Closed-loop Aggregated Baseline Load Estimation using Contextual Bandit with Policy Gradient

Yufan Zhang, Qiuwei Wu, *Senior Member, IEEE*, Qian Ai, *Senior Member, IEEE*, and João P. S. Catalão, *Senior Member, IEEE*

Abstract—**Demand response (DR) is an important technique to explore the demand-side flexibility. The wide deployment of smart meters makes it possible to quantify the baseline load. As an intermediate agent, demand response aggregator needs to obtain the aggregated baseline load (ABL) for the DR event. Previous studies about the household level estimation focus on the estimation method. However, for ABL estimation, customer division is an important issue. A major limitation is the mismatch between the objectives of segmentation and estimation. Therefore, this paper proposes a new closed-loop method for estimating the ABL, which utilizes the contextual bandit with policy gradient to link the segmentation with the estimation. As such, the ABL estimation accuracy can guide the segmentation to divide the customers. The segmentation and estimation optimize collaboratively to improve the ABL estimation accuracy. An ensemble method for combining network's weights during the training process is proposed. Moreover, a pre- and post-event adjustment method is developed to further improve the estimation accuracy. Comprehensive comparisons demonstrate the proposed method can achieve the best estimation performance with regard to the MAPE and RMSE. It improves the estimation accuracy by 7% in terms of MAPE, and 11% in terms of RMSE.**

***Index Terms*—Aggregated baseline load; Contextual bandit; Demand response; Adjustment method; Ensemble method**

I. Introduction

Demand response (DR) aims to modify the consumption pattern by a price signal or financial incentives. It can not only alleviate the utilities' pressure for reinforcing infrastructure to meet the demand increase, but also enable customers to pay lower bills [1]. In general, the residential sector has promising DR potential. However, it is difficult for a single customer to participate in the DR program by itself. Firstly, due to small scale, a single customer has little competitiveness in the retail market. Secondly, utilities have difficulty in managing such a large amount of DR participants directly [2]. Acting as an intermediate agent between the system and customers, the demand response aggregator (DRA) can help solve these problems. When supply-demand balancing or constraint management are needed, a DR program may be activated and the system operator may ask the DRA to respond to the price signal [3] or incentives [4]. How to quantitatively obtain the DRA's response capacity, referred to the overall response of the DRA towards the price or incentive in the DR event, is a significant task. To meet this end, the baseline load estimation is needed [5], such that the DR capacity can be obtained by the difference between the actual load consumption and the estimated baseline load. Due to the difficulty of accurate estimation, [6] expressed concerns about the DR program based on the baseline load. If the price responsiveness cannot be accurately estimated, the system operator's benefit may be jeopardized or DRA's motivation to participate in the DR program may be weakened. Hence, the study of accurate baseline load estimation at the DRA level is of great importance.

A lot of research has been conducted for the baseline load estimation in recent years. The methods for the baseline load estimation can be classified into four categories, i.e., similar day-based [7], control group-based [8], exponential moving average [9], and regression-based methods [10-12]. The similar day-based method uses the average of historical non-event days' loads for estimation. HighXofY, MidXofY and LowXofY are three typical similar day-based methods [7]. The control group-based method utilizes the synchronous load of non-participating customers who have similar consumption patterns with the target DR participating customer. In [8], k-means clustering was used to explore similarity between customers. And the inner-class-average based on it was proved to be stable and could consistently produce good results. The exponential moving average-based method is a linear model,

This work was supported in part by the National Natural Science Foundation of China under Grant U1866206

Yufan Zhang and Qian Ai are with the Key Laboratory of Control of Power Transmission and Conversion, Ministry of Education, Department of Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: zhangyufan@sjtu.edu.cn; aiqian@sjtu.edu.cn).

Qiuwei Wu is with Tsinghua-Berkeley Shenzhen Institute, Tsinghua Shenzhen International Graduate School, Tsinghua University, Shenzhen 518055, China (e-mail: quiwudtu@gmail.com).

João P. S. Catalão is with Faculty of Engineering of University of Porto and INESC TEC, 4200-465 Porto, Portugal (e-mail: catalao@fe.up.pt)

Corresponding author: Qiuwei Wu (e-mail: quiwudtu@gmail.com).

and is adopted by ISO New England (ISO-NE). The regression-based method aims to fit a model to represent the relationship between input features and baseline load [10]-[12]. Ref. [10] used artificial neural network for baseline load estimation and proved its superiority to the linear regression model-based estimation method. Utilizing the temperature data two hours before the DR event, [11] constructed a support vector regression (SVR) model for office building's baseline load estimation. The SVR model was proved to be accurate and stable. Using a quantile regression forests model, [12] utilized the historical data, weather information, and synchronous measurement from control group as input features, and the CBL estimations in quantile form were obtained. Moreover, to further improve the performance, the adjustment methods were reported in [9, 13]. Weather adjustment and morning adjustment are two kinds of widely used methods, which aim to handle the variations in weather or usage pattern, respectively. The aforementioned studies [7]-[13] are about the customer baseline load (CBL) estimation, which is the estimation at the household level.

