https://dx.doi.org/10.1016/j.apenergy.2023.121481

Large-scale comparison and demonstration of continual learning for adaptive data-driven building energy prediction

Ao Li[1], Chong Zhang[1,2], Fu Xiao[1, 3*], Cheng Fan[4,5], Yang Deng[6], Dan Wang[6]

[1] Department of Building Environment and Energy Engineering, The Hong Kong Polytechnic University, Hong Kong, China

[2] School of Architecture & Urban Planning, Huazhong University of Science and Technology, Wuhan, China

[3] Research Institute for Smart Energy, The Hong Kong Polytechnic University, Hong Kong, China

[4] Key Laboratory for Resilient Infrastructures of Coastal Cities (Shenzhen University), Ministry of Education, China

[5] Sino-Australia Joint Research Center in BIM and Smart Construction, Shenzhen University, Shenzhen, China

[6] Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China

Abstract

Data-driven models have been increasingly employed in smart building energy management. To avoid performance degradation over time, data-driven models need to be continually updated to adapt to the changes of building operations. However, several critical issues in the model update process raised wide concerns, especially the concept drift and catastrophic forgetting issues. The concept drift issue happens when the statistical properties of target variable change over time in unforeseen ways. The catastrophic forgetting issue refers to the process that the previously learnt knowledge or patterns may be diluted and eventually lost in model update. Although a few model update methods were proposed, there is a lack of comprehensive comparison of the methods for adaptive data-driven building energy prediction. This paper conducted a comprehensive investigation on the performance of three conventional model update methods and five emerging continual learning methods using 2-year data of 100 buildings extracted from open-source dataset. The results show that continual learning methods are more effective in ensuring long-term accuracy while cutting down on the computation time and data storage expenses. The CV-RMSE of Elastic weight consolidation and Gradient episodic memory decreased by around 14% and 8% on average compared with static model and accumulative learning. The comparison results are valuable to the development of adaptive data-driven building energy prediction models which are more reliable over time and robust against changing operation conditions, thus more practically applicable in smart building energy management.

Keywords: Building energy prediction, Model update, Continual learning, Accumulative learning, Incremental learning

1. Introduction

Machine learning is one of the most rapidly growing data-driven technical domains, which sits at the nexus of computer science and statistics, and forms the foundation of Artificial Intelligence and data science [1]. Machine learning-empowered data-driven models have been used in all aspects of smart building energy management applications, including building design optimization [2], building energy prediction [3], fault detection and diagnosis [4], and building retrofit analysis [5]. Numerous studies have demonstrated that machine learning models are capable of achieving comparable, even higher performance while requiring less expert knowledge and building physical information compared to white-box and gray-box models [2,6]. The application of data-intensive machine learning methods in smart building management leads to more evidence-based decision-making [7].

The deployment of machine learning models in smart buildings is not a one-and-done process. Due to the ever-changing working conditions and very different system characteristics under different conditions, the relation between the input and output of the models may change over time. As a result, the previously learnt models may not be able to produce accurate predictions on the new datasets [8,9]. This issue, known as concept drift or data drift, commonly exists in the deployment of machine learning models and has been recognized as the primary cause of performance degradation of data-driven models [10]. Concept drift can be classified into several categories, including sudden drift where the data changes suddenly (e.g., sudden machine failures), incremental and gradual drift (e.g., change of occupancy behavior), and reoccurring drift (e.g., seasonality of meteorological conditions and working patterns) [9]. To capture the changes of system characteristics and prevent performance degradation, continual model update (or retraining) is proposed. Two widely-used model update methods are accumulative learning and incremental learning, which update the model parameters based on the gradient descent algorithm (or its variants) as new data become available without altering the model architecture [11]. In this paper, with the most recent model update serving as the dividing line, the preceding and succeeding data are called the historical dataset and incoming dataset, respectively. The accumulative learning method fine-tunes or retrains a data-driven model using the integrated datasets of historical and the incoming datasets, while the incremental learning method only uses the latter. Some researchers have investigated the application of these two methods in smart building management, e.g., building energy prediction. Yang et al. [12] developed several adaptive Artificial Neural Network-based building energy prediction models based on accumulative learning and incremental retraining. The adaptive models showed enhanced capability of adapting to unexpected pattern changes in the incoming data compared with the static model. Fekri et al. [11] proposed an online adaptive RNN model based on incremental learning for building load forecasting. The online fine-tuning was activated once the model performance degradation exceeded a predefined threshold. The proposed approach was evaluated with data from five individual homes, and the results showed that the proposed approach achieved higher accuracy than the standalone offline RNN model. Generally, the existing model update methods adopt fine-tuning more frequently [11,13] than retraining the model from scratch [12,14].

Another challenging issue with updating data-driven models is catastrophic forgetting, i.e., the knowledge learnt from the historical dataset may be forgotten after the model is updated on newly collected data [15,16]. For instance, Deng's study showed that a model trained using data from the spring of the first year could have a poor performance in the spring of the second year since it was updated with data from summer to winter in the first year [9]. As a lightweight approach, incremental learning cannot address the catastrophic forgetting issue in principle. The model parameters before and after fine-tuning on the incoming dataset might be significantly changed, as the distribution of successive subsets can be drastically different. Accumulative learning can alleviate the catastrophic forgetting issue but requires high computation resources as the data used for model update accumulates substantially over time. As Internet of Things (IoT) technology progresses, a growing number of data-driven models are deployed at the network edge level or directly in the IoT devices to handle low-level tasks [8,17]. The limited computation resources at edge devices pose new challenges to the online data-driven model deployment with model update.

Conventional model update methods cannot address the concept drift and catastrophic forgetting issues in an effective manner. As a new subfield of machine learning, continual learning (also referred to as lifelong learning) concentrates on learning from a continuous data stream which "can stem from changing input domains or can be associated with different tasks" [18]. Continual learning has recently drawn increasing attention as a promising solution for concept drift and catastrophic forgetting issues [13,16]. For instance, regularization-based continual learning methods preserve learned knowledge when acquiring new knowledge by protecting the model parameters that are essential for previous prediction tasks in the model update process. This strategy is more efficient and less resource-intensive than accumulative learning, since it only needs to memorize the essential model parameters for previous tasks, rather than all historical data. Zhou et al. [20] proposed an adaptive building load prediction model based on a continual learning method, elastic weight consolidation. With much-reduced computation time and data storage, the continual learning-based model showed similar accuracy as the accumulative learning-based one, and outperformed the incremental learning and static models. Research proved that continual learning is promising to speed up the transformation of buildings in the era of pervasive Artificial Intelligence [9,20,21].

The primary goal of model update is to strike a balance between acquiring new knowledge and memorizing prior knowledge. Therefore, the performance of model update method/strategy is highly influenced by the building-specific concept drift types (e.g., gradual drift, seasonal drift), which vary with buildings. However, the data used in previous research are usually limited to a single building. The research results cannot comprehensively reflect the performance of various model update methods under different conditions (i.e., concept drift type). That is, the influence of concept drift type has not been in-depth evaluated in previous research, which limited their research contributions and generalizability. To the best of the authors' knowledge, there is a lack of studies to systematically compare the performance of different online data-driven model update methods in machine learning-empowered smart building management.

To this end, this study comprehensively investigates different online update methods for adaptive short-term building energy prediction. The main contributions of this research are listed below:

1. Three conventional methods (i.e., accumulative learning, incremental learning, ensemble learning) and five emerging continual learning methods (i.e., elastic weight consolidation, less-forgetting learning, synaptic intelligence, memory replay and gradient episodic memory) are tested and compared. The model update methods are compared in terms of both prediction accuracy and computation resources required.
2. Considering the diversity of buildings, the comparison study is conducted on one hundred buildings taken from an open-source dataset for comprehensive and generalizable results. The impact of model update frequency is also investigated by adopting different frequencies to update the model.
3. The model performances under different concept drifts are further analyzed by clustering all tested buildings according to their monthly-average electricity consumption profiles. The results intuitively reveal the characteristics of different model update methods.

The findings and conclusions obtained from this study can facilitate the selection of proper model update method for smart building energy management. The remaining part of the paper is constructed as follows. An overview of continual learning methods is provided in Section 2. Section 3 introduces the research methodology. The comparison study results and discussion are elaborated in Section 4. Section 5 concludes the paper.

