ML models. However, overfitting is more common in deep learning models compared to conventional machine learning models. This is due to the large number of parameters that can be adjusted, increasing the risk of fitting noise into the training data. To address this issue for deep learning models, the dropout technique is adopted in this study, which is a popular regularization technique. It refers to the process of randomly dropping some units based on a certain probability (i.e., dropout ratio) from the neural network during training [57]. In addition to the dropout ratio, other hyperparameters, which have a large influence on the model performance of deep learning models, also need to be optimized. The grid-search technique is used in this study to optimize critical hyperparameters for the three deep learning models based on the training datasets. The search ranges of the hyperparameters in each prediction model when using the grid-search technique will be introduced in Appendix A.

2.3.2. Ensemble model

In addition to the above AI-based models, the ensemble learning technique is adopted in this study which generates the final prediction output by integrating multiple base models. Ensemble models have gained increasing popularity due to the capability of providing improved accuracy and better generalization than individual forecasting models [58]. The major reasons behind this include 1) the single most accurate model might be biased due to inadequate training data; 2) each forecasting model has its strengths and weaknesses, the ensemble learning enables the base models to complement each other [27,58]. In this study, a simple ensemble model is adopted, i.e., averaging the prediction outputs of the above ten base models.

2.3.3. Inputs and output for different categories of models

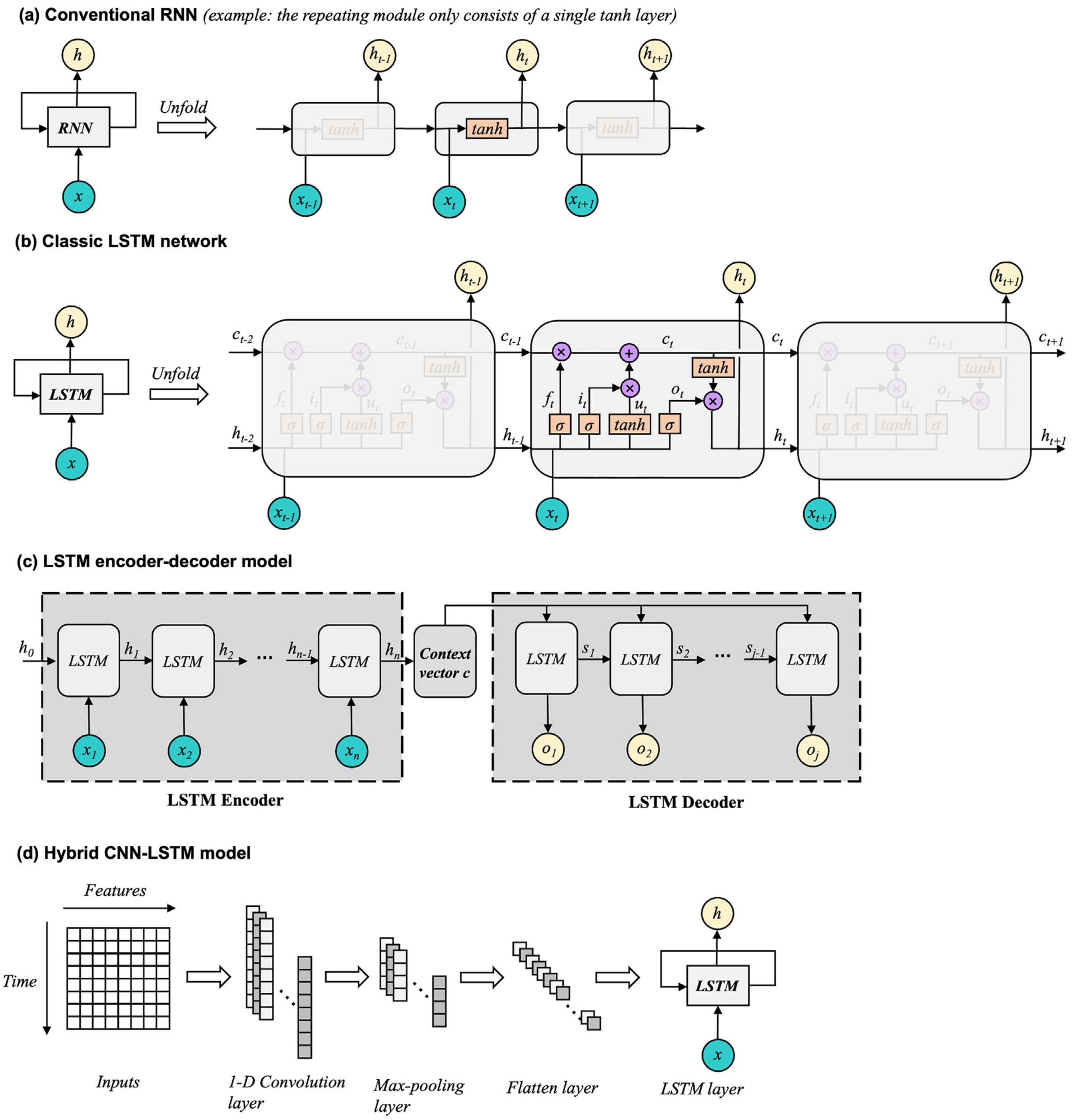
As shown in Fig. 5, different categories of models used in this study have different model inputs. For the two naive models, the input is simply the load profile of the previous day or the load profile of the same day in the previous week. For the five conventional ML models, the inputs include the historical features in the last 7 days, including meter readings and weather conditions, and features in the next 24 h, including predictions of weather conditions and temporal features. The air dry-bulb temperature, air dew point, and wind speed are selected in this study to describe weather conditions since they are likely to influence the building energy consumption behaviour. For the three deep learning models, the inputs include the historical features in the last 7 days, including meter readings, weather conditions, and temporal features, and the features in the next 24 h, including weather conditions and temporal features.

The categorical temporal features consist of the hour (i.e., 00:00 to 23:00), day type (i.e., Monday to Sunday), and month (i.e., January to December), which are used as indicators for seasonality and indoor occupancy. They need to be transformed into numerical formats to be accepted by the deep learning models. One-hot encoding, a commonly used approach, is used in this study to generate the feature matrix for each categorical temporal variable. After one-hot encoding, the columns of the hour, day type, and month matrices are 24, 7, and 12, respectively.

Note that deep learning models in our study have two categories of features with different window sizes. The features, therefore, need to be included in deep neural networks by using two separate input heads (i.e., input matrices) and each head has a corresponding group of temporal features. In contrast, there is only one input head for conventional ML models, and only the temporal features in the next 24 h are adopted as temporal inputs to label the input matrix, which helps to avoid the repeated temporal information, reduce the dimension of the input matrix, and save training time. More details on multi-headed deep learning models will be introduced in Appendix A.

2.3.4. Prediction evaluation metrics

In our study, prediction models are evaluated using a rolling walk-forward test approach. As shown in Fig. 6, it is an approach where a model is used to make a 24-hour ahead load forecasting on a day. After each forecast, the actual observation for that day is made available to the model so that it can be used for predicting the next-day load profile. The rolling walk-forward testing process is how one model may be used in practice and allows the model to make full use of the available historical data.



(a) Conventional RNN *(example: the repeating module only consists of a single tanh layer)*

(b) Classic LSTM network

(c) LSTM encoder-decoder model

(d) Hybrid CNN-LSTM model

Fig. 4. Architectures of conventional RNN and LSTM-based techniques: a) conventional RNN; b) classic LSTM model; c) LSTM encoder-decoder model; d) hybrid CNN-LSTM model.

After the rolling walk-forward test, three different metrics, as defined in Eqs. (5-a) - (5-c), are used in this study to evaluate the overall model prediction performance during the entire test session, including overall mean absolute error ($MAE_{overall}$), overall root mean square error ($RMSE_{overall}$), and overall coefficient of variation of the root mean square error ($CV - RMSE_{overall}$). In the context of the rolling walk-forward test approach, the number of the observation points, N in Eq. (5), is the total number of the observation points during the whole test session (i.e., N = Number of test days × Observation points within one day). Note that MAE and $RMSE$ are scale-dependent metrics, while $CV - RMSE$ is a scale-independent one, which is feasible for the performance comparisons among different buildings. When the $CV - RMSE$ is around 30 %, it means the developed model is calibrated and acceptable for engineering purposes [14,59].

$$MAE_{overall} = \frac{\sum_{k=1}^{N} |y_k - \hat{y}_k|}{N} \tag{5-a}$$

$$RMSE_{overall} = \sqrt{\frac{\sum_{k=1}^{N} (y_k - \hat{y}_k)^2}{N}} \tag{5-b}$$

$$CV - RMSE_{overall} = \sqrt{\frac{\sum_{k=1}^{N} (y_k - \hat{y}_k)^2}{N}} \Big/ \frac{\sum_{k=1}^{N} y_k}{N} \tag{5-c}$$

2.4. Correlation analysis: Best forecasting performance vs Load dispersion level

The major goal of our study is to investigate the effects of load dispersion level on its best load forecasting performance. The dimen-

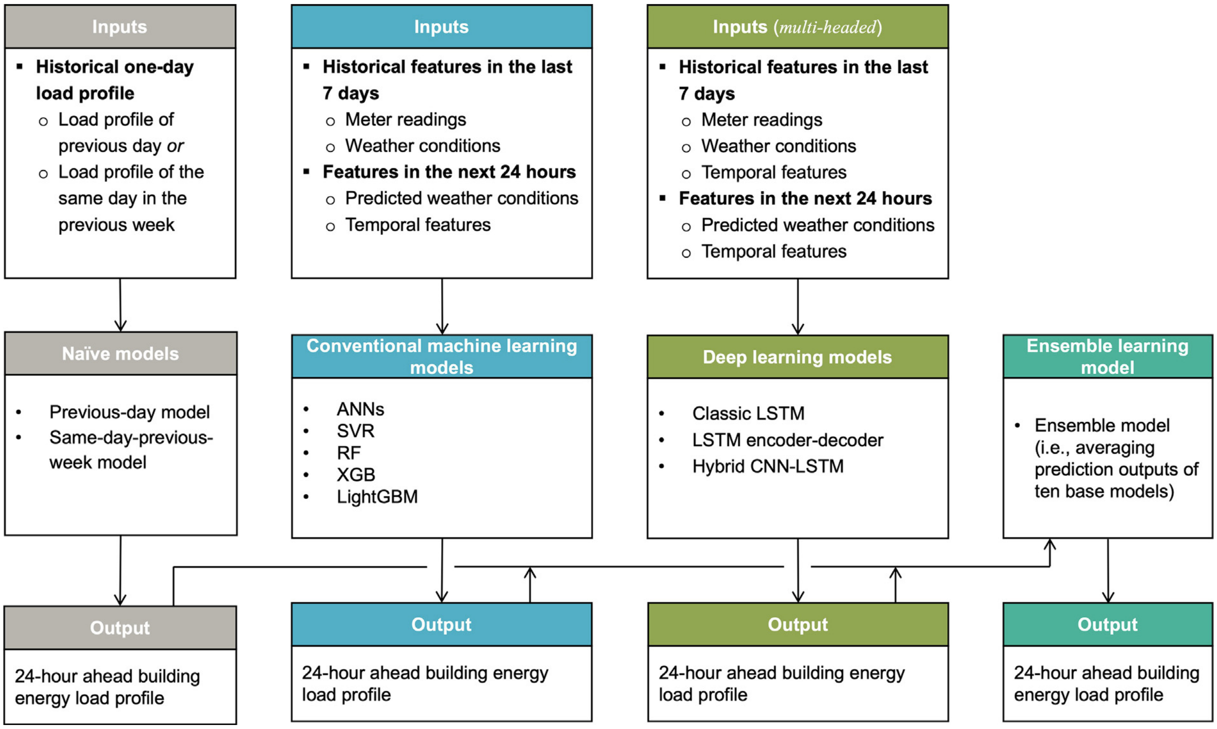

Inputs
Historical one-day load profile
Load profile of previous day or
Load profile of the same day in the previous week
Inputs
Historical features in the last 7 days
Meter readings
Weather conditions
Features in the next 24 hours
Predicted weather conditions
Temporal features
Inputs (multi-headed)
Historical features in the last 7 days
Meter readings
Weather conditions
Temporal features
Features in the next 24 hours
Predicted weather conditions
Temporal features
Naïve models
Previous-day model
Same-day-previous-week model
Conventional machine learning models
ANNs
SVR
RF
XGB
LightGBM
Deep learning models
Classic LSTM
LSTM encoder-decoder
Hybrid CNN-LSTM
Ensemble learning model
Ensemble model (i.e., averaging prediction outputs of ten base models)
Output
24-hour ahead building energy load profile
Output
24-hour ahead building energy load profile
Output
24-hour ahead building energy load profile
Output
24-hour ahead building energy load profile

Fig. 5. Comparison of model inputs for different categories of models.