To estimate the response capacity of DRA, the aggregated baseline load (ABL) estimation is needed, which calculates the load consumption of the aggregator if the DR event doesn't happen. It is important for power system applications such as flexibility modelling and tariff design [14]. For the aggregated flexibility modelling, the ABL can be used to specify the limits of the flexibility, which is the minimal or maximal flexibility levels of the aggregator. For tariff design, [14] leveraged the ABL as state information for deep reinforcement learning. Then, the imposed tariff on DRA is designed based on the given state.

To model the response behavior of smart households, [15] used the home energy management system to perform optimal scheduling, and the aggregated DR capacity was obtained. However, [15] obtained the DR capacity by the analytical optimization model. Since the analytical optimization model may involve simplification, data driven solutions to estimate the ABL are needed. For data driven methods, it is true that the load at the household level has larger variability and volatility than the aggregated load, and the methods for CBL estimation can be applied to the ABL estimation. However, different from the CBL estimation, apart from the estimation method, the ABL estimation highly relies on the segmentation of customers. Specifically, for ABL estimation, there are three typical ways of customer division. In the first category, each customer forms a cluster individually, and then the sum of CBL estimations forms the result of ABL estimation. In the second category, all customers form one big cluster. ABL is obtained by applying estimation method on the aggregated load. In the third category, customers are first divided into several groups. Then, ABL is obtained by aggregating the "middle-level" estimations produced by the clusters. The first two categories can be regarded as the special cases of the third category, and usually have worse performance. So, how to properly divide customers to form groups is an important issue facing ABL estimation, which makes it distinct from the CBL estimation. Ref. [16] proposed a Gaussian Mixture Model (GMM)-based method for ABL estimation. The customers were grouped into several clusters by GMM, and SVR was leveraged to estimate the ABL. The advantage of the model was demonstrated by comparing it with spectral clustering-based ABL estimation. Ref. [17] proposed a clustering-based aggregated forecasting method and showed that the number of clusters and the size of customer base affect the forecast accuracy. To further improve the performance of the ABL estimation, ensemble method is a promising approach. Ensemble estimation combines the results of different methods to take advantage of the strength of them and is expected to produce better results than a single method [18]-[19]. Ref. [18] utilized the homogeneous ensemble method and fine-grained sub-profiles to further improve the aggregated load forecasting accuracy. By varying the number of clusters for hierarchical clustering, clustering-based forecasting was implemented for each dataset division. And the problem for combing the deterministic forecast results was formulated as a linear programming (LP) problem which minimized the mean absolute percent error (MAPE).

However, the discussed clustering-based approach [16]-[17] and the clustering-based homogeneous ensemble method [18] treat the clustering and estimation as two separate procedures, which results in the mismatch of the objectives between these two parts. Therefore, the existing method leaves the following problem unsolved: the customer segmentation algorithm groups the customers by the criterion of minimizing the consumption patterns' dissimilarity rather than improving the estimation accuracy. Hence, how to link the estimation with the customer segmentation and construct the feedback remains an interesting question.

Among various clustering algorithms, adaptive clustering uses the external feedback to improve the clustering quality. During the iteration process, using the clustering performance as feedback, [20] proposed to select a weight-changing action to adaptively revise the distance function. The new distance function was applied for the clustering in the next iteration. Similarly, [21] leveraged the idea of adaptive clustering and proposed a distributed clustering algorithm. The number of clusters was determined adaptively during the learning process. The main purpose of adaptive clustering is to improve the clustering performance, such that customers with the similar consumption patterns are grouped into the same cluster. In contrast, for ABL estimation, the aim of the clustering is to group customers appropriately to improve the estimation accuracy. Also, the adaptive clustering relies heavily on the distance measurement. Different distance measurement can lead to the different results and how to choose proper distance measurement itself is a complex issue.