2. Overview of continual learning

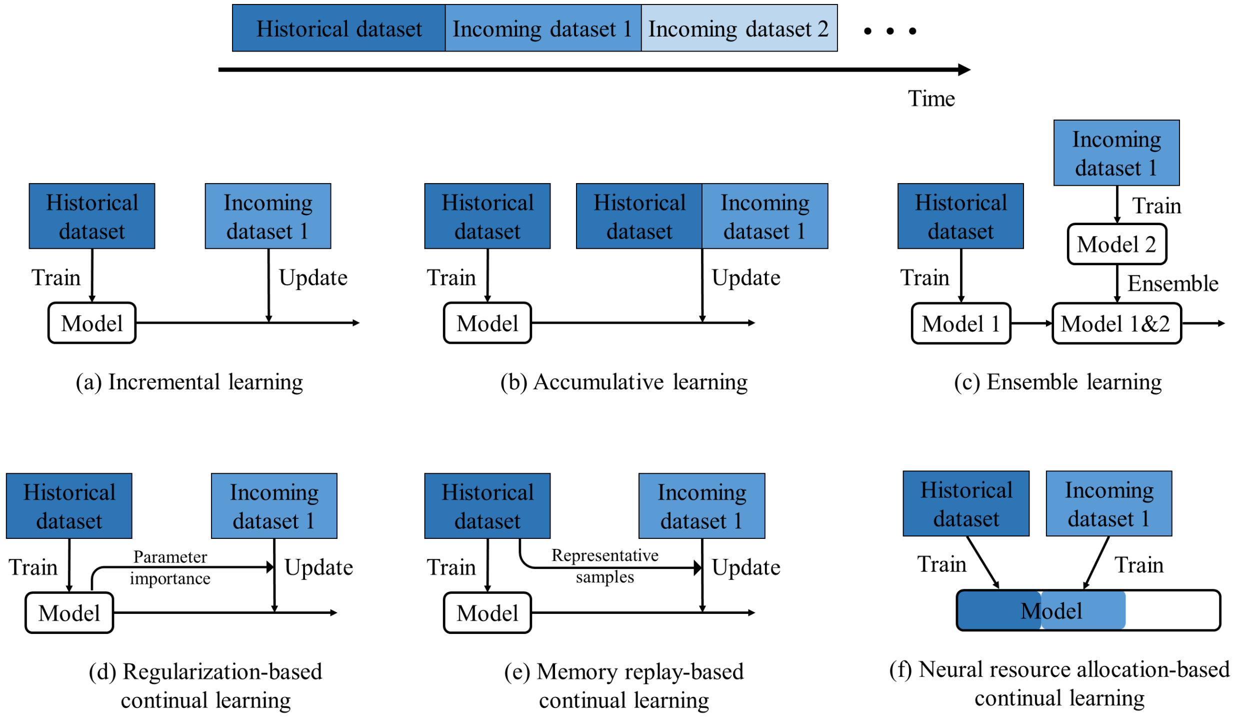


Figure 1. Diagram of data-driven model update methods

This section provides an overview of the theoretical background of continual learning, and the working principles of several representative methods tested in this study. Figure 1 shows typical update processes for data-driven models on a continuous data stream. Depending on the timesteps at which the model is updated, the continuous data stream can be divided into a number of subsets, i.e., historical dataset and incoming datasets. As illustrated in Section 1, the conventional gradient descent-based update methods (i.e., accumulative learning and incremental learning) have their limitations in tackling the concept drift issue. Accumulative learning is time-consuming and necessitates large data storage, while incremental learning cannot prevent catastrophic forgetting issue. Ensemble learning is another option which trains a separate model for each subset and then ensemble them together [22,23]. The model update time is much reduced compared with accumulative learning as only the incoming dataset is used, and storing previous models typically takes less memory than storing all historical data. The weights associated with each model can be set equally or updated based on the global prediction error [24]. The drawback of the ensemble learning strategy is also obvious, i.e., the ensemble model will grow in size over time. This is unacceptable for online optimization tasks that require fast model inference.

To address concept drift and catastrophic forgetting simultaneously, the data-driven models must have the capacity to acquire new knowledge and refine existing knowledge on the basis of continuous input (i.e., plasticity), while preventing the new input from significantly interfering with existing knowledge (i.e., stability) [19]. For instance, the neural weights of an artificial neural network can be regarded as a type of knowledge learnt from training dataset. The stability-plasticity dilemma is a well-known constraint for artificial neural systems [17,19]. Continual learning is a subfield of machine learning which aims to strike a balance between stability and plasticity. Based on the working principle, continual learning methods can be mainly divided into three categories: regularization-based, memory replay, and neural resource allocation [10,16,18,19].

Regularization-based continual learning methods alleviate the catastrophic forgetting issue by imposing different constraints on the updated model parameters according to their importance to previous tasks. That is, the parameters more important to previous tasks are protected or frozen in the update process, while the less-critical parameters are assigned greater plasticity, which can be modified in an extensive range. Widely adopted regularization-based methods include elastic weight consolidation [15], synaptic intelligence [25], and learning without forgetting [26]. Elastic weight consolidation (EWC), proposed by DeepMind, is one of the most widely used continual learning methods [15]. EWC adds a quadratic penalty on the difference between the parameters of previous and new models in the loss function, which inhibits the finetuning for task-relevant weights coding for previously learned knowledge. The loss function of EWC can be expressed by $L'(\theta) = L(\theta) + \lambda \sum_i b_i \, (\theta_i - \theta_i^b)^2$, where $L(\theta)$ represents the mismatch between predicted values and actual labels; θ_i are the updated model parameters; and θ_i^b are the model parameters learned from previous task. b_i represents the importance degree of parameter i on previous task(s), which is calculated based on Fisher information matrix. λ is a hyperparameter that indicates the relative importance of historical knowledge and new knowledge.

Synaptic intelligence (SI) is proposed by Zenke et al. [25] to alleviate the catastrophic forgetting issue by allowing individual synapses (i.e., model parameters, including weights between layers as well as biases) to estimate their importance for solving a learned task. Similar to elastic weight consolidation, synaptic intelligence penalizes changes of influential parameters so that new tasks can be learned with minimal forgetting. However, Synaptic intelligence computes the synaptic relevance in an online manner and over the entire learning trajectory in parameter space, whereas EWC synaptic importance is computed offline as the Fisher information at the minimum of the loss for a designated task. Experiments revealed that SI and EWC yielded similar performance (e.g., on the permuted MNIST benchmark [27]), yet they may exhibit different characteristics [16,27].

Memory replay of representative old training samples (referred to as memory replay) is another category of continual learning methods [28]. As the name implies, this strategy updates the model using sampled data from both the incoming dataset and the replay memory. The model's capacity to remember the previous knowledge can be modified by adjusting the sampling ratio or the sample weights of old task(s) and new task. The data samples of previous tasks can be reserved with either a constant memory size for each task, or a fixed capacity for all tasks. Memory replay has been shown as an effective solution, and achieves great performance for image classification [29].

Gradient episodic memory (GEM) is a special method in the memory replay branch, which yields beneficial transfer of knowledge to previous tasks when updating the model. The main feature of GEM is an episodic memory M_i, which stores a subset of the observed examples from task i (to calculate the loss gradient). GEM seeks to reduce the loss of the current task without increasing the loss of the previous task(s). While minimizing the loss on current task t, GEM treats the losses on the episodic memories of tasks $k < t$ as inequality constraints, avoiding their increase but allowing their decrease. Lee et al. [21] adopted GEM to improve and accelerate the learning performance of a multi-client power consumption prediction model deployed on an edge-cloud system, and reduce computation resources and alleviate hardware loads. The proposed method was robust to dynamically changed data features and time-variant stream data.

Neural resource allocation is also referred to as dynamic architecture. As optimizing the entire network on each task/subset will lead to catastrophic forgetting, neural resource allocation dynamically provides a different sub-network (or separate neural resources) for each task, e.g., re-training with an increased number of neurons or network layers. Rusu et al. [30] proposed Progressive network, which blocks any changes to the network trained on previous knowledge and expands the architecture by allocating novel sub-networks (with fixed capacity) to be trained with the new information. The learned parameters for the existing task(s) are left unchanged, while the new parameter set is learned for the incoming dataset/task. Intuitively, this method prevents catastrophic forgetting but leads the complexity of the architecture to grow with the number of learned tasks, or the number of batches of incoming stream data. Instead of continuously expanding the model architecture, Mallya and Lazebnik [31] proposed PackNet, which adopts network pruning techniques to gradually allocate/distribute unused neural recourses of a fixed model architecture to new tasks. There appeared some research works [32,33] which leveraged the principles of both Progressive network [30] and PackNet [31]. Briefly speaking, continual learning methods compromise incremental learning and accumulative learning to strike a better balance in the stability-plasticity dilemma.

3. Research methodology

3.1 Outline of research methodology

This research aims to comprehensively investigate different continual learning methods for adaptive data-driven building hourly electricity consumption prediction. The overall research methodology is shown in Figure 2, which mainly consists of three steps, i.e., data preprocessing, performance evaluation of model update methods, and post-analysis on building-specific concept drift differences. All the data used for the case study are extracted from a public-available benchmarking building dataset. Data preprocessing consists of data cleaning (e.g., filling in missing values, outlier detection) and feature selection. The entire dataset is segmented into several subsets based on the model update setting (i.e., length of the historical dataset, and update frequency). Then, a baseline neural network model is developed for building energy prediction. The model performance under a continuous data stream using different model update methods is evaluated regarding prediction accuracy and computation resources (required for model update). Further investigation on the stability-plasticity dilemma under different data drift types is conducted by clustering the tested buildings according to their energy consumption profiles.