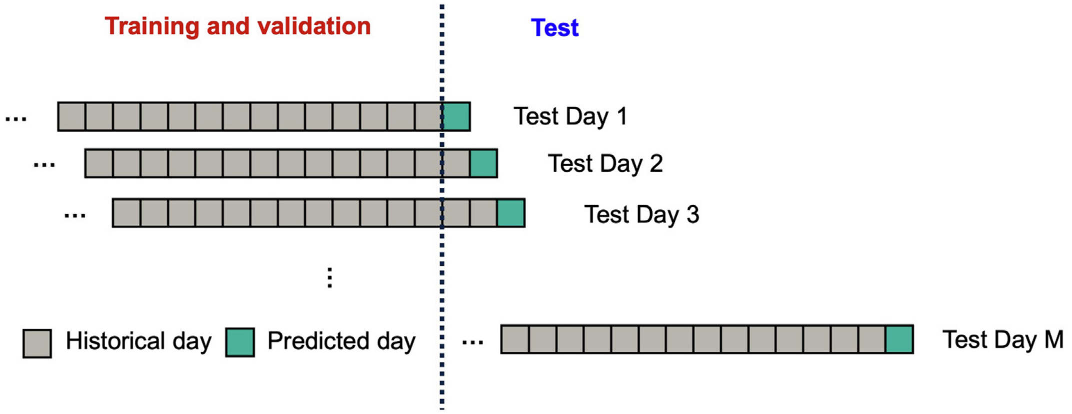


Fig. 6. Schematic diagram of rolling walk-forward testing process for prediction models.

sionless metric of $CV-RMSE$ is used to quantify the forecast accuracy in the correlation analysis for the building-to-building comparison. To achieve that goal, the commonly used Pearson correlation coefficient is used in this study to measure the correlation between the best achievable forecasting performance (i.e., $CV-RMSE_i^*$, the minimum value of $CV-RMSE$ achieved by all models) and the intraday load dispersion level (i.e., $CV_{overall}$ in Eq. (3)). The processing of the correlation analysis is given by Eqs. (6-a) and (6-b).

$$CV-RMSE_i^* = \min_{k\in[1,N]}\{CV-RMSE_{i,1},\cdots,CV-RMSE_{i,k},\cdots,CV-RMSE_{i,N}\} \tag{6-a}$$

$$r_{CV-RMSE^*,CV_{overall}} = \frac{cov(CV-RMSE^*, CV_{overall})}{\sigma_{CV-RMSE^*}\sigma_{CV_{overall}}} \tag{6-b}$$

where $r_{CV-RMSE^*,CV_{overall}}$ denotes the Pearson correlation coefficient between the dataset of the best achievable forecasting performance $CV-RMSE^*$ and the dataset of intraday load dispersion level $CV_{overall}$; $CV-RMSE_i^*$ denotes the best achievable forecasting performance of Building i; $CV-RMSE_{i,k}$ denotes the coefficient of variation of the root mean square error of model k for Building i; N denotes the total number of prediction models for each building; cov and σ refer to the covariance and standard deviation, respectively. Note that variables can be considered highly correlated when the magnitude of their correlation coefficient is larger than 0.7 [60].

3. Target buildings and results

3.1. Dataset description and experimental settings

The target buildings used in this study are 56 non-domestic buildings with various primary usage types across the UK, as shown in Fig. 7. The dataset to be analysed for each building includes:

- *Electricity meter readings*: the electricity meter readings are at the time interval of 30 min and span from 05 January 2009 to 26 September 2010 (630 days, 90 weeks, and around 1.73 years).
- *Building metadata*: the building metadata provides the general building information, including geographical location (longitude and latitude) and primary usage type. The 56 buildings are categorized into 7 primary usage types, and they are 3 farming buildings, 5 hotels, 5 manufacturing buildings, 4 offices, 30 pubs, 4 restaurants, and 5 supermarkets.
- *Outdoor weather conditions*: outdoor weather conditions are the key influential factors for building electricity usage. In this study, the corresponding weather data were collected from the Integrated Surface Database (ISD), National Centres for Environmental Information [61]. The weather data from 40 meteorological stations across the UK, as shown in Fig. 7, were first retrieved from the ISD. By calculating the Euclidean distance, the nearest meteorological station was then selected for each building. The coincident meteorological variables used in this study include wind speed (*m/s*), air dry-bulb temperature (℃), and air dew point (℃).

All prediction models are developed in the Python programming language. Specifically, SVR and RF are developed using the Scikit-learn ML library [62]. XGB and LightGBM are developed using XGBoost [63] and LightGBM [30] software libraries, respectively. Neural network models, including ANN and three deep learning models, are developed based on Keras [64], which is an open-source library for artificial neural networks. Regarding computation tools, the two naive models and five conventional ML models are developed and performed on a desktop computer (Dell Optiplex 7070) with Intel Core i7-9700 (3.00 GHz) and 16 GB of memory under the Windows 10 64-bit operating system. The three deep learning models are implemented on the Google Colab Pro platform [65] to use the high-performance GPUs (NVIDIA Tesla P100 PCIe 16G) for GPU-accelerated parallel computing.

3.2. Load profile visualization and dispersion quantification

Before short-term load forecasting using different prediction models, the load profiles of 56 non-domestic buildings are first visualized based on the newly proposed visualization method. Fig. 8 shows the load profiles for selected 14 buildings and for each type of building, two buildings with different levels of dispersion are selected for comparison purposes. As discussed in Section 3.2, three dimensions of knowledge can be directly observed from the plots, including intraday load shape, intraday load discrepancy, and intraday load dispersion. First, it can be found that buildings with different primary usage types have evident differences in intraday load shapes. For example, farming buildings (Fig. 8-a and Fig. 8-b) have two short-time demand peaks in the morning and evening, while office buildings (Fig. 8-g and Fig. 8-h) have long-lasting demand peaks during the common working day. For buildings with the same type of primary usage type, their intraday load shapes might differ significantly. For example, Restaurant #01 (Fig. 8-k) has one demand peak at noon and one demand peak at late night, while Restaurant #00 (Fig. 8-l) only has one demand peak before noon. This might be caused by the different opening hours of different restaurants. Second, we can find that different buildings have different levels of intraweek load discrepancy by comparing the average load profiles on weekdays and at weekends. For example, Pub #15 (Fig. 8-j) has a larger load pattern discrepancy between weekdays and weekends than Pub #11 (Fig. 8-i), which indicates the owner of Pub #15 has a larger difference in energy use behaviours through the week. Last, it can be found buildings have different levels of intraday load dispersion (widely

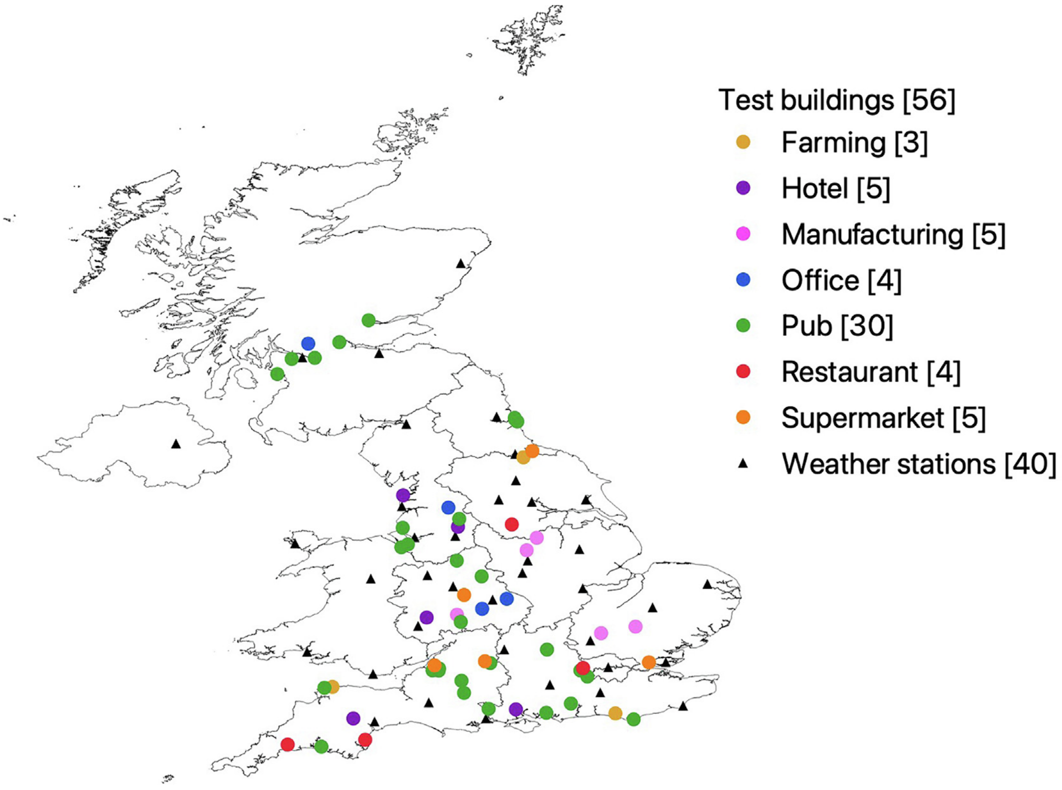


Fig. 7. Target non-domestic buildings with various primary usage types across the UK. The numbers of buildings in each group and weather stations are indicated in square brackets.

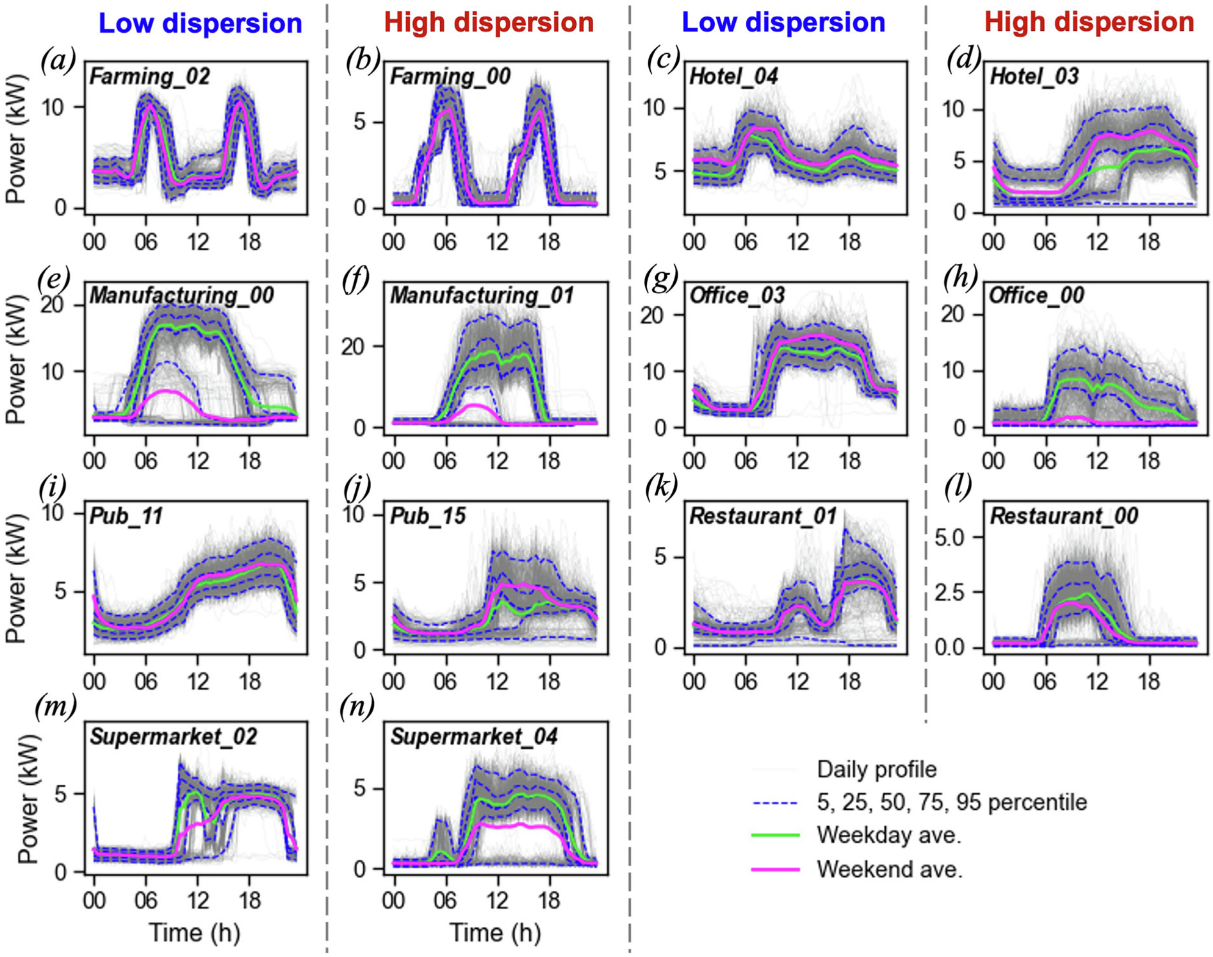


Fig. 8. Visualization of load profiles for selected 14 buildings (90 weeks). For each type of building, one low-dispersion building and one high-dispersion building are selected for comparison purposes.

or densely distributed) by observing the percentile set, even for the same type of buildings. For each type of building, one low-dispersion building and one high-dispersion building are given in Fig. 8 for comparison. Taking hotel buildings for example, Hotel #4 has a lower overall dispersion than Hotel #3 since its percentiles are less widespread than Hotel #3.