The bandit problem and reinforcement learning (RL) method are also famous for their applicability in solving problems in a closed-loop. And with the recent development of deep neural network (DNN), deep RL (DRL) is gaining emphasis. The bandit problem determines the actions without using any information about the state of the environment [22]. Ref. [23] proposed a risk-averse multi-armed bandit learning approach to provide the reliable secondary frequency regulation, such that customers with high estimated participation probability were

chosen to participate in the regulation. In contrast, the DRL algorithm leverages the state of the environment to choose the action. It involves an agent acting based on the current state observation, and the DRL algorithm learns the optimal control policy that maximizes expected cumulative reward [24]-[25]. Ref. [26] proposed on-line optimization of schedules for building energy management systems by using deep policy gradient (DPG) algorithm and demonstrated its superiority. Contextual bandit is an interpolation between DRL and bandit algorithms. Without the evolutional state, contextual bandit can be regarded as one-step RL, and the input features (context/state) of contextual bandit only affects the reward without affecting the next state [27]-[28], such that it can be regarded as a special case of RL. Therefore, the methods in RL area can also be applied to solve the contextual bandit problem.

Leveraging contextual bandit method, this work proposes a new ABL estimation method, which integrates the customer segmentation and estimation together. In this paper, the contextual bandit method with policy gradient is used. The customer segmentation problem is modelled by the agent which learns the stochastic policy by DNN that maps the state to a distribution of actions. The estimation problem (a supervised learning problem) is the environment. Specifically, the agent outputs the action which determines the clusters that customers are assigned in. And such action can affect the environment which makes the ABL estimation. The ABL estimation performance is then used as the reward to guide the decision-making process of the agent. As such, a closed loop is formed. Since the representative consumption patterns of customers are used as the state which has no revolution and isn't affected by the action, our model is a contextual bandit problem. Also, compared with the adaptive clustering method, the customer segmentation is fulfilled by the forward propagation of DNN, which is more similar to the forward propagation of a multi-classification task. There are two main advantages: First, since the customers are not grouped according to the distance measurement, the problem of selecting a particular distance metric is avoided. Second, the process only involves the matrix operations, which is more computational efficient than the iterative process that most clustering algorithm involves.

To summarize, the contributions of this paper are as follows:

1) Propose a new closed-loop contextual bandit-based method for the ABL estimation. Under this framework, the feedback mechanism is constructed by the reward and action. Therefore, the agent is able to gain the knowledge of the environment and properly divides customers to improve the ABL estimation accuracy. And the segmentation and estimation are consistently optimized toward the common goal.

2) Propose a weight selection method for the DNN of contextual bandit's agent. Recent practice of determining DNN's weight is choosing the one that has the best performance on the validation set. However, due to the mismatch of data structure on the validation and test sets, the DNN's weight performing well on the validation set doesn't guarantee to have good estimation results on the test set. Therefore, instead of relying on a particular DNN's weight, during the training process, multiple DNN's weights which have good estimation performance on the validation set are saved. Then, an ensemble method is applied to find an optimal combination way for those DNNs with different weights. As such, the estimation results are the weighted sum of estimations produced by multiple models, which is more robust to the unseen data than simply relying on one model.

3) Propose a pre- and post-event adjustment method for the ABL estimation. Since the estimation error is caused by the uncertainty part of loads, through the adjustment, the error caused by the similar load variation can be ameliorated. Therefore, with the pre- and post-event adjustment method, the accuracy of the ABL estimation can be further improved.

The remainder of this paper is organized as follows. Section II introduces the ABL estimation problem and illustrates the feature selection process. Section III proposes the integrated segmentation and estimation framework. Details of comparison methods are in Section IV. Results are discussed and evaluated in Section V, followed by the conclusions.

II. Problem Statement and Feature Selection

This work is focused on estimating the ABL. It refers to aggregated electricity that is consumed by a group of ToU consumers (customers who participate in the DR event) if there is no DR event. Fig. 1 illustrates the idea of the ABL. In either high-price (blue area) or low-price (pink area) occasions, the red curve and blue curve represent the ABL and actual consumption, respectively. Note that the ABL estimation is a posterior event estimation approach [12, 29]. Therefore, unlike the load forecasting, the load measurement throughout the day can be obtained.

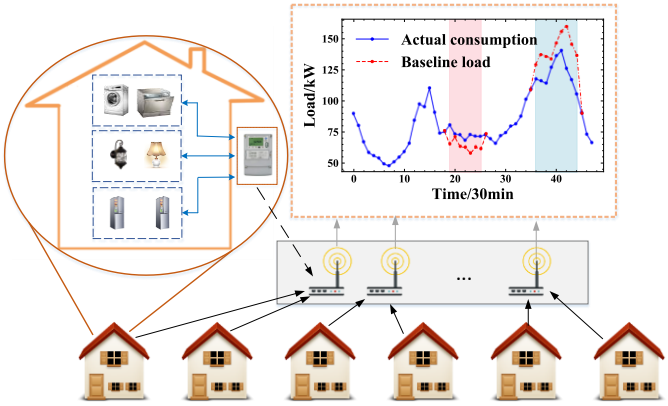

Fig. 1. Illustration of ABL and actual load.