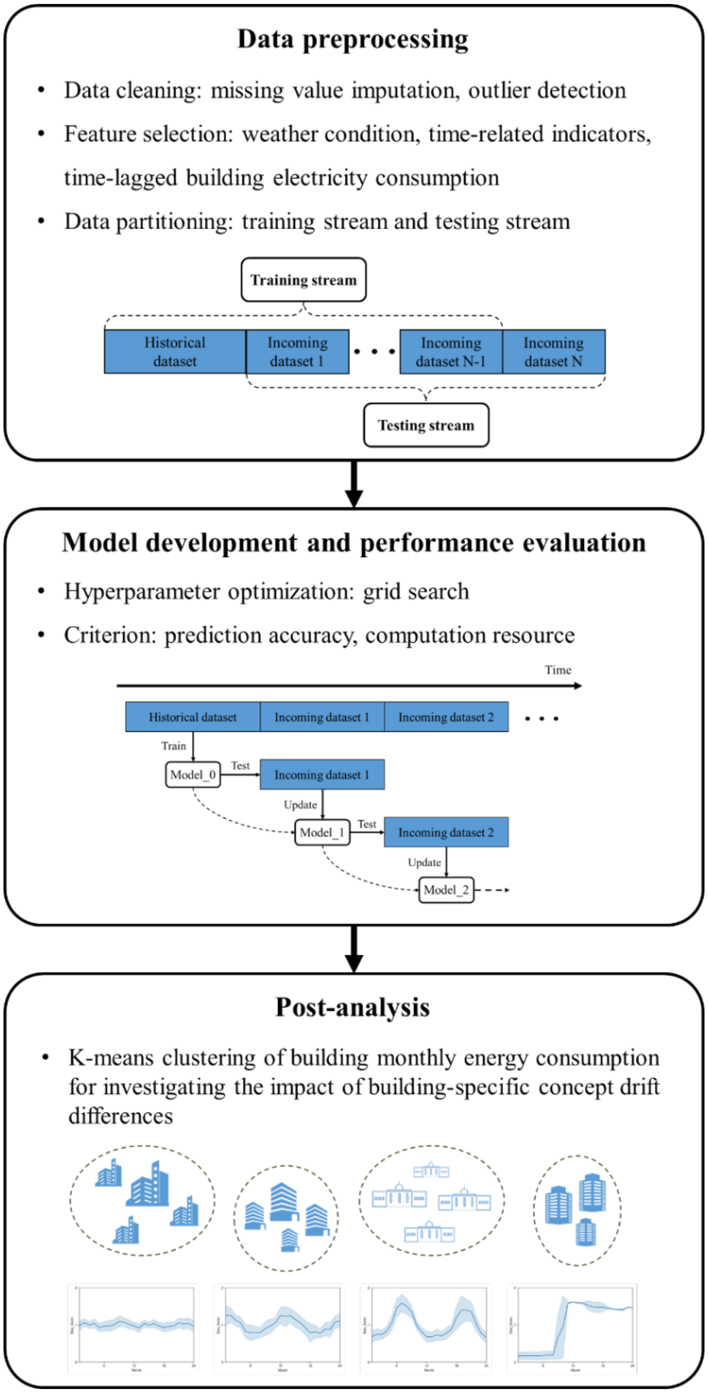

Data preprocessing
• Data cleaning: missing value imputation, outlier detection
• Feature selection: weather condition, time-related indicators, time-lagged building electricity consumption
• Data partitioning: training stream and testing stream
Training stream
Historical dataset
Incoming dataset 1
Incoming dataset N-1
Incoming dataset N
Testing stream
Model development and performance evaluation
• Hyperparameter optimization: grid search
• Criterion: prediction accuracy, computation resource
Time
Historical dataset
Incoming dataset 1
Incoming dataset 2
Train
Model_0
Test
Incoming dataset 1
Update
Model_1
Test
Incoming dataset 2
Update
Model_2
Post-analysis
• K-means clustering of building monthly energy consumption for investigating the impact of building-specific concept drift differences

Figure 2. Research outline

3.2 Data source and data preprocessing

The data used in this study is extracted from Building Data Genome Project 2 (BDGP2) [34]. BDGP2 is an open-source building dataset, consisting of more than one thousand non-residential buildings with a range of two years (2016 and 2017) at an hourly frequency. In this dataset, the parameters for each building mainly include building information (e.g., area, size, usage, year of construction), energy usage (e.g., electricity consumption, hot water) and climate conditions (e.g., temperature, humidity, wind speed and direction). To comprehensively evaluate the model update methods under various circumstances, one hundred buildings' data are randomly selected from this BDGP2. Data cleaning is performed to improve data quality, including filling in missing values and outlier detection based on statistical criteria. In this study, three main categories of features are selected as the input of the building energy prediction model, including weather features, time-related indicators, and time-lagged electricity consumption. Detailed information of input features is presented in Table 1. Min-max normalization is conducted for each feature to improve the numerical stability of the prediction model.

Table 1. Description of the input features

Input features		**Description**
Weather features	$T_{dry-bulb}$	Dry-bulb temperature
	T_{dew}	Dew temperature
Time-related indicators	M	Month (1,2,…,12)
	$Weekday$	Day of week (1,2,…,7)
	H	Hour (1,2,…,24)
Time-lagged electricity consumption	$Elec_{t-24}$	Electricity power in 24 hours ago
	$Elec_{yes}$	Total electricity consumption of yesterday

3.2 Performance evaluation of model update methods

The most straightforward Artificial neural network, i.e., multilayer perceptron (MLP), is adopted as the prediction model. Based on grid-search in preliminary test, the number of hidden layers, neuron number and activation function of each hidden layer are determined as 2, 30 and ReLU [35], respectively.

As shown in Table 2, eight online model update methods are tested, including three conventional methods (i.e., accumulative learning, incremental learning, and ensemble learning) and five continual learning methods (i.e., elastic weight consolidation, synaptic intelligence, less-forgetting learning, memory replay, and gradient episodic memory). The static model, which is trained on the historical dataset and never updated, serves as the baseline case.

Table 2. Description of the model update methods tested in this study, including static model as baseline, three conventional model update methods and five continual learning methods

Method	**Description**
Static model	The model is trained on the historical dataset and never updated.
Accumulative learning (AR)	Update the model using incoming data and all historical data.
Incremental learning (IL)	Update the model using only the incoming data.
Ensemble learning (EL)	Ensemble of models trained on each subset in an average manner.
Elastic weight consolidation (EWC)	Regularization-based continual learning method [15]. A separate penalty for each previous task/subset is kept.
Synaptic intelligence (SI)	Regularization-based continual learning method [25].
Less-forgetting learning (LFL)	Regularization-based continual learning method [36].
Memory replay (MR)	Memory replay method [37].
Gradient episodic memory (GEM)	Memory replay method [38].

This comparison study adopts the straightforward periodic update strategy, which updates the model on a regular basis. The whole data stream can be separated into chronological sequences of subsets $D_{All} = \{D_1, D_2, \dots, D_i\}$. D_1 is the historical dataset utilized for training the static model, and D_i $(i \geq 2)$ is the continual in-coming datasets. The update frequency is the length of subsets D_i . In this study, the length of the entire data stream from each building is two years. The length of the historical dataset D_1 is set as two months. Two update frequencies, i.e., one month and two months, are tested. The hyperparameters of each model update method are determined based on preliminary grid-search tests on one example building, and used for all other buildings. The settings and results of the grid search are shown in the Appendix. All the models and methods are tested using Python programming language, as well as PyTorch [39] and Avalanche [40] packages.

Data permutation experiment is widely adopted to compare different model update methods in regression tasks including data-driven time-series prediction modeling, i.e., training a model with a dataset along with a permuted version of the same dataset [41]. However, it is not applicable to building energy prediction, as the building dataset is in chronological order and should not be rearranged due to the inherent temporal correlation of the building operations. Therefore, this study increases the variety of data streams by testing on multiple buildings. An accuracy matrix $R \in \mathbb{R}^{T \times T}$ is constructed, where $R_{i,j}$ represents the model prediction accuracy (i.e., RMSE and CV(RMSE)) on subset D_j after update on subset D_i, $i, j \in [1, T]$. The overall accuracy of static model and adaptive models on the whole data stream (of one building) can be calculated by the following equations:

$$R_{static} = \frac{1}{T-1} \sum_{i=1}^{T-1} R_{1,i+1} \tag{1}$$

$$R_{adaptive} = \frac{1}{T-1} \sum_{i=1}^{T-1} R_{i,i+1} \tag{2}$$

A relative ratio indicating the effect of model update method x can be derived by $R_{Norm} = R_{adaptive}/R_{static}$. Two criteria metrics are adopted to assess the model prediction accuracy and construct the accuracy matrix R, i.e., the root mean squared error (RMSE), and the coefficient of variation of the root mean squared error (CV-RMSE):

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(y_i - \widehat{y_i})^2}{n}} \tag{3}$$

$$CV - RMSE = \frac{\sqrt{\sum_{i=1}^{n} \frac{(y_i - \widehat{y_i})^2}{n}}}{\frac{\sum_{i=1}^{n} y_i}{n}} \tag{4}$$

Where y_i is the actual value, $\widehat{y_i}$ is the model predicted value (i.e., building hourly electricity consumption).