Apart from developing a method to better visualize the intraday load dispersion, two metrics are developed in this study to quantify the load dispersion level as defined in Eqs. (3) - (4). After applying the metrics to 56 non-domestic buildings in the UK, their intraday load dispersion levels can be obtained as shown in Fig. 9. Two buildings of the same type but with different dispersion levels are marked as well in Fig. 9-a for comparison purposes. It can be found that the dispersion levels greatly differ among all buildings and within the same type of buildings. As shown in Fig. 9-b, the two metrics for intraday load dispersion, i.e., overall CV and CQV, have a similar trend, indicating both metrics can be used to effectively describe the load dispersion level and capture the dispersion difference between buildings in a quantitative way.

3.3. Model performance evaluation and comparison

The entire dataset for each building (90 weeks) is divided into training, validation, and test data with proportions of 72 % (65 weeks), 18 % (16 weeks), and 10 % (9 weeks), as shown in Fig. 10. The detailed process of model development is introduced in Appendix A. After training out the optimal hyperparameters in the conventional ML models and deep learning models based on the training (65 weeks) and validation (16 weeks) datasets, 24-hour ahead load profiles during the test session (9 weeks) are forecasted using the developed 11 models and evaluated using three metrics, i.e., $MAE_{overall}$, $RMSE_{overall}$, and $CV-RMSE_{overall}$. The total number of observation points, N in Eq. (6), equals 3,024 (63 test days × 48 observation points in each day).

Figs. 11–13 show the forecasting performances of 11 prediction models for 56 buildings based on different performance metrics. Three types of statistical features, including mean, median, and the count of being the best (i.e., times of having the minimum value of one specific metric), are used to evaluate the overall model performance over 56 buildings, as shown in Table 1. The forecasting performance distributions are also shown in Fig. 14 using a violin plot, which is a combination of KDE (kernel density estimation) and box plot.

Two types of comparisons are made in this section: 1) ***cross-model comparison***, by which we attempt to answer what model is the best in general; and 2) ***cross-building comparison***, by which we attempt to investigate the performance differences across different buildings. It is worth mentioning two points: 1) all three metrics, i.e., $MAE_{overall}$, $RMSE_{overall}$, and $CV-RMSE_{overall}$, can be used for cross-model comparison, while only $CV-RMSE_{overall}$ can be used for cross-building comparison since the other two are scale-dependent metrics and not suitable for building-to-building comparison; and 2) based on different combinations of performance metric and statistical feature (performance metric × statistical fea-

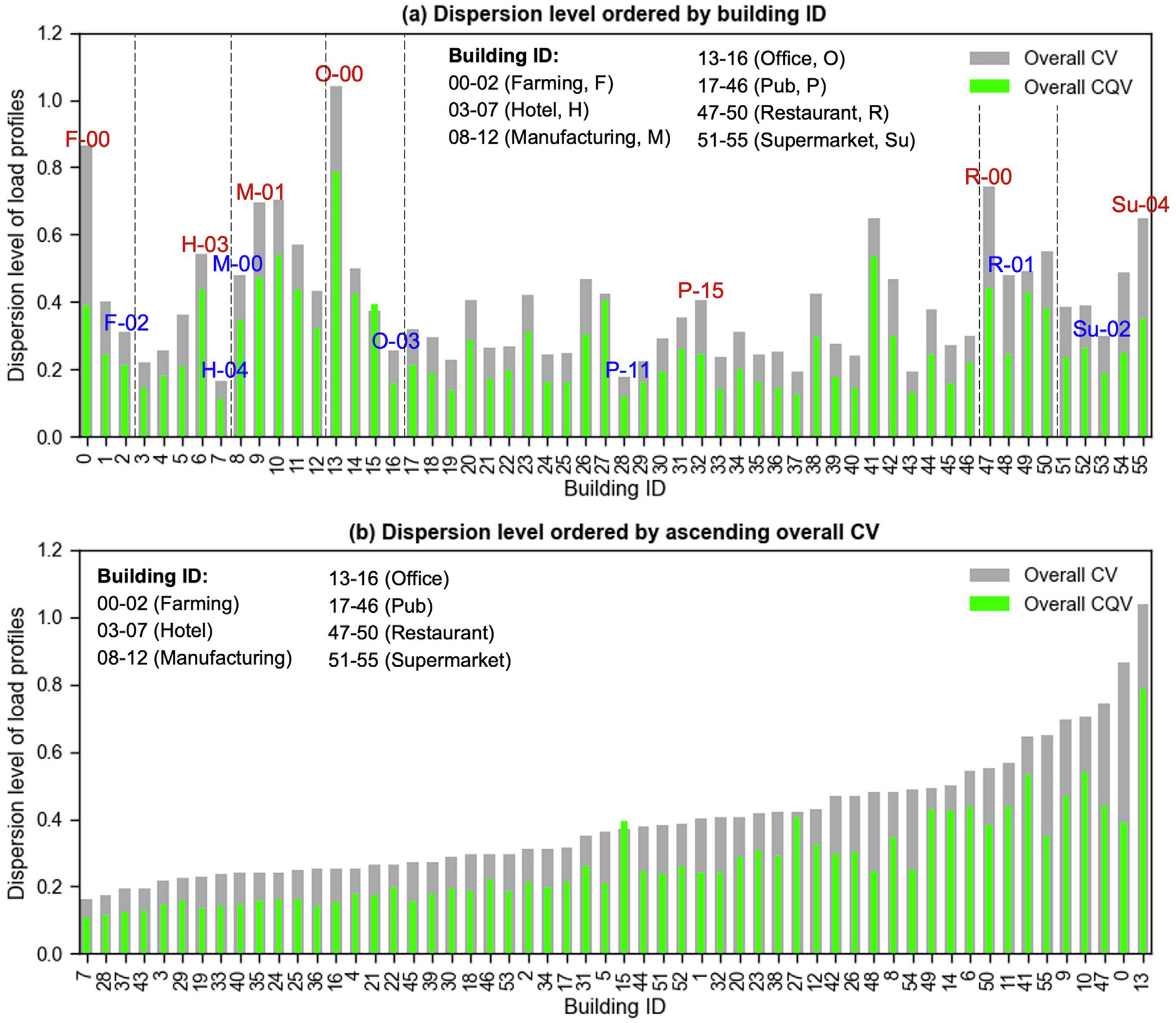


Fig. 9. Dispersion levels of intraday load profiles (90 weeks) for 56 non-domestic British buildings: a) dispersion level ordered by building ID. For each type of building, one low-dispersion building and one high-dispersion building are marked in different colours for comparison. b) dispersion level ordered by ascending overall CV.

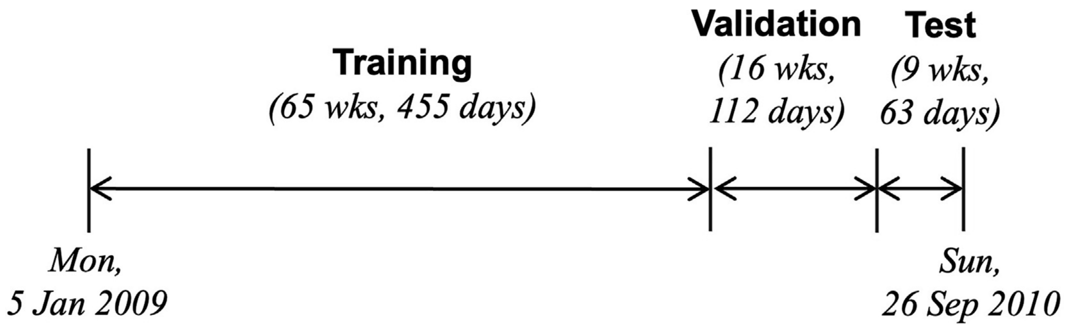


Fig. 10. Data split for each building.

ture), the rankings of the best forecasting models might slightly differ as shown below.

3.3.1. Cross-model comparison

For the cross-model comparison, both the performance of the same category of models and the performance across different categories of models are compared in this subsection.

- *Comparison among the same category of models.*

For the category of naive models, it is found that Naive model 2 (same-day-previous-week) outperforms Naive model 1 (Previous-day), regardless of what type of performance metric is used. It is because the energy use behaviour of occupants and energy load profiles have not only the daily seasonality but also the weekly seasonality. Among the five conventional ML models (ML 1–5), the

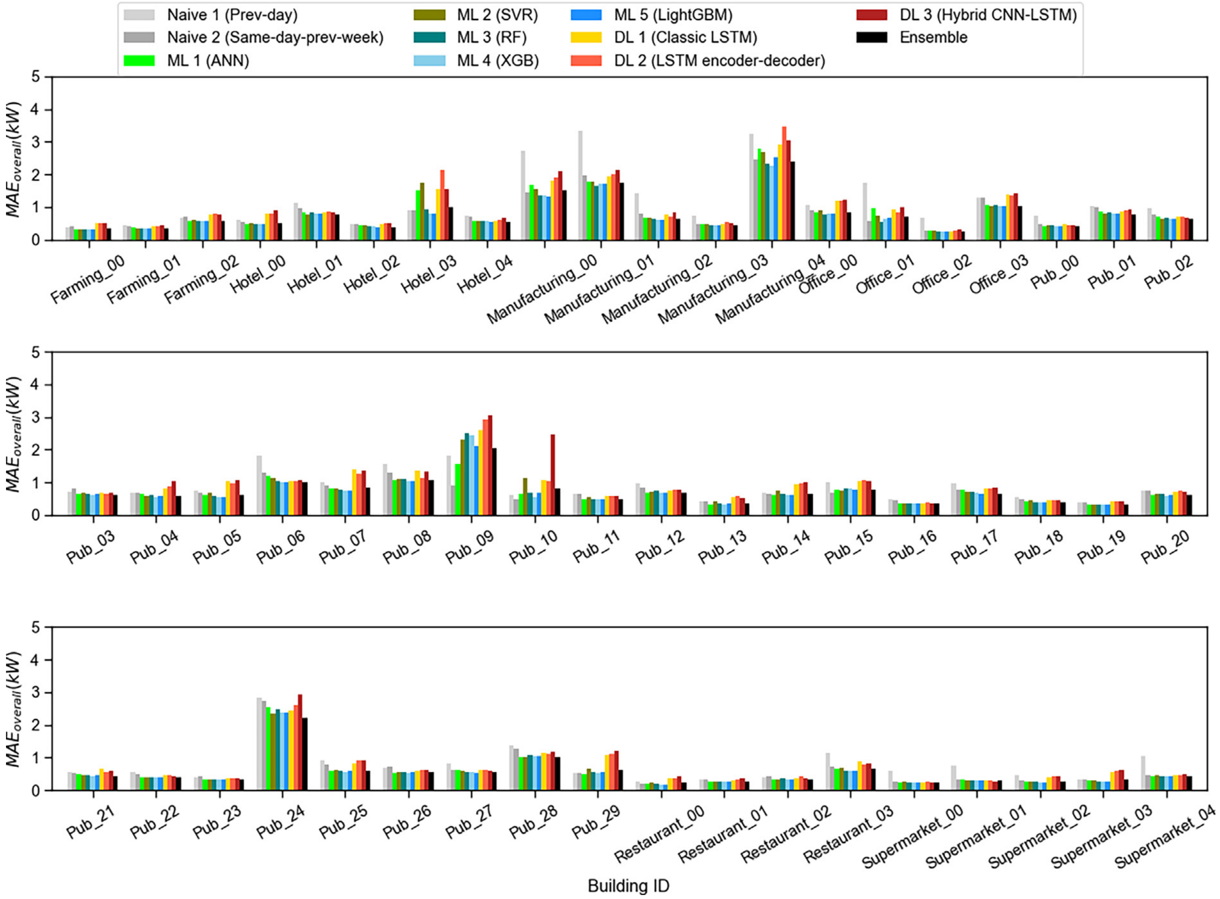


Fig. 11. Forecasting performance (MAE) during the test session (56 buildings × 11 models × 63 test days).