For the feature selection procedure, two kinds of day types are defined. Let Ω^E and Ω^B denote the sets of DR days and non-DR days, respectively. DR days refer to the days when an DR event happens, and others are non-DR days. Also, let Ω^{Week} denote the set of weekdays (Monday to Friday), and $\Omega^{Weekend}$ denote the set of weekends (Saturday and Sunday). In this paper, $\left|\Omega^E\right|+\left|\Omega^B\right|=\left|\Omega^{Week}\right|+\left|\Omega^{Weekend}\right|=365$, where $|\ |$ is the cardinality of the set.

For a given ToU customer i, if a day $h\in\Omega^E\cap\Omega^{Week}$, the daily load data $\boldsymbol{d}_{i,h}$ can be divided into non-event data

$\boldsymbol{d}_{i,h}^{base} \in \mathrm{R}^{1\times\left|c_h^b\right|}$ and event data $\boldsymbol{d}_{i,h}^{event} \in \mathrm{R}^{1\times\left|c_h^e\right|}$, where c_h^b, c_h^e denote the sets of non-event and event time slots of day h, respectively. And $\left|c_h^b\right|+\left|c_h^e\right|=T$ where T is the total number of time slots in a day. Let $D_i\left(Y\right)_h \in \Omega^B \cap \Omega^{Week}$ be a set of Y most recent non-DR days prior to the day h and they are also weekdays. In this paper, Y is set as 7 [7]. Similarly, customer i's Y load profiles can also be partitioned into event and non-event parts: $\boldsymbol{D}_{i,h}^{event} \in \mathrm{R}^{Y\times\left|c_h^e\right|}, \boldsymbol{D}_{i,h}^{base} \in \mathrm{R}^{Y\times\left|c_h^b\right|}$. $\left\{\boldsymbol{D}_{i,h,t}^{base}, d_{i,h,t}^{base}\right\}_{h\in\Omega^E\cap\Omega^{Week};t\in c_h^b}$ and $\left\{\boldsymbol{D}_{i,h,t}^{event}, d_{i,h,t}^{event}\right\}_{h\in\Omega^E\cap\Omega^{Week};t\in c_h^e}$ are feature and label pairs.

Likewise, for day $h\in\Omega^E\cap\Omega^{Weekend}$, through a similar way as described above, the feature and label pairs can be obtained: $\left\{\tilde{\boldsymbol{D}}_{i,h,t}^{base}, \tilde{d}_{i,h,t}^{base}\right\}_{h\in\Omega^E\cap\Omega^{Weekend};t\in c_h^b}, \left\{\tilde{\boldsymbol{D}}_{i,h,t}^{event}, \tilde{d}_{i,h,t}^{event}\right\}_{h\in\Omega^E\cap\Omega^{Weekend};t\in c_h^e}$

To sum up, the training feature $\boldsymbol{X}_i^{Train}$ used to fit the estimation model can be expressed as,

$$\boldsymbol{X}_i^{Train}=\left[\left\{\boldsymbol{D}_{i,h,t}^{base}\right\}_{h\in\Omega^E\cap\Omega^{Week};t\in c_h^b};\left\{\tilde{\boldsymbol{D}}_{i,h,t}^{base}\right\}_{h\in\Omega^E\cap\Omega^{Weekend};t\in c_h^b}\right] \quad (1)$$

The response variable of the training feature $\boldsymbol{X}_i^{Train}$ is the stack of target baseline load during the non-event hours:

$$\boldsymbol{Y}_i^{Train}=\left[\left\{d_{i,h,t}^{base}\right\}_{h\in\Omega^E\cap\Omega^{Week};t\in c_h^b};\left\{\tilde{d}_{i,h,t}^{base}\right\}_{h\in\Omega^E\cap\Omega^{Weekend};t\in c_h^b}\right] \quad (2)$$

At the test stage, the input test feature $\boldsymbol{X}_i^{Test}$ is:

$$\boldsymbol{X}_i^{Test}=\left[\left\{\boldsymbol{D}_{i,h,t}^{event}\right\}_{h\in\Omega^E\cap\Omega^{Week};t\in c_h^e};\left\{\tilde{\boldsymbol{D}}_{i,h,t}^{event}\right\}_{h\in\Omega^E\cap\Omega^{Weekend};t\in c_h^e}\right] \quad (3)$$

And the response variable of $\boldsymbol{X}_i^{Test}$ is:

$$\boldsymbol{Y}_i^{Test}=\left[\left\{d_{i,h,t}^{event}\right\}_{h\in\Omega^E\cap\Omega^{Week};t\in c_h^e};\left\{\tilde{d}_{i,h,t}^{event}\right\}_{h\in\Omega^E\cap\Omega^{Weekend};t\in c_h^e}\right] \quad (4)$$

It is the target baseline load of the ToU customer i.