3.4 Post-analysis on building-specific concept drift differences

As mentioned in the previous section, data permutation lacks practical significance for datasets naturally generated in chronological order. On the other hand, previous studies did not investigate the impact of building-specific concept drift differences (e.g., sudden drift, gradual drift, and reoccurring drift) on the effectiveness/performance of various model update methods [20,21]. The main challenge is the absence of a widely-accepted indicator which can adequately quantify the data drift of building energy usage patterns. As an exploratory attempt, this study classifies the tested buildings according to their electricity consumption pattern. Considering information conservation and computation effectiveness, the time interval is set as one month (i.e., monthly average building electricity consumption). K-means clustering algorithm [42] is adopted here, which aims to partition all observations into k clusters while minimizing the within-cluster variances. The cluster number k is determined based on the Silhouette score [43]. The cluster-average results will be analyzed to investigate the impact of building-specific concept drift differences and provide insights for model update method selection.

4. Results and discussions

Section 4.1 summarizes the results on all tested buildings for systematic comparison of different model update methods. Section 4.2 presents the results for one example building to reveal the stability-plasticity dilemma and visualize the model performance on the continuous data stream with varied data characteristics. Section 4.3 provides the building clustering results and analyzes the cluster-average performance to investigate the building-specific concept drift impact. Section 4.4 provides a discussion on research results, limitations and future direction.

4.1 Results of all tested buildings

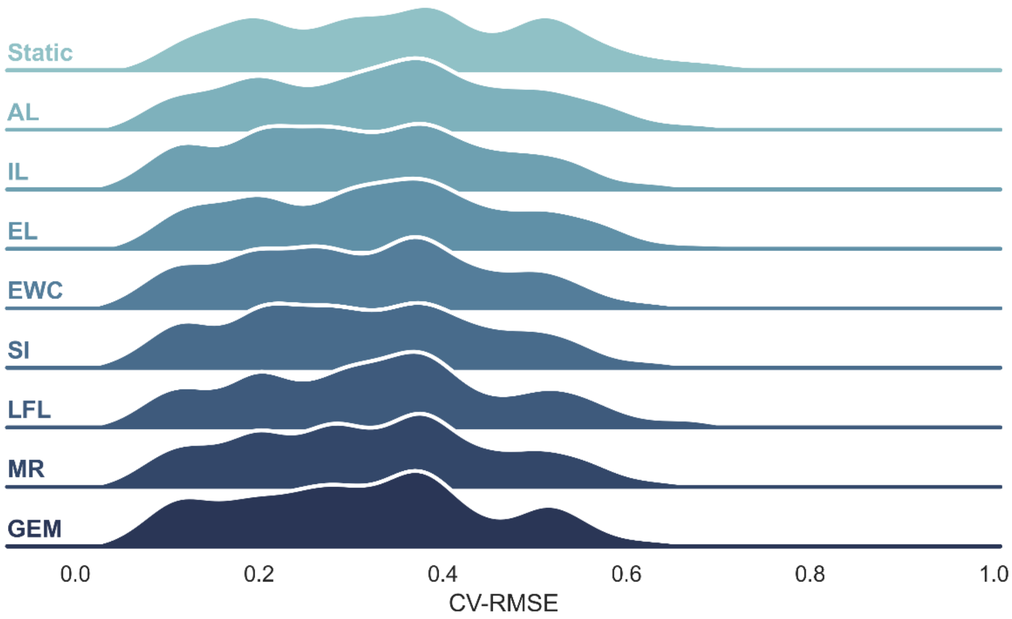


Figure 3. Model performance distribution (CV-RMSE) on all tested buildings (update frequency: one month)

Table 3. Statistical summary of model performance on all tested buildings

Setting 1: update frequency: one month										
Update method		Static	AL	IL	EL	EWC	SI	LFL	MR	GEM
$CV_{average}$		36.6%	34.4%	31.7%	34.0%	**31.6%**	31.7%	33.4%	32.6%	31.8%
R_{Norm}	Average	1	0.942	0.875	0.935	**0.868**	0.873	0.916	0.894	0.869
	Median	1	0.984	0.946	0.978	0.932	0.948	0.975	0.947	**0.923**
	Max	1	1.23	1.03	1.26	1.02	1.02	1.2	1.03	**1.01**
	Min	1	0.557	**0.257**	0.521	0.33	0.261	0.383	0.433	0.459
	Std	0	**0.114**	0.157	0.115	0.147	0.159	0.139	0.133	0.140
Improving ratio			76%	91%	87%	**98%**	95%	87%	93%	93%
Setting 2: update frequency: two months										
Update method		Static	AL	IL	EL	EWC	SI	LFL	MR	GEM
$CV_{average}$		36.6%	34.7%	33.8%	34.6%	33.6%	33.8%	34.1%	33.9%	**33.1%**
R_{Norm}	Average	1	0.948	0.928	0.945	0.918	0.927	0.932	0.928	**0.906**
	Median	1	0.981	0.979	0.976	0.967	0.981	0.979	0.972	**0.948**
	Max	1	1.27	1.2	1.28	1.32	1.22	**1.09**	1.23	1.22
	Min	1	0.548	**0.33**	0.548	0.459	0.336	0.479	0.517	0.466
	Std	0	**0.114**	0.144	0.111	0.141	0.144	0.121	0.126	0.134
Improving ratio			71%	71%	75%	78%	75%	75%	75%	**79%**
Setting 3: update frequency: four months										
Update method		Static	AL	IL	EL	EWC	SI	LFL	MR	GEM
$CV_{average}$		36.6%	35.0%	35.6%	34.4%	34.9%	35.4%	34.7%	35.1%	**33.4%**
R_{Norm}	Average	1	0.964	0.981	0.946	0.963	0.974	0.956	0.966	**0.923**
	Median	1	1.000	1.005	**0.979**	0.987	1.003	0.992	1.000	0.982
	Max	1	1.372	1.696	1.274	1.545	1.728	1.455	1.530	**1.258**
	Min	1	0.518	0.443	0.561	0.445	0.478	0.508	0.539	**0.433**
	Std	0	0.137	0.184	**0.117**	0.173	0.187	0.156	0.151	0.151
Improving ratio			49%	44%	**61%**	56%	46%	54%	48%	58%

Table 3 provides the statistical results on all tested buildings (under two update frequency settings) and Figure 3 shows the model performance distribution under one-month update frequency. A scale-independent indicator $CV_{average}$ is calculated by averaging the CV-RMSE prediction accuracy of each method on all tested buildings. The static model is regarded as the baseline in the comparison study. The prediction performance of adaptive models is normalized based on the static model, obtaining R_{Norm}. That is, a R_{Norm} less than one indicates that the update method achieves a performance improvement (over the static model). The improving ratio represents the proportion of buildings on which the update method achieves better prediction performance over the static model. For each indicator, the best results are highlighted in bold.

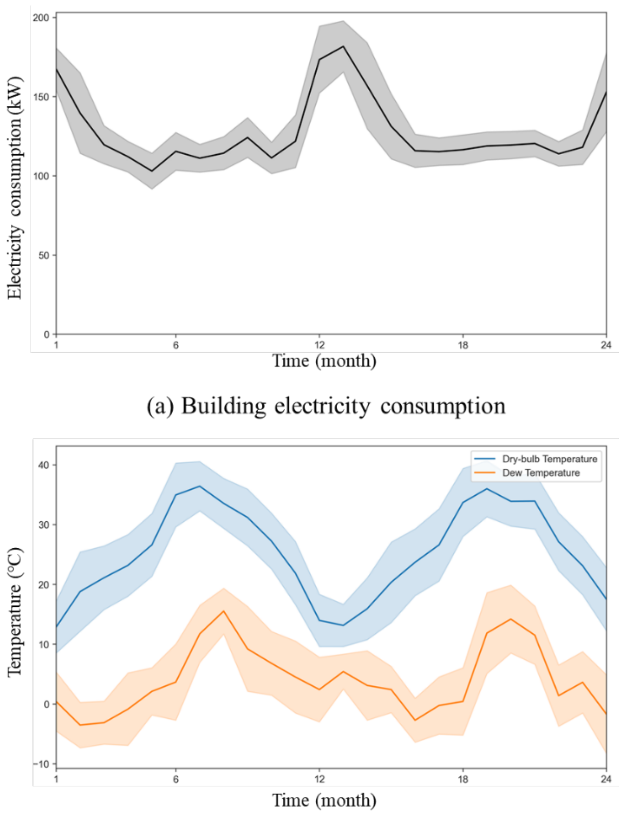
The results showed that, with the same prediction model architecture, all update methods improved the model performance to a pretty large extent. Compared with the static model, continual learning methods not only greatly improves the upper limit (i.e., min value of R_{Norm}), but also ensure the lower limit (i.e., max value of R_{Norm}). This reflects the reliability and robustness of continual learning methods. Increased update frequency further strengthened the performance of incremental learning and all continual learning methods, while this beneficial effect was less clear for accumulative learning and ensemble learning. Changing the update frequency from two months to one month can reduce the $CV_{average}$ by up to two percentages (i.e., IL and EWC) and increase the improving ratio by up to twenty per cent (i.e., IL, EWC and SI). when changing the update frequency from two months to four months, GEM still performs the best. Notably, the performance of Ensemble learning remains basically the same. This reflects the robustness of the ensemble learning method. The performance of other adaptive models has declined to varying degrees.