best forecasting performance is achieved by either XGB or LightGBM, depending on the chosen performance metric. Either RF or ANN is in third place, and SVR is the model with the least accuracy in all cases. To some extent, the finding is consistent with the results of the ASHRAE Great Energy Predictor III competition [32], in which gradient boosted decision trees, including XGB and LightGBM, dominate this energy prediction competition. In the category of deep learning model (DL 1–3), the best forecasting performance is achieved by the classic LSTM model in most cases except when in order of the median of $MAE_{overall}$ and the count of having the minimum $MAE_{overall}$. The least accurate DL model is the hybrid CNN-LSTM model in most cases except when the count of having the minimum $MAE_{overall}$ is used as performance metric.

- *Comparison across different categories of models.*

As listed in Table 1, under 9 different ranking criteria, the best forecasting performance is achieved by ensemble model 6 times, and by ML 4 (XGB) 3 times. The top 3 most accurate models (in no particular order) are ensemble model, ML 4 (XGB), and ML 5 (LightGBM) in most cases, except when in order of the median value of $MAE_{overall}$. Naive model 1 (Previous-day) is the least accurate in most cases except when the median value of $MAE_{overall}$ is chosen as the ranking criterion, in which the least accurate model is DL 3 (Hybrid CNN-LSTM).

In summary, by the cross-model comparison of the test results (56 buildings × 11 models × 63 test days), the following findings can be observed: 1) Naive model 2 (same-day-previous-week), ML 4 (XGB), and DL 1 (Classic LSTM model) are the best of its kind, respectively; 2) among all categories of models, the best forecasting performance is achieved by either ensemble model or ML 4 (XGB), depending on the selected performance metric and statistical feature; 3) Naive model 2 (same-day-previous-week) is not naive. Overall, it has close forecast accuracy with ML models but has better performances than deep learning models after a 9-week-long test.

3.3.2. Cross-building comparison

In addition to the cross-model comparison, a cross-building comparison is conducted to analyse the performance differences across different buildings based on the scale-independent forecasting performance metric $CV - RMSE_{overall}$. To investigate the link between the load profile dispersion and load forecasting performance, the load profiles of selected buildings with a relatively low or high dispersion level are also visualized in Fig. 13 for comparison convenience.

It can be observed from Fig. 13 that the best forecasting performances (i.e., minimum $CV - RMSE_{overall}$ achieved by 11 prediction models) of 56 buildings significantly differ and each building has its best forecasting performance. For example, the highest achievable forecast accuracy of Farming #00, Hotel #3, Manufacturing # 1, Office #00, Pub #15, Restaurant #0, and Supermarket #4 is lower than the other buildings with the same primary usage type. More importantly, a significant link can be found between the best forecasting performances of buildings and their load dispersion levels. Buildings with higher dispersion levels are more likely to have lower forecasting performance. For example, the high-dispersion buildings in Fig. 13, including Hotel #3, Office #00, Pub #15, and Restaurant #00, have lower forecasting performance compared with the low-dispersion buildings with the same primary usage

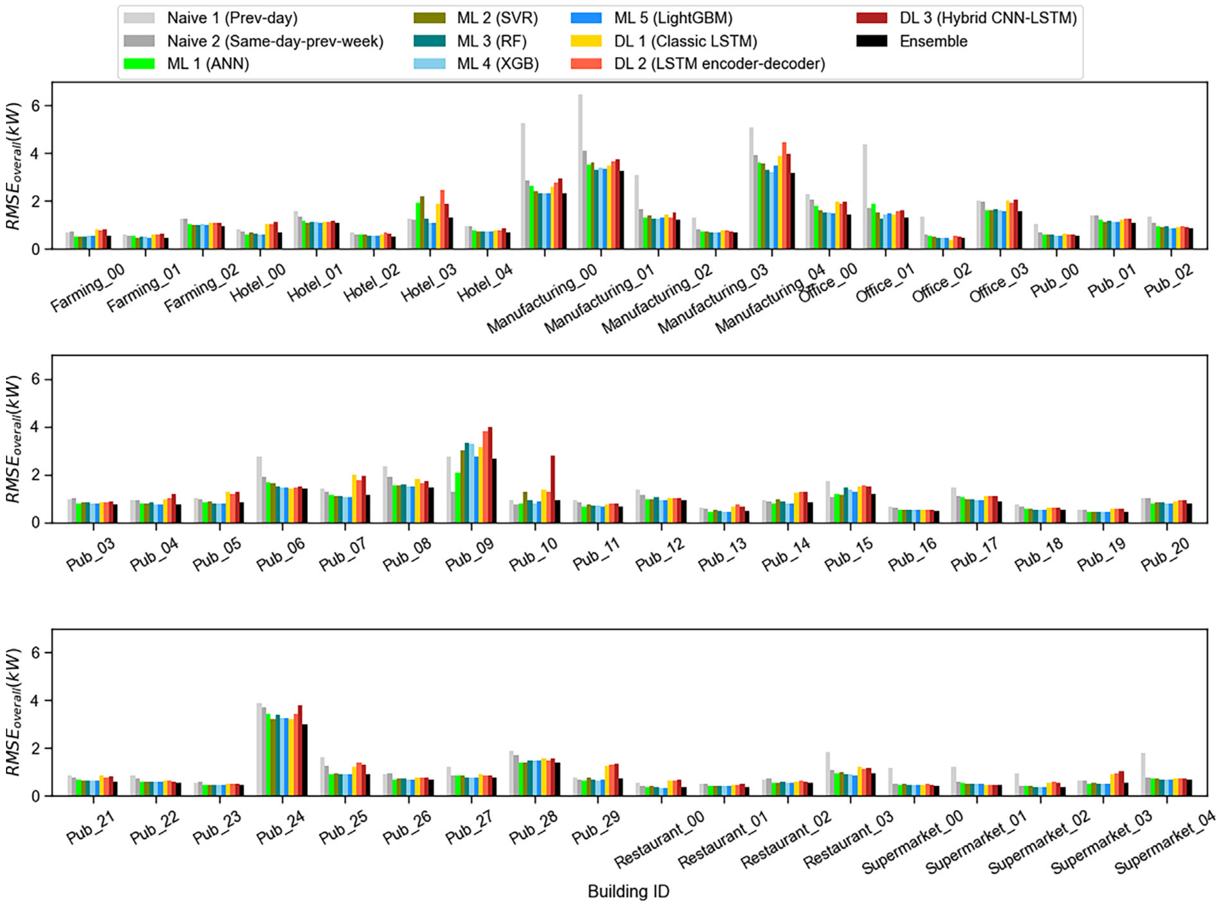


Fig. 12. Forecasting performance (RMSE) during the test session (56 buildings × 11 models × 63 test days).

type, i.e., Hotel #04, Office #03, Pub #11, and Restaurant #01. To quantitatively investigate the relation between load dispersion level and best achievable forecasting performance, correlation analysis is conducted in the following subsection.

3.4. Correlation analysis

Correlation analysis is conducted in this subsection to investigate the effects of load dispersion level on the best achievable forecasting performance. To have an intuitive sense of the link between these two variables, the forecast accuracy ($CV - RMSE_{min}$) and dispersion level ($CV_{overall}$ and $CQV_{overall}$) are first plotted against the building IDs in the order of ascending $CV - RMSE_{min}$, as shown in Fig. 15. Dispersion levels on all datasets (90 weeks, Fig. 15-a) and on the test dataset (9 weeks, Fig. 15-b) are both calculated and plotted against building IDs. The correlations between the best achievable forecasting performances and different types of dispersion levels are shown in Fig. 16 using the scatterplot matrix.

The two dispersion level metrics ($CV_{overall}$ and $CQV_{overall}$) on all datasets (Fig. 15-a) and test dataset (Fig. 15-b) are found to have identical trends and the correlation coefficients between them on all datasets and test dataset are 0.92 and 0.9, respectively, as shown in Fig. 16, which means the two metrics are very highly correlated and both can play the same role in characterizing the dispersion levels of load profiles. Moreover, the dispersion level on all datasets and the dispersion level on the test dataset differ but are related, which can be indicated by the correlation coefficient between $CV_{overall,alldatasets}$ and $CV_{overall,testdataset}$ (i.e., 0.83) and the correlation coefficient between $CQV_{overall,alldatasets}$ and $CQV_{overall,testdataset}$ (i.e., 0.78). Most importantly, it can be found that the best forecasting performance ($CV - RMSE_{min}$) has the same trend with both dispersion level metrics ($CV_{overall}$ and $CQV_{overall}$). The dispersion level metrics on different datasets, including $CV_{overall,alldatasets}$, $CQV_{overall,alldatasets}$, $CV_{overall,testdataset}$, and $CQV_{overall,testdataset}$, are all correlated with the best forecasting performance, $CV - RMSE_{min}$, with the correlation coefficients of 0.78, 0.76, 0.86, and 0.78, respectively. Compared with the dispersion level on all datasets, the dispersion level on the test dataset has a slightly higher correlation with the best forecasting performance.

4. Discussion

Reliability and interpretability play a key role in the development and practical application of artificial intelligence and ML techniques in power and energy systems. For building load forecasting, a forecasting technique will ideally have reliable and robust performance under different weather conditions and occupant behaviours across different buildings. It is also expected that the success of one forecasting technique, especially a black-box ML model, can be interpreted in terms of domain knowledge to assist with operational decision support. However, without applying different forecasting techniques to numerous buildings at a large scale over a long period, it's difficult to evaluate the reliability of prediction models and to interpret the prediction results with domain knowledge alone. In this section, we explain how the test results answer the two major research questions with an emphasis on the reliability and interpretability of artificial intelligence in the field of building energy forecasting.