III. Integrated Segmentation and Estimation Method

Instead of treating customer segmentation and estimation as two separate procedures, in this section, a contextual bandit-based method is proposed to integrate those two parts. As such, a closed-loop is formed, and the customer segmentation and estimation problems are optimized toward a common goal. The proposed procedure can be divided into the training, ensemble, and test stages, which is summarized in Algorithm 1. The corresponding flow chart is in Fig. 2.

A. Data Splitting

The estimation part is implemented based on the obtained feature and label pairs from Section II, where $\boldsymbol{X}_i^{Train}=[\boldsymbol{x}_{i,1}^T;...;\boldsymbol{x}_{i,N_{tr}}^T]\in\mathrm{R}^{N_{tr}\times Y}$, $\boldsymbol{X}_i^{Test}=[\boldsymbol{x}_{i,1}^T;...;\boldsymbol{x}_{i,N_{te}}^T]\in\mathrm{R}^{N_{te}\times Y}$. Here, N_{tr}, N_{te} are the number of samples in the training and test sets respectively. As shown in Fig. 3, the dataset consists of three parts, namely the sub-training, validation, and test sets. For each customer, the feature and label pairs of training set are further divided into a sub-training set $\left\{\boldsymbol{X}_i^{sub-Tr}\in\mathrm{R}^{N_{sub-tr}\times Y}, \boldsymbol{Y}_i^{sub-Tr}\in\mathrm{R}^{N_{sub-tr}\times 1}\right\}$ and a validation set $\left\{\boldsymbol{X}_i^{Vali}\in\mathrm{R}^{N_{va}\times Y}, \boldsymbol{Y}_i^{Vali}\in\mathrm{R}^{N_{va}\times 1}\right\}$, where N_{sub-tr}, N_{va} are the number of samples in the sub-training and validation sets respectively.

$$N_{sub-tr}\approx 80\%\times N_{tr}, N_{va}=N_{tr}-N_{sub-tr} \quad (5)$$

The sub-training set is used to train the estimation models. The validation set is used for guiding the agent and combing estimation results. The test set is used to test the effectiveness of the proposed method for the ABL estimation. Therefore, both the sub-training and validation sets are used in the training process, while the test set is not.

Algorithm 1: Contextual bandit-based closed-loop ABL estimation method

Initialize DNN with random weights θ

Initialize the number of clusters K, the number of saved DNN's weights N, the learning rate $\eta=1e-3$, exploration rate $\varepsilon_{\max}=1, \varepsilon_{\min}=0.01, \varepsilon_{\text{decay}}=0.995$.

Training Stage

for epoch = 1 to arbitrary number **do**

With probability ε select random actions $\left\{a_m^e\right\}_{m=1}^M$

Otherwise select $\left\{a_m^e=\arg\max\, p_\theta\left(a_m^e\mid s_m\right)\right\}_{m=1}^M$

Execute $\left\{a_m^e\right\}_{m=1}^M$ in the environment.

for $k=1:K$ **do**

Train estimation model $f_k(\cdot)$ based on the feature and label pair $\left\{\sum_{i\in C_k}\boldsymbol{x}_{i,t}^{sub-Tr}, \sum_{i\in C_k}y_{i,t}^{sub-Tr}\right\}$.

Estimate the k_{th} cluster's baseline load for the validation set: $\hat{y}_{k,t}^{Vali,agg}=f_k\left(\sum_{i\in C_k}\boldsymbol{x}_{i,t}^{Vali}\right)$.

end

Calculate the ABL on the validation set $\hat{y}_t^{Vali,agg}=\sum_{k=1}^K\hat{y}_{k,t}^{Vali,agg}$, and return the reward to the agent according to (8).

Calculate $L(\theta), \nabla_\theta L(\theta)$ according to (9) and (11), and update the parameters of DNN: $\theta\leftarrow\theta+\eta\cdot\nabla_\theta L(\theta)$.

if $\varepsilon>\varepsilon_{\min}$

$\varepsilon\leftarrow\varphi(e,\varepsilon)$

else

$\varepsilon\leftarrow\varepsilon_{\min}$

end

end

Ensemble Stage

According to (14), solve the optimization problem to determine the combining weights for ABL results produced by N division ways.