The static model achieved an average CV-RMSE of 36.6% on all tested buildings. It can be observed from the statistical results of R_{Norm} that, EWC and GEM outperform other methods overall. In setting 2, GEM improved the static model from 36.6% to 33.1% in terms of $CV_{average}$. In setting 1, EWC improved the model performance from 36.6% to 31.6% in terms of $CV_{average}$.

The performance of accumulative learning and ensemble learning are similarly mediocre. The low standard deviation of R_{Norm} indicates that these two methods perform stably with different buildings. Among the traditional methods, incremental learning is the most effective. It has achieved top overall performance under both settings. It is worth mentioning that, incremental learning obtained the smallest R_{Norm} value in both settings, which indicates the maximum amount by which the update method can outperform the static model. More detailed analysis of this phenomenon will be provided in the following section.

4.2 Results of one example building

This section provides a detailed elaboration on the model performance on an example building under the continuous data stream. A college classroom (building identity: Fox_education_Rosie) is selected from the benchmark dataset as an example building. This building is located in Phoenix, America. Figure 4 shows the monthly-average electricity consumption profile of this building and the local climate condition (i.e., dry-bulb temperature and dew temperature) throughout the whole period (from 2016 to 2017). The solid line and shaded area represent the average value and standard deviation, respectively. The local climate exhibits obvious seasonal patterns, i.e., hot summers and mild winters. The building's energy usage is relatively steady most of the time, with a significant increase during the winter months.



(a) Building electricity consumption

(b) Climate condition

Figure 4. The (a) monthly average electricity consumption and (b) climate condition of the example building throughout the whole time period (from 2016 to 2017)

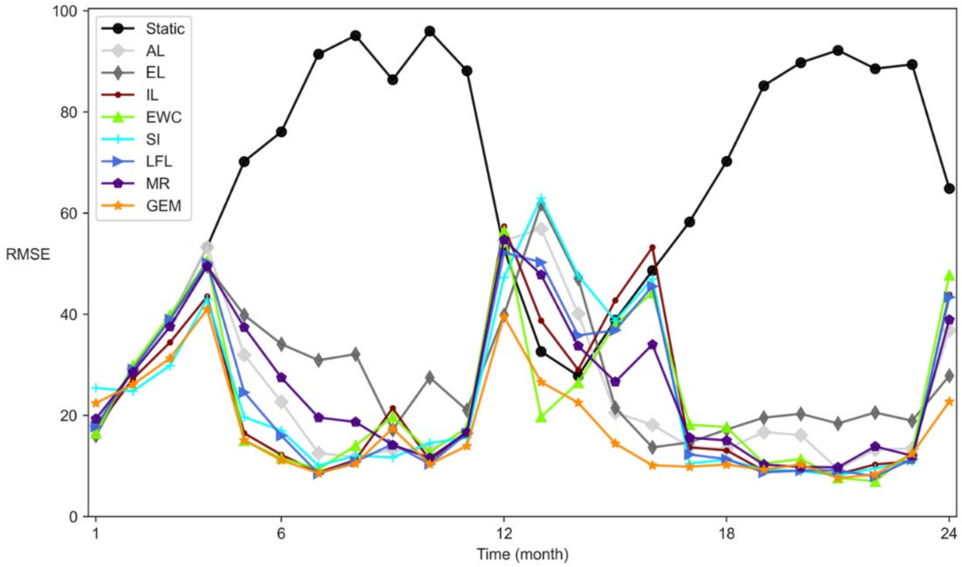


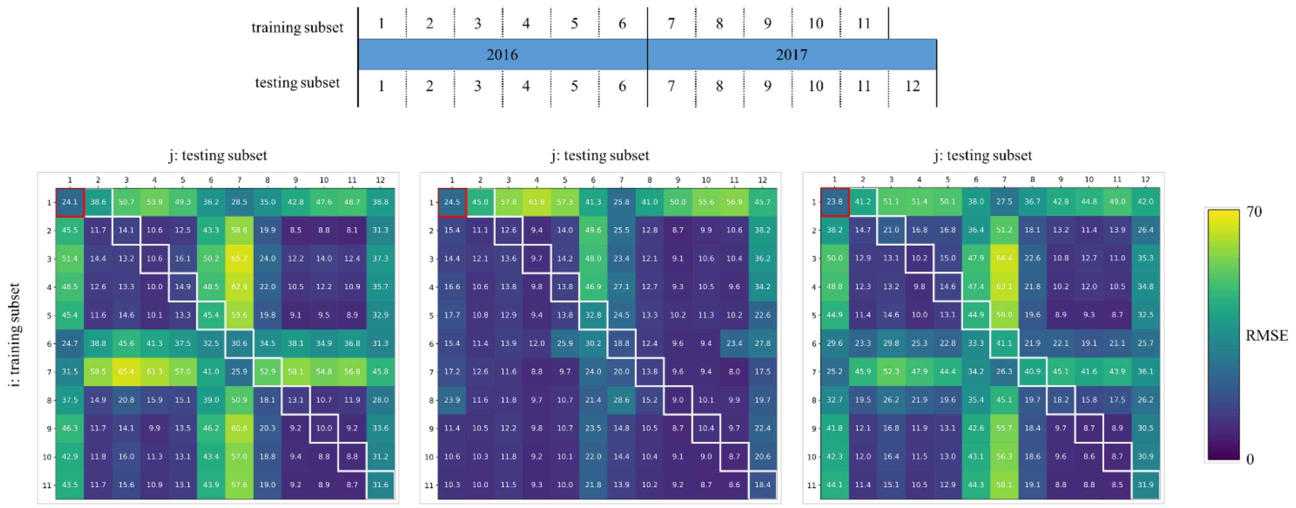
Figure 5. The monthly average RMSE of building energy predictions using different update methods in the whole time period. (Update frequency: one month).

Table 4. Model prediction accuracy over the whole time period with different update methods under varied model architecture and update frequency.

Setting 1: update frequency: one month									
Method	Static	AL	IL	EL	EWC	SI	LFL	MR	GEM
CV-RMSE	29.0%	18.2%	15.0%	16.5%	14.6%	15.1%	16.1%	15.6%	**13.5%**
RMSE	35.5	24.0	19.9	22.3	19.3	20.1	21.5	20.7	**18.2**
Setting 2: update frequency: two months									
Method	Static	AL	IL	EL	EWC	SI	LFL	MR	GEM
CV-RMSE	29.0%	19.4%	18.8%	18.9%	16.6%	17.6%	19.6%	17.9%	**13.4%**
RMSE	35.5	25.3	24.6	24.9	21.9	23.2	25.6	23.5	**17.5**
Setting 3: update frequency: four months									
Method	Static	AL	IL	EL	EWC	SI	LFL	MR	GEM
CV-RMSE	29.0%	24.4%	24.2%	24.9%	**23.4%**	24.0%	24.9%	23.8%	25.1%
RMSE	35.5	32.0	31.9	33.2	**30.9**	31.8	32.9	31.4	33.3

The results on the example building are summarized in Table 4. The trend/conclusions are similar to that in Table 3. Among all, accumulative learning and ensemble learning have the least amount of improvement over the static model. Elastic weight consolidation and Gradient episodic memory achieve the best performance. Although incremental learning cannot address the catastrophic forgetting problem, it generates better outcomes than several continual learning methods.

Figure 5 depicts the monthly-average model accuracy in terms of RMSE for different update methods. The performance profile of the static model clearly exhibits the concept drift phenomenon, i.e., its accuracy declines dramatically in the second half of both years. The most likely causes are the large discrepancies between the training and testing periods in terms of building electricity consumption patterns and climate conditions. In comparison, the continuously-updated models, no matter the update methods, perform better as they can adapt to the change of building electricity consumption pattern. However, for many adaptive models, the catastrophic forgetting phenomenon can be observed on the 13th and 14th months (i.e., the 7th testing subset). In these two months, the static model performs well while many continuously-updated models clearly forget the knowledge learned from the historical dataset (when acquiring new knowledge). It can be observed from Figure 5 that, the monthly-average performances of adaptive models are fairly similar. Among different model update methods, EWC and GEM are the most effective ones for alleviating the catastrophic forgetting problem. Particularly, the prediction model updated using GEM demonstrates excellent and consistent performance during the entire period. The CV-RMSE is 0.134, showing a 62% improvement over the static model.