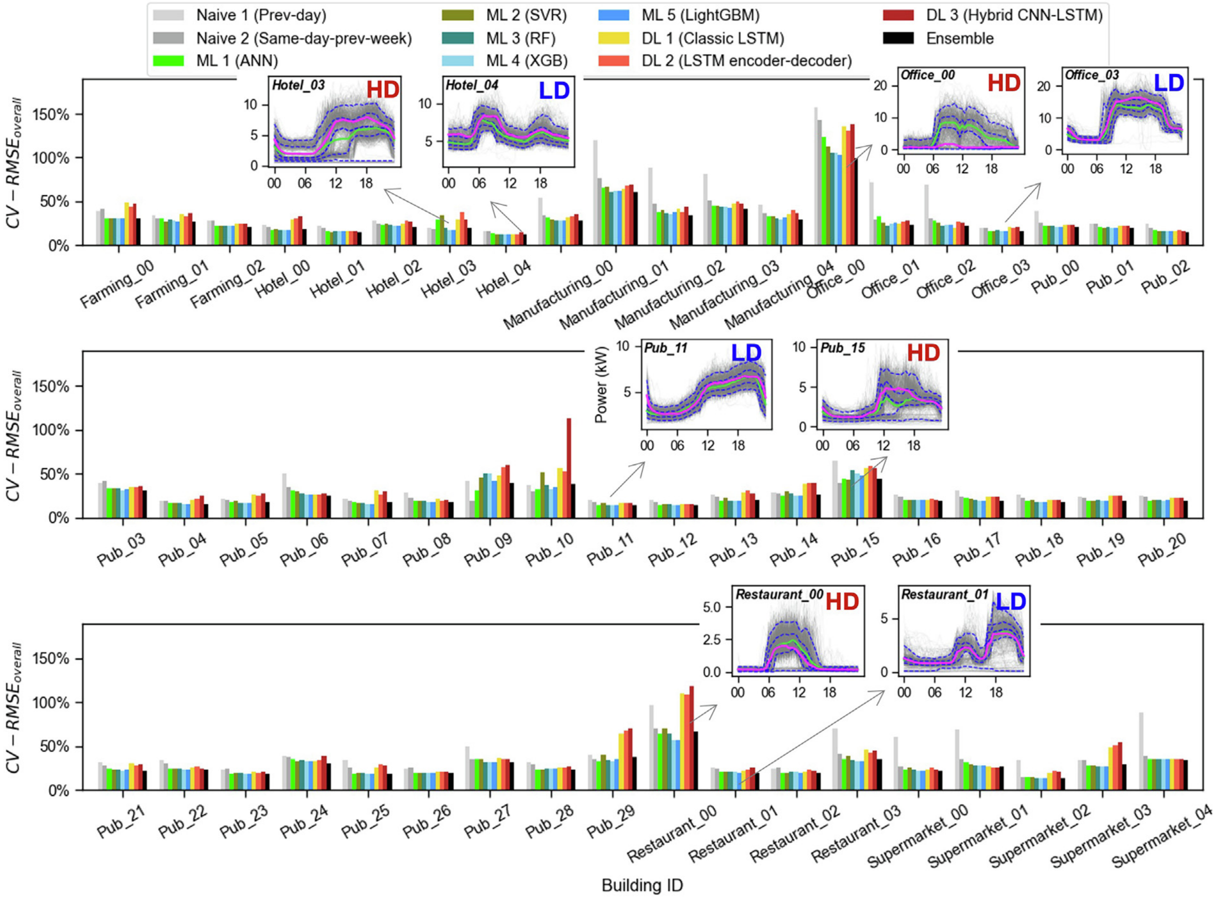


Fig. 13. Forecasting performance (CV-RMSE) during the test session (56 buildings × 11 models × 63 test days) and load profiles of selected buildings with a relatively low or high dispersion (LD or HD) level.

Table 1
Statistical features of different performance metrics of 11 prediction models for 56 buildings. (***, **, and * denote the top 3 most accurate models when using the corresponding statistical feature).

Performance metrics	Statistical features (N = 56)	Naive 1 (Prev-day)	Naive 2 (Same-day-prev-week)	ML 1 (ANN)	ML 2 (SVR)	ML 3 (RF)	ML 4 (XGB)	ML 5 (LightGBM)	DL 1 (Classic LSTM)	DL 2 (LSTM encoder-decoder)	DL 3 (Hybrid CNN-LSTM)	Ensemble
$MAE_{overall}$(kW)	Mean of $MAE_{overall}$	0.967	0.754	0.720	0.741	0.694	**0.670*****	**0.673****	0.865	0.894	0.938	**0.686***
	Median of $MAE_{overall}$	0.727	0.664	0.603	0.615	**0.572***	**0.562*****	**0.567****	0.736	0.731	0.742	0.595
	Count of having the min.$MAE_{overall}$	0	3	3	1	4	**20*****	**9***	0	0	1	**15****
$RMSE_{overall}$(kW)	Mean of $RMSE_{overall}$	1.563	1.169	1.053	1.075	1.033	**1.007***	**1****	1.199	1.238	1.287	**0.982*****
	Median of $RMSE_{overall}$	1.184	0.926	0.796	0.826	0.811	**0.771****	**0.772***	0.948	0.977	1.023	**0.767*****
	Count of having the min.$RMSE_{overall}$	0	3	5	0	3	**7****	**7****	1	1	0	**29*****
$CV - RMSE_{overall}$	Mean of $CV - RMSE_{overall}$	42.5 %	31.6 %	28.1 %	28.8 %	27.5 %	**26.8 %***	**26.6 %****	33.4 %	34.0 %	35.8 %	**26.5 %*****
	Median of $CV - RMSE_{overall}$	32.4 %	26.3 %	23.3 %	23.8 %	22.6 %	**22.5 %***	**22.3 %****	26.0 %	26.7 %	27.0 %	**21.7 %*****
	Count of having the min.$CV - RMSE_{overall}$	0	3	5	0	3	**7****	**7****	1	1	0	**29*****

4.1. Load dispersion quantification based on better visualization

Better visualization of building electricity consumption data can help to provide intuitive insights into the energy use behaviours of customers. By using the newly developed visualization method, the following insights into load profiles can be quickly captured, including intraday load shape (i.e., the trends and load peaks during a day), intraweek load discrepancy (i.e., load pattern discrepancy between weekdays and weekends), and intraday load dispersion (the dispersion level of the load profiles, i.e., widely or densely distributed). Two dispersion level metrics ($CV_{overall}$ and $CQV_{overall}$) are developed to quantify the intraday load dispersion levels, and test results show that the two metrics are very highly correlated, and both can play the same role in characterizing the dispersion levels of load profiles.

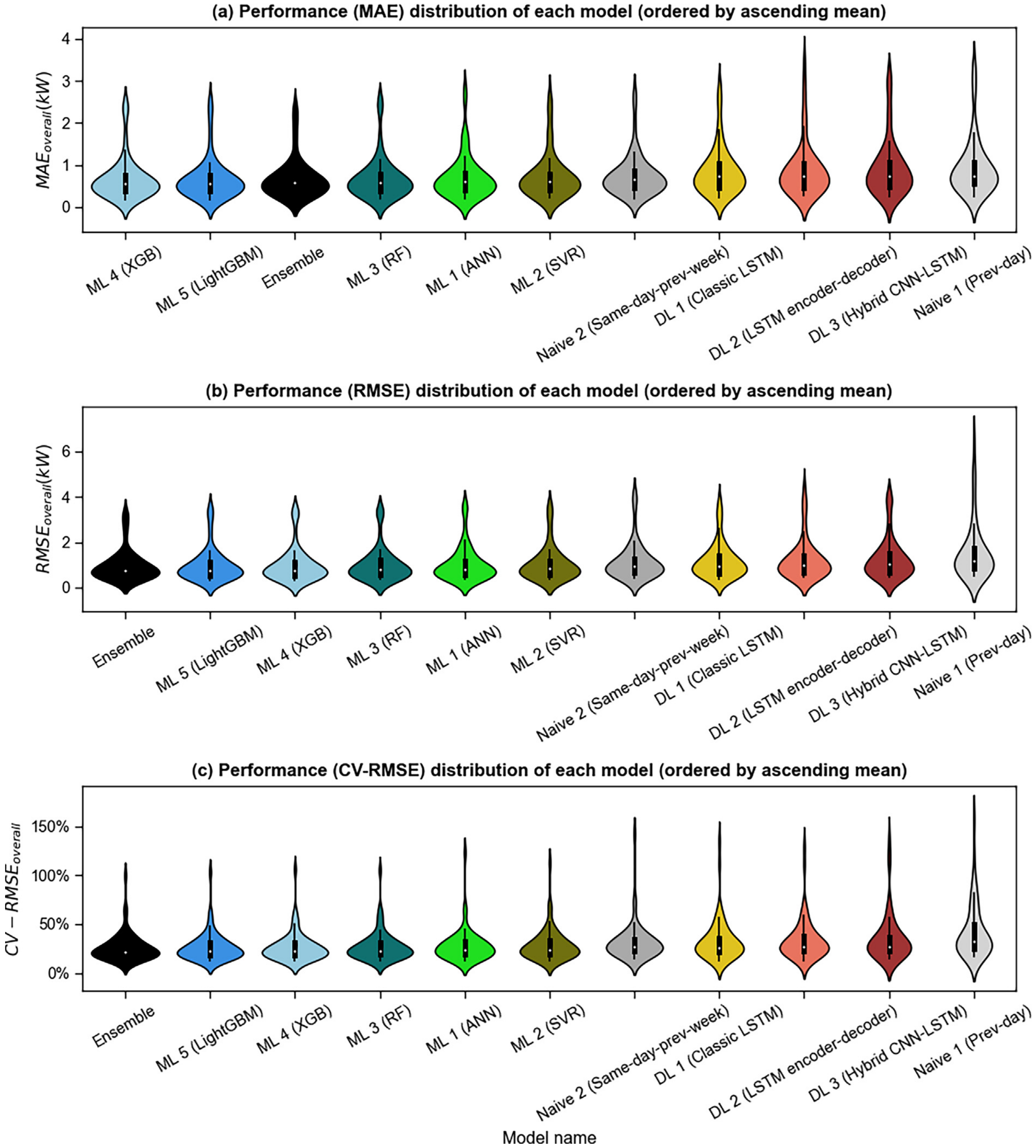


Fig. 14. Violin plots of forecasting performance distributions of 11 prediction models for 56 buildings: a) MAE; b) RMSE; c) CV-RMSE. The violin plot is a combination of KDE (kernel density estimation) and box plot. For each violin plot, the violin width reflects the relative frequency; the inner white point and black box indicate the median and interquartile range, respectively. The distribution plots here are ordered by ascending mean.

4.2. Effects of algorithms on forecast accuracy

Prediction techniques used in this study include two naive models, five conventional ML models, and three deep learning models. Moreover, the ensemble learning technique is used to integrate the performances of the individual base models. 9 different ranking criteria are used to evaluate and rank the forecasting performances of 11 models based on a 9-week-long test (56 buildings × 11 models × 63 test days). The ranking criteria are the combinations of performance metrics ($MAE_{overall}$, $RMSE_{overall}$, or $CV-RMSE_{overall}$) and statistical features (mean, median, or the count of being the best). Test results show that overall, ensemble learning can help to improve the forecast accuracy and provide better forecasting performance than using individual base models alone. Even though the performances of some individual models are poor, the ensemble model can achieve satisfactory performance. This is because ensemble learning can take the most advantage of the individual base models and make robust predictions.

In addition, although deep learning techniques have been increasingly used for building load forecasting in recent years [15,34–39], we find that conventional ML models still provide more accurate forecasting performances than complex deep learning models. Likewise, a recent study by Elsayed et al. [66] demonstrated that conventional ML approaches such as gradient boosting

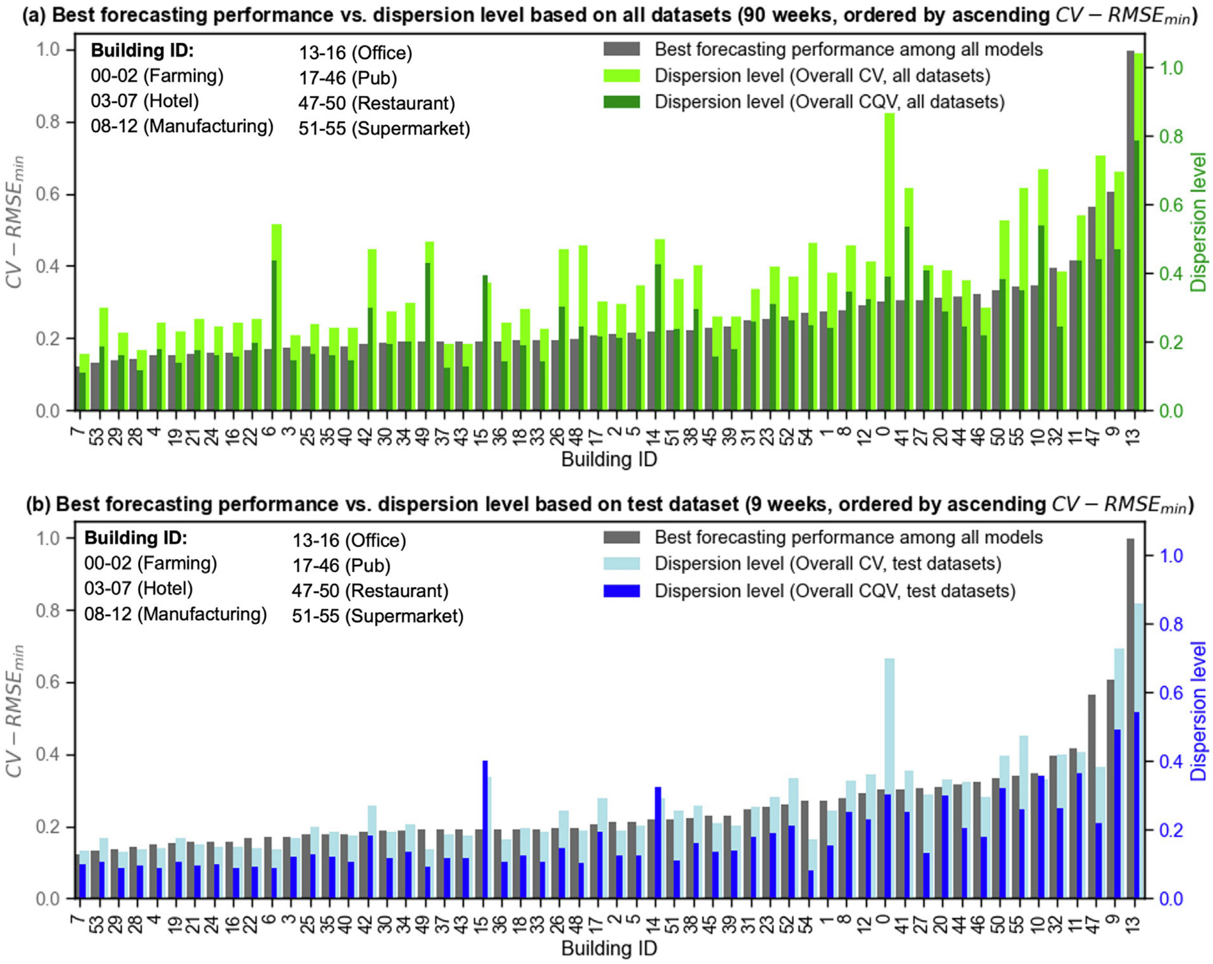


Fig. 15. Best forecasting performance vs load dispersion level based on a) all datasets or b) test dataset.