Test stage

$\forall h\in\Omega^E, t\in c_h^e$, make the adjustment for $\hat{y}_{n,k,t}^{Test,agg}$ according to (15) to obtain $\hat{y}_{n,k,t}^{Test,agg,adj}$. And then calculate the $\hat{y}_t^{Test,agg,adj}$. MAPE and RMSE are used to evaluate its performance.

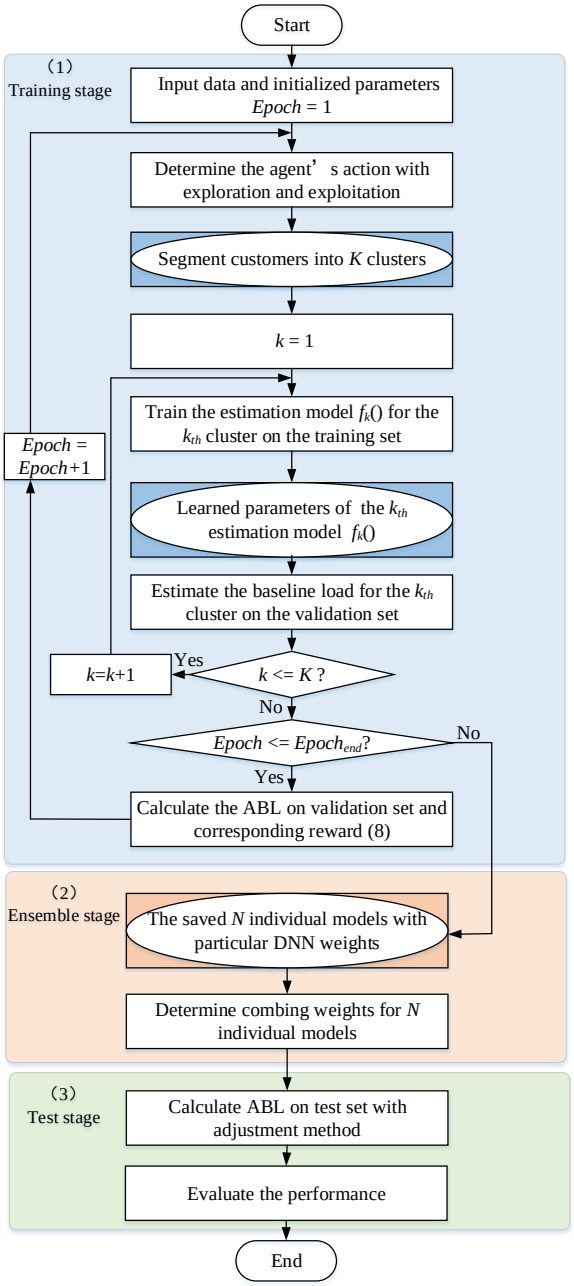

Fig. 2. Flowchart of proposed integrated segmentation and estimation framework.

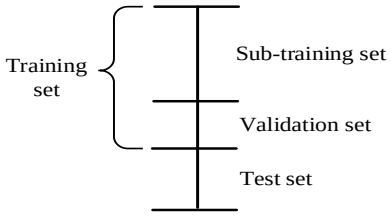

Fig. 3. Illustration of dataset splitting.

B. Training Stage

The training stage is based on the sub-training and validation sets. Take the sub-training set for example, for any customer i whose input feature for the estimation at the time stamp t is $\boldsymbol{x}_{i,t}^{sub-Tr}$, the ABL can be described by,

$$\begin{aligned} \hat{y}_{k,t}^{sub-Tr,agg} &= f_k\left(\sum_{i\in C_k} \boldsymbol{x}_{i,t}^{sub-Tr}\right) \\ \hat{y}_t^{sub-Tr,agg} &= \sum_{k=1}^{K} \hat{y}_{k,t}^{sub-Tr,agg} \end{aligned} \tag{6}$$

where K is the given number of clusters. $\hat{y}_{k,t}^{sub-Tr,agg}$ is the estimated load for the k_{th} cluster and the sum of $\hat{y}_{k,t}^{sub-Tr,agg}$ is the ABL estimation $\hat{y}_t^{sub-Tr,agg}$ at the time stamp t. $f_k(\cdot)$ is the regression function for the k_{th} cluster C_k which is the k_{th} partition of M customers and satisfies the following property:

$$\begin{aligned} &\cup_{k=1}^{K} C_k = \{1,2,...,M\} \\ &C_i \cap C_j = \varnothing, \forall i \neq j \end{aligned} \tag{7}$$

The integrated model in the training stage consists of two dependent sub-problems, namely the estimation and customer partition. For the estimation problem, the goal is to find fitted regression functions $\{f_k(\cdot)\}_{k=1}^{K}$ with the objective of minimizing the difference between the estimation and true values. The aim of the partition is to find the optimal customer portfolio, i.e., the elements within the cluster, to help realize the accurate ABL estimation. To coordinate the two sub-problems, we propose a contextual bandit-based method and the illustration is shown in Fig. 4.