(a) Incremental learning: $R_{average} = 24.6$ (b) Gradient episodic memory: $R_{average} = 17.5$ (c) Less-forgetting learning: $R_{average} = 25.6$

Figure 6. Accuracy matrix R_{ij} (RMSE) of (a) Incremental learning, (b) Gradient episodic memory, and (c) Less-forgetting learning. Update frequency: two months. The results marked with white boxes represent the adaptive model performance ($R_{i,i+1}, i \in [1,2,\dots,T-1]$).

Figure 6 presents the accuracy matrix R of incremental learning, gradient episodic memory and less-forgetting learning (update frequency: two months). The length of each training/testing subset is 2 months. The results of the first row (i.e., $R_{1,j}$, $j \in [1,2,\dots,12]$) represent the model performance after training on the historical dataset D_1 (i.e., the static model). The results marked with white boxes represent the model performance ($R_{i,i+1}, i \in [1,2,\dots,11]$) during the entire period. The variation in model performance on the same testing subset (e.g., D_1) reflects how effectively the model retains previously learned knowledge. For instance, it can be observed from the seventh column that (i.e., $R_{i,7}, i \in [1,2,3,4]$) of Figure 6(a) that the model initially performs well on subset D_7 after training on the historical dataset D_1 (i.e., $R_{1,7} = 28.5$), but the error increased fast due to the finetuning of model parameters on subsequent subsets. This indicates a catastrophic forgetting occurrence. On the other hand, as an effective continual learning method, GEM can help the prediction model retain the previously learned knowledge and perform well when encountering similar task(s), as shown in Figure 6(b). However, maintaining a strong anti-forgetting capability (over a long period) sacrifices the model 'plasticity', which may occasionally result in a net loss. For instance, less-forgetting learning shows improved anti-forgetting capability than incremental learning, as the model performance of memory replay on subset D_7 declines slower. However, the overall performance of less-forgetting learning (i.e., $R_{average} = 25.6$) is not as good as incremental learning (i.e., $R_{average} = 24.6$) on the whole period.

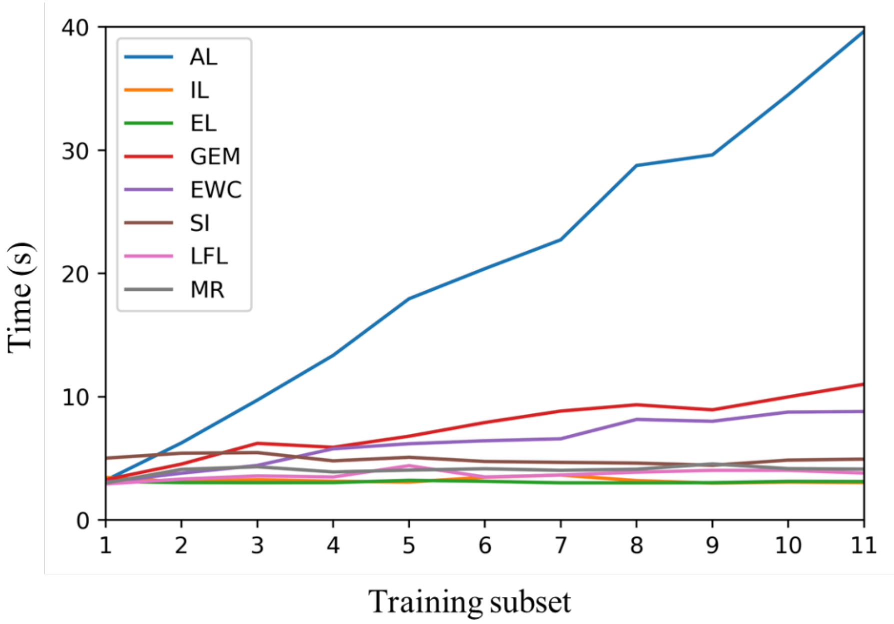


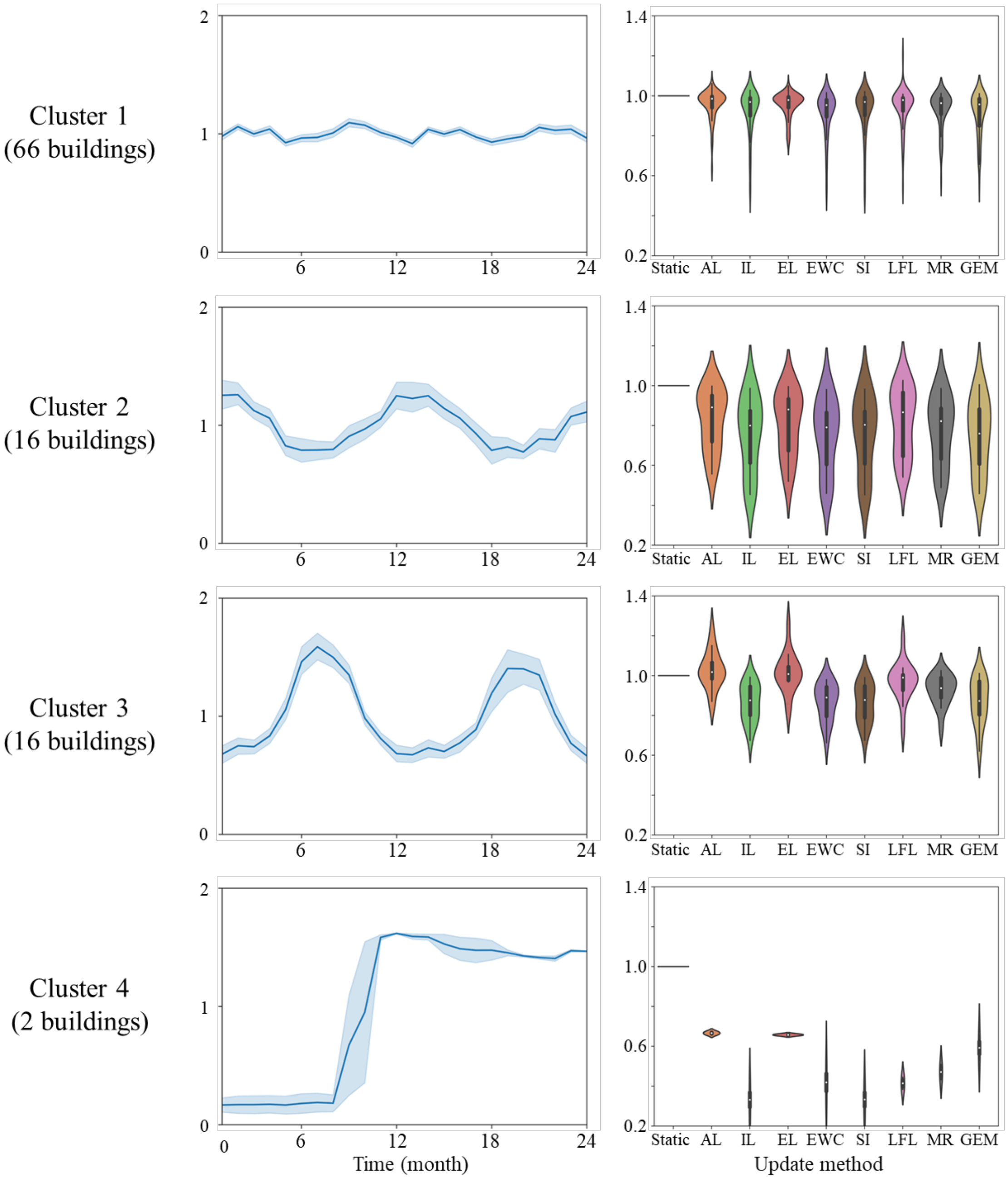
Figure 7. Model training time on each training subset

Figure 7 illustrates the model training time using different update methods on each training subset. The training epoch number is set as 20. An early stopping strategy is adopted with a patience of 3. Figure 7 shows that, the model training time for accumulative learning continues to increase (i.e., from 3.2s to 39.7s), since the volume of the training dataset grows continually. Ensemble learning, which performs similarly to accumulative learning, trades the model size for training time. The model training time required for IL, EL, SI, LFL and MR is almost stable throughout the training process. On the other hand, the model training times for GEM and EWC exhibit a similar upward trend as accumulative learning, with a smaller slope (from 3s to 11s).

4.3 Post-analysis on concept drift effect

To further demonstrate the stability-plasticity trade-off for different concept drift types, the buildings are classified via k-means clustering according to their monthly-average electricity consumption profiles. The cluster number k is identified as four based on the Silhouette coefficient.