tree models outperformed deep learning approaches in the time series forecasting context.

Another unforeseen finding from this study is that the naive persistent model 2 (i.e., using the load profile of the same day in the previous week as the forecasted profile) is not naive at all but has close forecast accuracy with some ML models and outperforms all deep learning models after a 9-week-long test. To some extent, this finding is consistent with the finding in a day-ahead electricity demand forecasting competition [67], in which the benchmarking persistent model outperformed most teams after 30 rounds of day-ahead prediction. The reason why Naive model 2 (same-day-previous-week) outperforms some complicated AI-based models in our study is that building load profiles are by nature time-series data and contain hidden temporal dependences (e.g., daily and weekly seasonality) between observations. The essence of the prediction problem in this study is a multivariate multi-step time series forecasting problem. In addition, the energy-related behaviours are variable, and statistical learning based on a long period of historical data sometimes is less robust to the changes in such behaviours than the naive approach.

4.3. Effects of load dispersion level on forecast accuracy

After the cross-building comparison, we found that each building has its best achievable forecasting performance; and more importantly, a strong link can be observed between the best forecasting performances of buildings and their load dispersion levels. As shown in Fig. 13, buildings with higher dispersion levels are found more likely to have lower forecasting performance. To quantitively assess the relation between the two variables, a correlation analysis is further conducted to investigate the effects of load dispersion level on the best load forecasting performance. Results show the dispersion level metrics are highly correlated with the best achievable forecasting performance. This means the dispersion level of load profiles has a large influence on forecasting performance. Compared with the algorithms (i.e., prediction models), the load dispersion level, might contribute more to the forecast accuracy when testing over a long period of time.

5. Conclusions

In this paper, we examined the impacts of building load dispersion level on its best load forecasting accuracy. This was accomplished by comparing the performances of 11 different prediction models over 9 weeks of operation for 56 non-domestic buildings with different primary usage types in the UK. The major conclusions and suggestions for the researchers and engineers in the area of building load forecasting are given as follows:

1) **Load dispersion quantification based on better visualization**. Better visualization gives better insights. Before developing prediction models for a specific building, it is recommended to plot the building load profiles using our proposed visualization method, which can help to quickly capture some intuitive and useful insights, including intra-

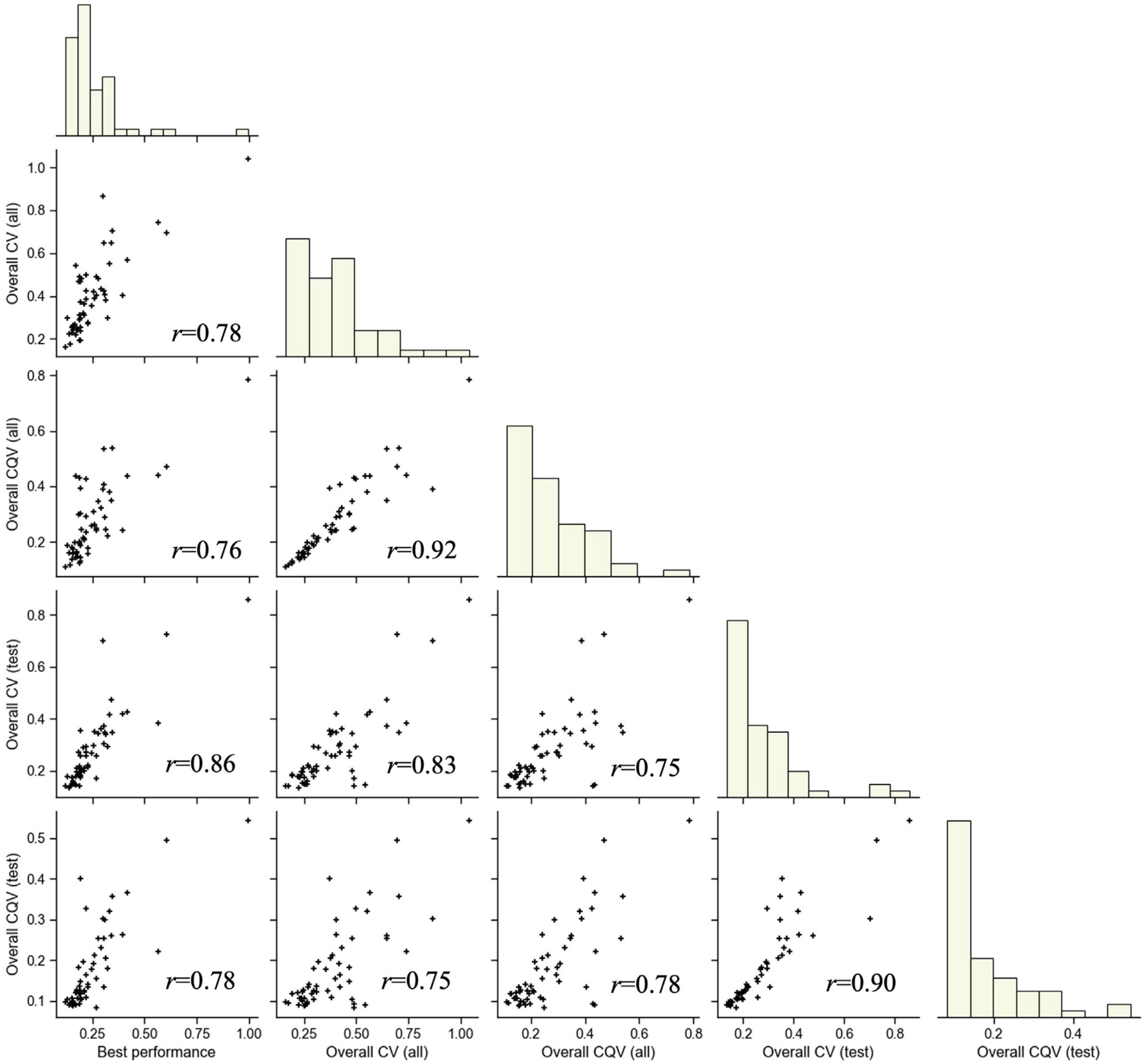


Fig. 16. Scatterplot matrix for the correlations between best achievable forecasting performance and different types of dispersion levels. r denotes the Pearson correlation coefficient.

day load shape, intraweek load discrepancy, and intraday load dispersion. Based on the newly proposed visualization method, we propose novel metrics to quantify the dispersion level of building intraday load profiles.

2) **Prediction techniques**. There is not a panacea-like individual model for all building load forecasting problems. Comparisons among different categories of models based on different mechanisms are needed to determine the best model for a specific building. Specifically, the following findings and suggestions are given for prediction model development: *i)* A complex model structure does not guarantee a higher forecasting performance, and the models with simple structures could outperform other complicated ones. We find that conventional ML models still outperform complex deep learning models, despite the increasing application of deep learning techniques for building load forecasting in recent years. We also demonstrate that the naive persistent model (i.e., using the load profile of the same day in the previous week as the forecasted profile) is not naive at all but has close forecast accuracy with some ML models after a long period of testing. We conclude that it is feasible to use the naive persistent model as the benchmarking model in the building load forecasting problem due to its simplicity and computational efficiency. *ii)* Individual prediction models have their strengths and weaknesses, and their prediction results might be poor due to inadequate training data. To solve this problem, ensemble learning is recommended to take the most advantage of individual base models and make robust predictions. Moreover, it is key to diversify the categories of the base models to guarantee the performance of the ensemble model.

3) **Data or algorithm?** We find that each building has its best achievable forecasting performance and most importantly, it is largely influenced by its load dispersion level. Buildings with higher dispersion levels are more likely to have lower forecast accuracy. For a specific building, it's recommended to quantify the dispersion levels using the proposed dispersion metrics of building load profiles. When its load profiles have a low dispersion level, the naive model (same-day-previous-week) is more likely to have a satisfactory forecasting performance and could be directly used for engineering purposes, which helps to save a large amount of time for model development.

Our study is a fundamental starting point to understand the relationship between the building load forecast accuracy and the influential factors coming from the load characteristics. Complementary to the proposed load dispersion level, we will explore other metrics to characterize the building load profiles in the future. Also, since the building energy use data in our study were collected around 2010, we need to test our methodology again when the up-to-date energy consumption data are collected considering the energy consumption patterns might be affected by the occupants' behavioral changes. Moreover, the energy consumption data of more consumers will be collected, so that we can categorize the load profiles using clustering techniques such as k-means. Last, we will improve the ensemble learning performance by optimizing the weights of individual base models.

Data availability

Data will be made available on request.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors would like to acknowledge Opus Energy (www.opusenergy.com) for providing the data used in this paper. The research work presented in this paper was financially supported by the UK Engineering and Physical Sciences Research Council (EPSRC) under the grant (EP/S030131/1).

Appendix A. Development of prediction models

For conventional ML models and deep learning models, the hyperparameters of each model need to be optimized with the assistance of cross-validation and grid-search techniques. The hyperparameters to be optimized and their search ranges in each prediction model are listed in Table A1. In addition, the early stopping training technique is used to avoid the overfitting of training data, in which the training process is terminated when the resulting accuracy on validation data stops increasing after a certain number of epochs.

For ANNs, considering the volume of the dataset for each building, a common four-layer back propagation neural network model is adopted in this study, which consists of an input layer, two hidden layers, and an output layer. Adam algorithm is selected as the optimizer which aims to find the optimal learning rate for each parameter [68]. Other parameters to be optimized in ANNs consists of the units in each layer (ranging from 12 to 48 with an increment of 12) and activation functions (including *Tanh*, *Sigmoid*, and *ReLU*). For SVM, a Gaussian radial basis function (*RBF*) kernel is used in this study and other major parameters to be determined are the regularization parameter C (ranging from 0.1 to 10) and the kernel efficient γ (ranging from 0.001 to 0.1). RF, XGB, and LightGBM are all decision tree-based techniques. The major tuned hyperparameters of RF models include the total number of trees (ranging from 50 to 100) and the maximum depth of each tree (ranging from 4 to 7). XGB and LightGBM have the same parameters to be tuned, including the maximum depth of each tree (ranging from 4 to 7) and the learning rate (ranging from 0.01 to 0.2). The learning rate indicates how quickly a tree model adjusts the errors in the previous iteration.

Table A1
Search ranges of the hyperparameters when using the grid-search technique for each prediction model.