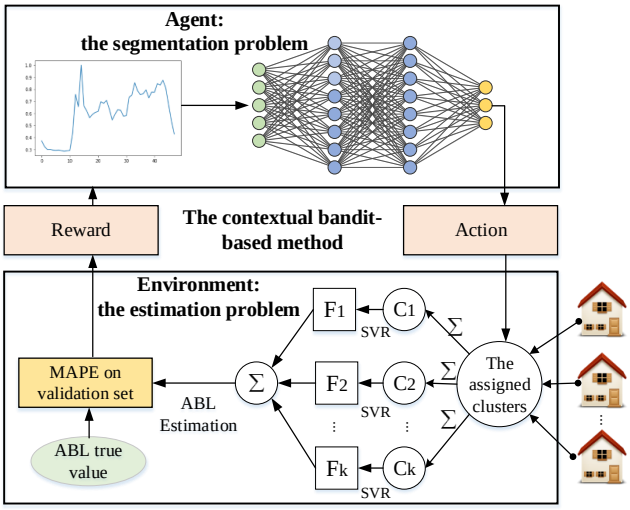

Fig. 4. Overview of proposed contextual bandit-based framework.

Generally, the contextual bandit problem has five fundamental elements: agent, environment, state $\boldsymbol{S}$, action $\boldsymbol{A}$, and reward R. In this paper, the fundamental elements are defined as follows:

1) Agent: the customer segmentation problem.

2) Environment: the estimation problem.

3) State: Customer's representative load pattern (RLP) is used as the state. Specifically, for a given ToU customer i, let $\boldsymbol{X}_i \in \mathrm{R}^{365\times T}$ be the load profiles in the whole year. The yearly average load $\boldsymbol{x}_i^{mean} \in \mathrm{R}^{1\times T}$ is used as the RLP.

4) Action: the assigned cluster of a customer is the agent's

output action.

5) Reward: To encourage the model to have good generalization ability, after fitting the regression models on the sub-training set, the negative value of MAPE on the validation set is used as the reward. In this way, the reward can guide the agent to reduce the MAPE on the unseen data instead of on the training data, and therefore the generalization ability is improved. The reward is expressed as,

$$R = -\frac{1}{N_{va}} \sum_{t=1}^{N_{va}} \frac{\left| \hat{y}_t^{Vali,agg} - y_t^{Vali,agg} \right|}{y_t^{Vali,agg}} \tag{8}$$

where $\hat{y}_t^{Vali,agg}, y_t^{Vali,agg}$ are the estimation and true values on the validation set.

As shown in Fig. 4, the agent learns a policy π by DNN to maximize the total expected reward:

$$\max L(\theta) = \frac{1}{M} \sum_{m=1}^{M} R^e \cdot p_\theta \left(a_m^e \mid s_m \right) \tag{9}$$

where θ represents the weights of the agent's DNN. R^e is the gained reward at the e_{th} epoch. s_m is the state of the m_{th} customer and a_m^e is the e_{th} epoch's action which is a one-hot vector determining the cluster that the m_{th} customer is assigned in. The determined actions $\left\{ a_m^e \right\}_{m=1}^{M}$ are then passed to the environment and used as the basis for the determination of $\left\{ C_k \right\}_{k=1}^{K}$. To balance the exploration and exploitation, the decaying $\varepsilon-\text{greedy}$ algorithm is used to determine the action. The exploration rate ε decays exponentially from $\varepsilon_{\max}$ to a small constant value $\varepsilon_{\min}$, which is defined as $\varphi(e, \varepsilon)$:

$$\varphi(e, \varepsilon) = \left(\varepsilon_{\text{decay}} \right)^{e-1} \cdot \varepsilon_{\max} \tag{10}$$

Using the decaying $\varepsilon-\text{greedy}$ algorithm, there is $1-\varepsilon$ probability to choose actions $\left\{ a_m^e = \arg\max\, p_\theta \left(a_m^e \mid s_m \right) \right\}_{m=1}^{M}$, and there is ε probability to choose random actions. Therefore, with the decaying exploration rate, the agent can explore more at the beginning and exploit more at the end.

Based on the sub-training set, the estimated ABL is obtained according to (6). Then, the K regression models are fitted and the parameters are learned. The estimated ABL $\hat{y}_t^{Vali,agg}$ on the validation set is obtained by the learned regression model, and the e_{th} epoch's reward R^e is calculated according to (8).