Figure 8(a) shows the representative profiles of all building clusters. The solid line and shaded area represent the average value and standard deviation, respectively. The four clusters consist of 66, 16, 6 and 2 buildings, respectively. The statistical summary of model prediction performance is summarized in Table 5. The improvement percentages are calculated by $(CV_{average}^{static} - CV_{average}^{adaptive})/CV_{average}^{static\ model} \times 100\%$. The results of normalized prediction accuracy (i.e., R_{Norm}) are shown in Figure 8(b).



(a) Electricity consumption pattern

(b) R_{Norm} (CV-RMSE)

Figure 8. (a) Normalized monthly electricity consumption (average ± standard deviation); (b) Normalized prediction accuracy for each update method

Table 5. Prediction accuracy under different model update methods for buildings in each cluster. Settings: update frequency: two months.

Cluster 1 (building number: 66)									
Update method	Static	AL	IL	EL	EWC	SI	LFL	MR	GEM
$CV_{average}$	31.1%	30.2%	29.3%	30.0%	**29.0%**	29.3%	29.9%	29.5%	**29.0%**
Improvement		2.9%	5.8%	3.5%	6.8%	5.8%	3.9%	5.1%	6.8%
Average R_{Norm}	1	0.958	0.924	0.955	0.911	0.922	0.942	0.929	**0.908**
Improving ratio		78.8%	86.4%	95.5%	**97%**	92.4%	90.9%	90.9%	92.4%
Cluster 2 (building number: 16)									
Update method	Static	AL	IL	EL	EWC	SI	LFL	MR	GEM
$CV_{average}$	46.8%	38.2%	34.3%	37.2%	34.2%	34.3%	37.6%	35.3%	**34.1%**
Improvement		18.4%	26.7%	20.5%	26.9%	26.7%	19.7%	24.6%	27.1%
Average R_{Norm}	1	0.827	0.749	0.806	0.746	0.748	0.816	0.769	**0.743**
Improving ratio		**100%**	**100%**	**100%**	**100%**	**100%**	87.5%	**100%**	93.8%
Cluster 3 (building number: 16)									
Update method	Static	AL	IL	EL	EWC	SI	LFL	MR	GEM
$CV_{average}$	45.7%	47.0%	40.2%	46.6%	40.4%	**40.1%**	44.9%	42.8%	40.5%
Improvement		-2.8%	12.0%	-2.0%	11.6%	12.3%	1.8%	6.3%	11.4%
Average R_{Norm}	1	1.030	0.863	1.010	0.869	**0.861**	0.971	0.926	0.870
Improving ratio		37.5%	**100%**	37.5%	**100%**	**100%**	68.8%	93.8%	93.8%
Cluster 4 (building number: 2)									
Update method	Static	AL	IL	EL	EWC	SI	LFL	MR	GEM
$CV_{average}$	63.8%	42.6%	**22.2%**	42.0%	27.9%	**22.2%**	26.9%	30.6%	37.0%
Improvement		18.4%	26.7%	20.5%	26.9%	26.7%	19.7%	24.6%	27.1%
Average R_{Norm}	1	0.666	**0.332**	0.658	0.419	0.333	0.414	0.471	0.592
Improving ratio		100%	100%	100%	100%	100%	100%	100%	100%

The building load levels of cluster 1 remain stable over the whole period. The load profiles of both clusters 2 and 3 demonstrate distinct annual periodicity. The two buildings in cluster 4's load profiles have risen significantly in the second half of the first year. The static model achieves CV-RMSE of 31.1% for cluster 1, around 46% for clusters 2 and 3, and 63.8% for cluster 4.

Considering the load profile characteristics of each cluster, this progressive increase can be expected. For cluster 1 (with a steady load level), the impact of model update is marginal/minimal. Various model update methods only enhance the static model by 4 to 9%. Meanwhile, the improvement brought by model update is more significant for clusters 2 and 3 (with distinct annual periodicity). For instance, each of the IL, EWC, SI, MR and GEM methods achieves more than 20% improvement for cluster 2. For cluster 3, update methods including IL, EWC, SI and GEM can provide an improvement of around 14 per cent. For cluster 4, model update exhibits the most noticeable positive influence. Particularly, incremental learning achieves the most significant effect among all update methods, which improves the model performance from 63.8% to 22.2%. This corresponds to the min value of R_{Norm} in Table 3. It is not difficult to comprehend this phenomenon by observing the temporal variation in the electricity consumption of the two buildings in cluster 4, which only contains a sudden drift. When facing the tradeoff between stability and plasticity (i.e., the stability-plasticity dilemma), incremental learning completely leans towards plasticity and entirely neglects the catastrophic forgetting problem. However, for the two buildings in cluster 4, forgetting the knowledge learned from previous period is not penalized by the subsequent subsets. In comparison, continual learning methods (and also accumulative learning) cannot fully adapt to the changes of data characteristics as they need to memorize previous knowledge, which does not reward/compensate them in subsequent tasks.

4.4 Discussions

This study comprehensively compares and visualizes the stability-plasticity trade-off of different model updating methods. The incremental learning and continual learning methods are more "plastic" than accumulative learning and ensemble learning. Accumulative learning is typically regarded as the performance upper boundary of continual learning methods in many research. However, due to the evaluation criteria adopted for data streams naturally generated in chronological order (i.e., Equation 2), the performance of accumulative learning in this study is mediocre. The overall results show that higher plasticity leads to better prediction performance for building energy prediction. For some cases (e.g., buildings in cluster 4), forgetting previously learned knowledge is not penalized. The investigation on building-specific concept drift impacts in this study is of essential guiding significance for future research. Besides the building energy prediction model, there are other more models in the building energy field that require continuous updates, e.g., the resistance-capacitance network model, chiller performance model, human comfort prediction model, and fault diagnosis model. For a given task, it is possible to estimate the composition of concept drift based on domain expertise and engineering experience. For instance, re-occurring drift often dominates the concept drift of a process that is significantly affected by climate factors. In these circumstances, continual learning methods can be a wise option. For tasks where gradual drift is the primary factor (e.g., equipment performance degradation), the incremental learning method might be the most appropriate/effective choice.

Continual learning methods are applicable for not only regression tasks, but also classification tasks, e.g., fault detection and diagnosis (FDD). For FDD in the building management field, most studies assumed that all fault types were known in advance (i.e., the training dataset consists of all fault types), but this may not always be the case in practice. That is, the trained FDD models may encounter new fault types that were not present in the training set. This class-incremental challenge, rarely considered or addressed in previous FDD studies, has achieved increasing attention in continual/lifelong learning aspect [44]. Future FDD research in building management will benefit from the appropriate utilization of these continual learning methods.

This study has some limitations and likewise offers prospects for further investigation. The model update strategy also has a significant impact on the performance of adaptive models. This study only tests the most straightforward option, i.e., the periodic update strategy. Other strategies include drift detector-based strategy and performance degradation-based (or error rate) strategy [45,46]. That is, model updates are triggered when concept drift or a predetermined level of model performance degradation is observed/detected. These strategies, which are obviously more targeted, have the potential to further enhance the effectiveness of continual learning methods and merit further in-depth investigations in future research. Moreover, this study only tested on a MLP neural network with two hidden layers. More complex or advanced neural networks as well as other machine learning models can be tested in future research.

5. Conclusion

During model deployment, continuous model update is essential to adapt to the changes of building characteristics and alleviate performance decline. This research conducts a comprehensive comparison between eight update methods for building energy prediction, including three conventional methods and five continual learning methods. The case study on one hundred buildings from an open-source dataset evaluates these methods in terms of prediction accuracy and computation time. The results reveal that continual learning methods (especially Gradient episodic memory and Elastic weight consolidation) can significantly improve the prediction accuracy and reduce the model training time compared with the widely-used accumulative learning method. The impact of building-specific concept drift differences on the update methods is also investigated based on clustering analysis. The research insights can guide and facilitate the application and deployment of advanced machine learning algorithms in the building management sector. Future research can be undertaken to investigate the effectiveness of model update methods with different update trigger mechanisms and with different machine learning models.

Acknowledgement

The authors gratefully acknowledge the support of this research by National Key Research and Development Program of China (2021YFE0107400), the Research Grants Council (C5018-20GF) and Innovation and Technology Fund (ITP/002/22LP) of the Hong Kong SAR, China.

References

1. Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. Science, 349(6245), 255-260.
2. Hong, T., Wang, Z., Luo, X., & Zhang, W. (2020). State-of-the-art on research and applications of machine learning in the building life cycle. Energy and Buildings, 212, 109831.
3. Li, A., Xiao, F., Zhang, C., & Fan, C. (2021). Attention-based interpretable neural network for building cooling load prediction. Applied Energy, 299, 117238.
4. Zhao, Y., Li, T., Fan, C., Lu, J., Zhang, X., Zhang, C., & Chen, S. (2019). A proactive fault detection and diagnosis method for variable-air-volume terminals in building air conditioning systems. Energy and Buildings, 183, 527-537.
5. Seyedzadeh, S., Rahimian, F. P., Oliver, S., Rodriguez, S., & Glesk, I. (2020). Machine learning modelling for predicting non-domestic buildings energy performance: A model to support deep energy retrofit decision-making. Applied Energy, 279, 115908.
6. Fan, C., Yan, D., Xiao, F., Li, A., An, J., & Kang, X. (2021, February). Advanced data analytics for enhancing building performances: From data-driven to big data-driven approaches. In Building Simulation (Vol. 14, No. 1, pp. 3-24). Tsinghua University Press.
7. Zhang, L., Wen, J., Li, Y., Chen, J., Ye, Y., Fu, Y., & Livingood, W. (2021). A review of machine learning in building load prediction. Applied Energy, 285, 116452.