Prediction model	Hyperparameters in each model	Search ranges
ANNs	Number of hidden layers	2
	Units in each layer	12, 24, 36, 48
	Activation function	ReLU, Sigmoid, Tanh
	Optimizer	Adam
SVR	Kernel	rbf (radial basis function)
	Regularization parameter C	0.1, 1, 10
	Kernel coefficient, γ	0.001, 0.01, 0.1
RF	Number of trees	50, 100
	Maximum depth of each tree	4, 5, 6, 7
	Bootstrap	True, False
XGB/LightGBM	Maximum depth of each tree	4, 5, 6, 7
	Learning rate	0.01, 0.025, 0.05, 0.075, 0.1, 0.2
Classic LSTM model/LSTM encoder-decoder model	Units in each LSTM layer	12, 24, 36, 48
	Activation function in each LSTM layer	ReLU, Sigmoid, Tanh
	Dropout ratio in each LSTM layer	0, 0.1, 0.2, 0.3
	Units in each Dense layer	12, 24, 36, 48
	Activation function in each Dense layer	ReLU, Sigmoid, Tanh
Hybrid CNN-LSTM model	Filter number in each 1D convolution layer	12, 24, 36, 48
	Activation function in each convolution layer	ReLU, Sigmoid, Tanh
	Kernel size in each convolution layer	4, 6, 8
	Stride size in each convolution layer	1, 2
	Units in each LSTM layer	12, 24, 36, 48
	Activation function in each LSTM layer	ReLU, Sigmoid, Tanh
	Dropout ratio in each LSTM layer	0, 0.1, 0.2, 0.3
	Units in each Dense layer	12, 24, 36, 48
	Activation function in each Dense layer	ReLU, Sigmoid, Tanh

In our study, the three deep learning models, including the classic LSTM model, LSTM encoder-decoder model, and hybrid CNN-LSTM model, are developed based on Keras [64], which is an open-source library for artificial neural networks. Taking farming building #00 for example, the detailed network structures of the three multi-headed deep learning models based on Keras platform are shown in Figs. A1 - A3. As shown in Table A, the classic LSTM model and LSTM encoder-decoder model includes two types of neural network layers: LSTM layer and fully connected layer (i.e., Dense layer). For the LSTM layers, the optimized hyperparameters consist of the units in each layer (ranging from 12 to 48 with an increment of 12), activation functions (including *Tanh*, *Sigmoid*, and *ReLU*), and dropout ratio (ranging from 0 to 0.3). For the Dense layers, the units in each layer and activation functions are the major parameters. In the hybrid CNN-LSTM model, along with the LSTM and Dense layers, convolution layers are included for feature extraction. The parameters in the convolution layers include filter number (ranging from 12 to 48 with an increment of 12), activation function, kernel size (ranging from 4 to 8), and stride size (including 1 and 2).

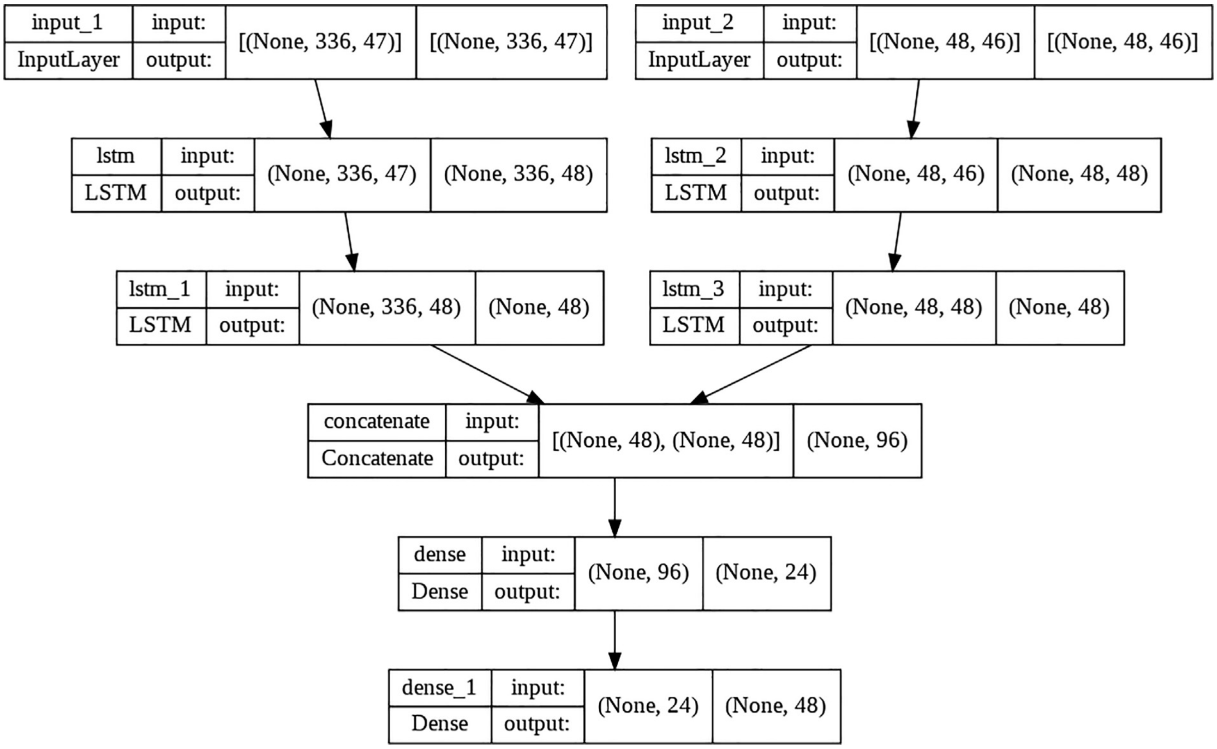


Fig. A1. Network structure of multi-headed classic LSTM model based on the Keras platform.

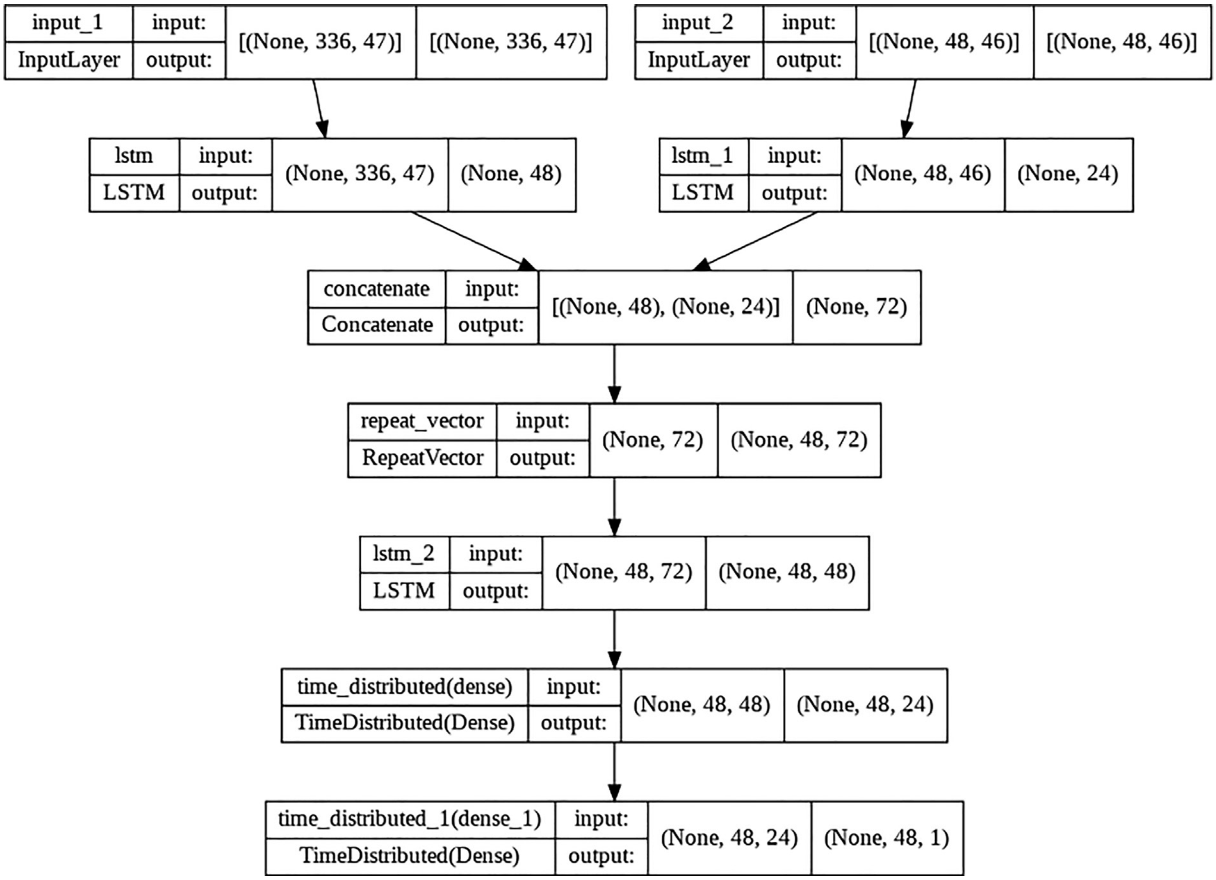


Fig. A2. Network structure of multi-headed LSTM encoder-decoder model based on the Keras platform.

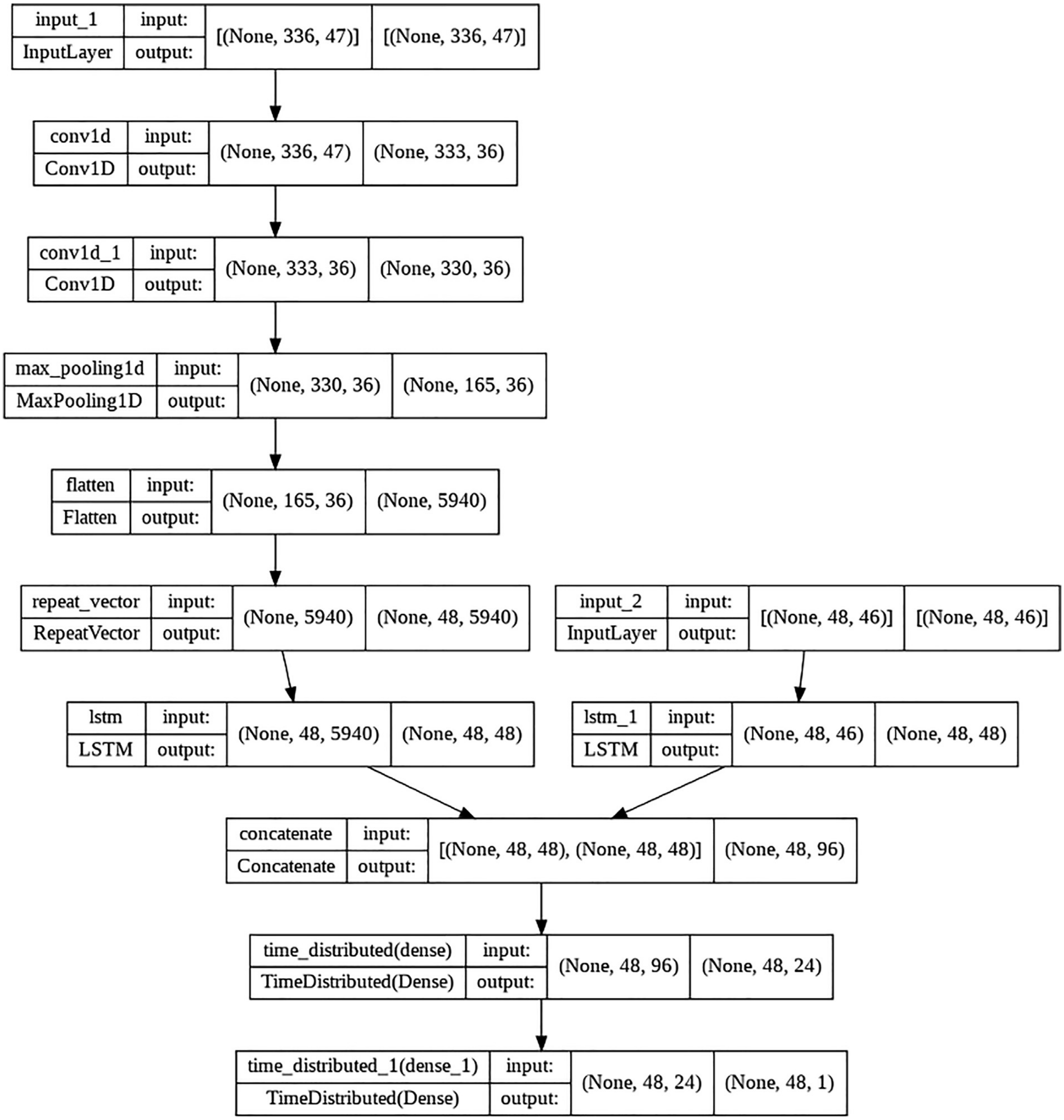

input_1 | InputLayer | input: | output: | [(None, 336, 47)] | [(None, 336, 47)]
conv1d | Conv1D | input: | output: | (None, 336, 47) | (None, 333, 36)
conv1d_1 | Conv1D | input: | output: | (None, 333, 36) | (None, 330, 36)
max_pooling1d | MaxPooling1D | input: | output: | (None, 330, 36) | (None, 165, 36)
flatten | Flatten | input: | output: | (None, 165, 36) | (None, 5940)
repeat_vector | RepeatVector | input: | output: | (None, 5940) | (None, 48, 5940)
input_2 | InputLayer | input: | output: | [(None, 48, 46)] | [(None, 48, 46)]
lstm | LSTM | input: | output: | (None, 48, 5940) | (None, 48, 48)
lstm_1 | LSTM | input: | output: | (None, 48, 46) | (None, 48, 48)
concatenate | Concatenate | input: | output: | [(None, 48, 48), (None, 48, 48)] | (None, 48, 96)
time_distributed(dense) | TimeDistributed(Dense) | input: | output: | (None, 48, 96) | (None, 48, 24)
time_distributed_1(dense_1) | TimeDistributed(Dense) | input: | output: | (None, 48, 24) | (None, 48, 1)

Fig. A3. Network structure of multi-headed hybrid CNN-LSTM model based on the Keras platform.

References

[1] United Nations Environment Programme. 2020 Global Status Report for Buildings and Construction: Towards a Zero-Emission, Efficient and Resilient Buildings and Construction Sector. United Nations Environment Programme Nairobi, Kenya; 2020.

[2] IEA. Energy Technology Perspectives 2020. 2020. https://www.iea.org/reports/energy-technology-perspectives-2020.

[3] Climate Change Committee. The Sixth Carbon Budget: Buildings. 2020. https://www.theccc.org.uk/wp-content/uploads/2020/12/Sector-summary-Buildings.pdf.

[4] S. Asadi, S.S. Amiri, M. Mottahedi, On the development of multi-linear regression analysis to assess energy consumption in the early stages of building design, Energy and Buildings. 85 (2014) 246–255.

[5] M. Hu, F. Xiao, J.B. Jørgensen, R. Li, Price-responsive model predictive control of floor heating systems for demand response using building thermal mass, Applied Thermal Engineering. 153 (2019) 316–329.

[6] X. Li, J. Wen, Review of building energy modeling for control and operation, Renewable and Sustainable Energy Reviews. 37 (2014) 517–537.

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

[8] M. Hu, F. Xiao, J.B. Jørgensen, S. Wang, Frequency control of air conditioners in response to real-time dynamic electricity prices in smart grids, Applied Energy. 242 (2019) 92–106.

[9] H. Fontenot, B. Dong, Modeling and control of building-integrated microgrids for optimal energy management – A review, Applied Energy. 254 (2019).

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

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

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

[13] M. Hu, D. Ge, R. Telford, B. Stephen, D.C.H. Wallom, Classification and characterization of intra-day load curves of PV and non-PV households using interpretable feature extraction and feature-based clustering, Sustainable Cities and Society. 75 (2021).

[14] C. Fan, F. Xiao, Y. Zhao, A short-term building cooling load prediction method using deep learning algorithms, Applied Energy. 195 (2017) 222–233.

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

[16] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature. 521 (2015) 436–444.

[17] A.S. Ahmad, M.Y. Hassan, M.P. Abdullah, H.A. Rahman, F. Hussin, H. Abdullah, et al., A review on applications of ANN and SVM for building electrical energy consumption forecasting, Renewable and Sustainable Energy Reviews. 33 (2014) 102–109.

[18] S. Seyedzadeh, F.P. Rahimian, I. Glesk, M. Roper, Machine learning for estimation of building energy consumption and performance: a review, Visualization in Engineering. 6 (2018) 5.

[19] Q. Li, Q. Meng, J. Cai, H. Yoshino, A. Mochida, Applying support vector machine to predict hourly cooling load in the building, Applied Energy. 86 (2009) 2249–2256.

[20] Q. Li, Q. Meng, J. Cai, H. Yoshino, A. Mochida, Predicting hourly cooling load in the building: A comparison of support vector machine and different artificial neural networks, Energy Conversion and Management. 50 (2009) 90–96.

[21] Xuemei L, Lixing D, Yan L, Gang X, Jibin L. Hybrid genetic algorithm and support vector regression in cooling load prediction. Knowledge Discovery and Data Mining, 2010 WKDD'10 Third International Conference on: IEEE; 2010. p. 527-31.

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

[23] Liu D, Chen Q. Prediction of building lighting energy consumption based on support vector regression. 2013 9th Asian Control Conference (ASCC)2013. p. 1-5.

[24] Z. Wang, Y. Wang, R. Zeng, R.S. Srinivasan, S. Ahrentzen, Random Forest based hourly building energy prediction, Energy and Buildings. 171 (2018) 11–25.

[25] G.K.F. Tso, K.K.W. Yau, Predicting electricity energy consumption: A comparison of regression analysis, decision tree and neural networks, Energy. 32 (2007) 1761–1768.

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

[27] C. Fan, F. Xiao, S. Wang, Development of prediction models for next-day building energy consumption and peak power demand using data mining techniques, Applied Energy. 127 (2014) 1–10.

[28] X. Wang, Y. Sun, D. Luo, J. Peng, Comparative study of machine learning approaches for predicting short-term photovoltaic power output based on weather type classification, Energy. 240 (2022).

[29] Chen T, Guestrin C. XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Francisco, California, USA: Association for Computing Machinery; 2016. p. 785–94.

[30] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, et al., Lightgbm: A highly efficient gradient boosting decision tree, Advances in neural information processing systems. 30 (2017) 3146–3154.

[31] Prokhorenkova L, Gusev G, Vorobev A, Dorogush AV, Gulin A. CatBoost: unbiased boosting with categorical features. arXiv preprint arXiv:170609516. 2017.

[32] C. Miller, P. Arjunan, A. Kathirgamanathan, C. Fu, J. Roth, J.Y. Park, et al., The ASHRAE Great Energy Predictor III competition: Overview and results, Science and Technology for the Built Environment. 26 (2020) 1427–1447.

[33] S. Hochreiter, J. Schmidhuber, Long Short-Term Memory, Neural Computation. 9 (1997) 1735–1780.

[34] L. Sehovac, K. Grolinger, Deep Learning for Load Forecasting: Sequence to Sequence Recurrent Neural Networks With Attention, IEEE Access. 8 (2020) 36411–36426.

[35] C.H. Kim, M. Kim, Y.J. Song, Sequence-to-sequence deep learning model for building energy consumption prediction with dynamic simulation modeling, Journal of Building, Engineering. 43 (2021).

[36] W. Kong, Z.Y. Dong, Y. Jia, D.J. Hill, Y. Xu, Y. Zhang, Short-Term Residential Load Forecasting Based on LSTM Recurrent Neural Network, IEEE Transactions on Smart Grid. 10 (2019) 841–851.

[37] Y. Gao, Y. Ruan, C. Fang, S. Yin, Deep learning and transfer learning models of energy consumption forecasting for a building with poor information data, Energy and Buildings. 223 (2020).

[38] T.-Y. Kim, S.-B. Cho, Predicting residential energy consumption using CNN-LSTM neural networks, Energy. 182 (2019) 72–81.

[39] X. Shao, C.-S. Kim, P. Sontakke, Accurate Deep Model for Electricity Consumption Forecasting Using Multi-channel and Multi-Scale Feature Fusion CNN–LSTM, Energies. 13 (2020).

[40] C. Fan, Y. Sun, F. Xiao, J. Ma, D. Lee, J. Wang, et al., Statistical investigations of transfer learning-based methodology for short-term building energy predictions, Applied Energy. 262 (2020).

[41] Z.C. Lipton, The Mythos of Model Interpretability: In machine learning, the concept of interpretability is both important and slippery, Queue. 16 (2018) 31–57.

[42] R. Hamon, H. Junklewitz, I. Sanchez, Robustness and explainability of artificial intelligence, Publications Office of the European Union, 2020.

[43] Chen Z, Xiao F, Guo F, Yan J. Interpretable machine learning for building energy management: A state-of-the-art review. Advances in Applied Energy. 2023;9.

[44] Y. Gao, Y. Ruan, Interpretable deep learning model for building energy consumption prediction based on attention mechanism, Energy and Buildings. 252 (2021).

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

[46] Ribeiro MT, Singh S, Guestrin C. " Why should i trust you?" Explaining the predictions of any classifier. Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining2016. p. 1135-44.

[47] S.M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, Advances in neural information processing systems. 30 (2017).

[48] Wastensteiner J, Weiss TM, Haag F, Hopf K. Explainable AI for tailored electricity consumption feedback–an experimental evaluation of visualizations. arXiv preprint arXiv:220811408. 2022.

[49] M. Zdravković, I. Ćirić, M. Ignjatović, Explainable heat demand forecasting for the novel control strategies of district heating systems, Annual Reviews in Control. 53 (2022) 405–413.

[50] X. Jin, F. Xiao, C. Zhang, A.G.E.I.N. Li, An interpretable benchmarking framework towards all building types based on machine learning, Energy and Buildings. 260 (2022).

[51] X. Chang, W. Li, J. Ma, T. Yang, A.Y. Zomaya, Interpretable machine learning in sustainable edge computing: A case study of short-term photovoltaic power output prediction, in: ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP): IEEE, 2020, pp. 8981–8985.

[52] A. Bellahsen, H. Dagdougui, Aggregated short-term load forecasting for heterogeneous buildings using machine learning with peak estimation, Energy and Buildings. 237 (2021).

[53] R.J. Hyndman, X. Liu, P. Pinson, Visualizing Big Energy Data: Solutions for This Crucial Component of Data Analysis, IEEE Power and Energy Magazine. 16 (2018) 18–25.

[54] W.S. McCulloch, W. Pitts, A logical calculus of the ideas immanent in nervous activity, The bulletin of mathematical biophysics. 5 (1943) 115–133.

[55] C. Cortes, V. Vapnik, Support-vector networks, Machine Learning. 20 (1995) 273–297.

[56] L. Breiman, Random Forests, Machine Learning. 45 (2001) 5–32.

[57] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, The journal of machine learning research. 15 (2014) 1929–1958.

[58] Dietterich TG. Ensemble Methods in Machine Learning. Berlin, Heidelberg: Springer Berlin Heidelberg; 2000. p. 1-15.

[59] T.A. Reddy, I. Maor, C. Panjapornpon, Calibrating Detailed Building Energy Simulation Programs with Measured Data—Part II: Application to Three Case Study Office Buildings (RP-1051), HVAC&R Research. 13 (2007) 243–265.

[60] M.M. Mukaka, Statistics corner: A guide to appropriate use of correlation coefficient in medical research, Malawi Med J. 24 (2012) 69–71.

[61] National Centers for Environmental Information. Integrated Surface Database. 2021. Available: https://www.ncei.noaa.gov/products/land-based-station/integrated-surface-database.

[62] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, et al., Scikit-learn: Machine learning in Python, the Journal of machine Learning research. 12 (2011) 2825–2830.

[63] Chen T, Guestrin C. XGBoost: A scalable tree boosting system In Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,(pp. 785–794). New York, NY, USA: ACM. 2016;10.

[64] Chollet F. Keras. 2015. Available: https://github.com/fchollet/keras.

[65] Bisong E. Google Colaboratory. Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners. Berkeley, CA: Apress; 2019. p. 59-64.

[66] Elsayed S, Thyssens D, Rashed A, Jomaa HS, Schmidt-Thieme L. Do we really need deep learning models for time series forecasting? arXiv preprint arXiv:210102118. 2021.

[67] M. Farrokhabadi, Day-ahead electricity demand forecasting: Post-COVID paradigm, IEEE DataPort. (2020).

[68] Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv preprint arXiv:14126980. 2014.