8. Liang, F., Hatcher, W. G., Xu, G., Nguyen, J., Liao, W., & Yu, W. (2019, July). Towards online deep learning-based energy forecasting. In 2019 28th International Conference on Computer Communication and Networks (ICCCN) (pp. 1-9). IEEE.
9. Deng, Y., Fan, J., Jiang, H., He, F., Wang, D., Li, A., & Xiao, F. (2022, June). Behavior testing of load forecasting models using BuildChecks. In Proceedings of the Thirteenth ACM International Conference on Future Energy Systems (pp. 76-80).
10. Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., & Zhang, G. (2018). Learning under concept drift: A review. IEEE Transactions on Knowledge and Data Engineering, 31(12), 2346-2363.
11. Fekri, M. N., Patel, H., Grolinger, K., & Sharma, V. (2021). Deep learning for load forecasting with smart meter data: Online Adaptive Recurrent Neural Network. Applied Energy, 282, 116177.
12. Yang, J., Rivard, H., & Zmeureanu, R. (2005). On-line building energy prediction using adaptive artificial neural networks. Energy and buildings, 37(12), 1250-1259.
13. Khan, I. A., Akber, A., & Xu, Y. (2019, May). Sliding window regression based short-term load forecasting of a multi-area power system. In 2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE) (pp. 1-5). IEEE.
14. Alberg, D., & Last, M. (2018). Short-term load forecasting in smart meters with sliding window-based ARIMA algorithms. Vietnam Journal of Computer Science, 5(3), 241-249.
15. Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., ... & Hadsell, R. (2017). Overcoming catastrophic forgetting in neural networks. Proceedings of the national academy of sciences, 114(13), 3521-3526.
16. Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., & Wermter, S. (2019). Continual lifelong learning with neural networks: A review. Neural Networks, 113, 54-71.
17. Qi, X., & Liu, C. (2018, October). Enabling deep learning on iot edge: Approaches and evaluation. In 2018 IEEE/ACM Symposium on Edge Computing (SEC) (pp. 367-372). IEEE.
18. De Lange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., ... & Tuytelaars, T. (2021). A continual learning survey: Defying forgetting in classification tasks. IEEE transactions on pattern analysis and machine intelligence, 44(7), 3366-3385.
19. Awasthi, A., & Sarawagi, S. (2019, January). Continual learning with neural networks: A review. In Proceedings of the ACM India Joint International Conference on Data Science and Management of Data (pp. 362-365).

20. Zhou, Y., Tian, X., Zhang, C., Zhao, Y., & Li, T. (2022). Elastic weight consolidation-based adaptive neural networks for dynamic building energy load prediction modeling. Energy and Buildings, 265, 112098.
21. Lee, C., Kim, S. H., & Youn, C. H. (2020, November). An Accelerated Continual Learning with Demand Prediction based Scheduling in Edge-Cloud Computing. In 2020 International Conference on Data Mining Workshops (ICDMW) (pp. 717-722). IEEE.
22. Krawczyk, B., Minku, L. L., Gama, J., Stefanowski, J., & Woźniak, M. (2017). Ensemble learning for data stream analysis: A survey. Information Fusion, 37, 132-156.
23. Sagi, O., & Rokach, L. (2018). Ensemble learning: A survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8(4), e1249.
24. Kolter, J. Z., & Maloof, M. A. (2005, August). Using additive expert ensembles to cope with concept drift. In Proceedings of the 22nd international conference on Machine learning (pp. 449-456).
25. Zenke, F., Poole, B., & Ganguli, S. (2017, July). Continual learning through synaptic intelligence. In International Conference on Machine Learning (pp. 3987-3995). PMLR.
26. Li, Z., & Hoiem, D. (2017). Learning without forgetting. IEEE transactions on pattern analysis and machine intelligence, 40(12), 2935-2947.
27. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.
28. Wang, L., Zhang, X., Yang, K., Yu, L., Li, C., Hong, L., ... & Zhu, J. (2022). Memory Replay with Data Compression for Continual Learning. arXiv preprint arXiv:2202.06592.
29. Hou, S., Pan, X., Loy, C. C., Wang, Z., & Lin, D. (2019). Learning a unified classifier incrementally via rebalancing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 831-839).
30. Rusu, A. A., Rabinowitz, N. C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., ... & Hadsell, R. (2016). Progressive neural networks. arXiv preprint arXiv:1606.04671.
31. Mallya, A., & Lazebnik, S. (2018). Packnet: Adding multiple tasks to a single network by iterative pruning. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (pp. 7765-7773).

32. Hung, C. Y., Tu, C. H., Wu, C. E., Chen, C. H., Chan, Y. M., & Chen, C. S. (2019). Compacting, picking and growing for unforgetting continual learning. Advances in Neural Information Processing Systems, 32.

33. Hung, S. C., Lee, J. H., Wan, T. S., Chen, C. H., Chan, Y. M., & Chen, C. S. (2019, June). Increasingly packing multiple facial-informatics modules in a unified deep-learning model via lifelong learning. In Proceedings of the 2019 on International Conference on Multimedia Retrieval (pp. 339-343).

34. Miller, C., Kathirgamanathan, A., Picchetti, B., Arjunan, P., Park, J. Y., Nagy, Z., ... & Meggers, F. (2020). The building data genome project 2, energy meter data from the ASHRAE great energy predictor III competition. Scientific data, 7(1), 1-13.

35. Nair, V., & Hinton, G. E. (2010, January). Rectified linear units improve restricted boltzmann machines. In Icml.

36. Jung, H., Ju, J., Jung, M., & Kim, J. (2016). Less-forgetting learning in deep neural networks. arXiv preprint arXiv:1607.00122.

37. Rolnick, D., Ahuja, A., Schwarz, J., Lillicrap, T., & Wayne, G. (2019). Experience replay for continual learning. Advances in Neural Information Processing Systems, 32.

38. Lopez-Paz, D., & Ranzato, M. A. (2017). Gradient episodic memory for continual learning. Advances in neural information processing systems, 30.

39. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems, 32.

40. Lomonaco, V., Pellegrini, L., Cossu, A., Carta, A., Graffieti, G., Hayes, T. L., ... & Maltoni, D. (2021). Avalanche: an end-to-end library for continual learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 3600-3610).

41. Kemker, R., McClure, M., Abitino, A., Hayes, T., & Kanan, C. (2018, April). Measuring catastrophic forgetting in neural networks. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 32, No. 1).

42. Lloyd, S. (1982). Least squares quantization in PCM. IEEE transactions on information theory, 28(2), 129-137.

43. Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of computational and applied mathematics, 20, 53-65.

44. Masana, M., Liu, X., Twardowski, B., Menta, M., Bagdanov, A. D., & van de Weijer, J. (2020). Class-incremental learning: survey and performance evaluation on image classification. arXiv preprint arXiv:2010.15277.

45. Bifet, A., & Gavalda, R. (2007, April). Learning from time-changing data with adaptive windowing. In Proceedings of the 2007 SIAM international conference on data mining (pp. 443-448). Society for Industrial and Applied Mathematics.

46. Gama, J., Medas, P., Castillo, G., & Rodrigues, P. (2004, September). Learning with drift detection. In Brazilian symposium on artificial intelligence (pp. 286-295). Springer, Berlin, Heidelberg.

Appendix

Table 6. Grid-search settings for determining hyperparameter (bold values were adopted in this study)

Hyperparameters	Grid-search values
The activation function in hidden layers	**ReLU**, Sigmoid, Tanh
The number of hidden layer	1, **2**
The neuron number in each hidden layer	10, 15, **20**
GEM memory strength: offset to add to the projection direction in order to favour backward transfer (gamma in original paper)	0.5, 1, 2, 5, **10**, 15, 20, 30, 40, 50, 100
GEM number of patterns per experience in the memory	5, 10, 15, 20, **30**, 40, 50, 100, 200
LFL lambda: Euclidean loss hyper parameter	0.5, 0.7, 1, **1.3**, 1.5, 2, 5, 10, 15, 20, 30
EWC lambda: hyperparameter to weigh the penalty inside the total loss	0.5, 1, 2, 5, 10, 15, **20**, 30, 40, 50, 100, 150, 200, 300
SI lambda	0.5, 1, 2, 5, 10, 15, 20, **30**, 40, 50, 100, 150, 200, 300
MR replay buffer size	30, **50**