At each epoch, after receiving the reward, the parameters of the agent's DNN are updated by the stochastic gradient ascent with the gradient calculated by (11).

$$\nabla_\theta L(\theta) = \frac{1}{M} \sum_{m=1}^{M} R^e \cdot \nabla \log p_\theta \left(a_m^e \mid s_m \right) \tag{11}$$

Hence, when the training converges, the agent learns how to group the customers and the regression models learn how to make accurate ABL estimation.

C. Ensemble Stage

During the training process, N DNN's weights are saved which result in the first N lowest MAPE scores on the validation set. Consequently, N partition ways are obtained for the M customers. Therefore, according to the estimation performance on the validation set, this stage aims to determine the combining weights of the estimation results obtained from those N division fashions. And the final estimation result is the weighted sum of N estimations:

$$\bar{y}_t^{Vali,agg} = \sum_{n=1}^{N} \omega_n \cdot \hat{y}_{n,t}^{Vali,agg} \tag{12}$$

where $\hat{y}_{n,t}^{Vali,agg}$ is the ABL estimation of the n_{th} partition, and ω_n is the corresponding weight.

The ensemble method proposed in [18] is used to determine the weights. For deterministic estimation, the optimization objective is minimizing the MAPE.

$$\begin{aligned} \boldsymbol{w} &= \arg\min_{\boldsymbol{\omega}} \sum_{t=1}^{N_{va}} \frac{1}{N_{va}} \frac{\left| y_t^{Vali,agg} - \bar{y}_t^{Vali,agg} \right|}{y_t^{Vali,agg}} \\ s.t.\ & \sum_{n=1}^{N} \omega_n = 1, \omega_n \geq 0,\ \bar{y}_t^{Vali,agg} = \sum_{n=1}^{N} \omega_n \cdot \hat{y}_{n,t}^{Vali,agg} \end{aligned} \tag{13}$$

By introducing the auxiliary decision variables $u_t = \max\left\{ \left(\bar{y}_t^{Vali,agg} - y_t^{Vali,agg} \right), \left(y_t^{Vali,agg} - \bar{y}_t^{Vali,agg} \right) \right\}$, (13) is transformed into the following LP problem:

$$\begin{aligned} \boldsymbol{w} &= \arg\min_{\boldsymbol{\omega}} \frac{1}{N_{va}} \sum_{t=1}^{N_{va}} \frac{u_t}{y_t^{Vali,agg}} \\ s.t.\ & \bar{y}_t^{Vali,agg} = \sum_{n=1}^{N} \omega_n \cdot \hat{y}_{n,t}^{Vali,agg}, \sum_{n=1}^{N} w_n = 1, w_n \geq 0 \\ & u_t \geq \bar{y}_t^{Vali,agg} - y_t^{Vali,agg}, u_t \geq y_t^{Vali,agg} - \bar{y}_t^{Vali,agg} \end{aligned} \tag{14}$$

D. Test Stage

In the test stage, MAPE and root mean squared error (RMSE) are chosen as the evaluation metrics to assess the performance of ABL estimation. And pre- and post-event adjustment is proposed to further improve the accuracy.

Pre- and post-event adjustment can handle the daily variation of the consumption pattern. It is based on the ratio of actual load to the estimated load values during pre- and post-event hours in an event day. Concretely, if day $h \in \Omega^E$, $\forall t \in c_h^e$, the adjustment is made according to the following equation:

$$\hat{y}_{n,k,t}^{Test,agg,adj} = \hat{y}_{n,k,t}^{Test,agg} \cdot \frac{\sum_{t' \in c_h^b} y_{n,k,t'}^{sub-Tr/Vali,agg}}{\sum_{t' \in c_h^b} \hat{y}_{n,k,t'}^{sub-Tr/Vali,agg}} \tag{15}$$

where $y_{n,k,t'}^{sub-Tr/Vali,agg}, \hat{y}_{n,k,t'}^{sub-Tr/Vali,agg}$ denote the actual and estimated loads in non-event hours which are in either sub-training or validation sets. $\hat{y}_{n,k,t}^{Test,agg}$ is the estimated ABL of the k_{th} cluster by the n_{th} DNN on the test set. So, the estimated baseline load in the test set after adjustment can be expressed as:

$$\begin{aligned}\hat{y}_{n,t}^{Test,agg,adj} &= \sum_{k=1}^{K} \hat{y}_{n,k,t}^{Test,agg,adj} \\ \hat{y}_{t}^{Test,agg,adj} &= \sum_{n=1}^{N} \omega_n \cdot \hat{y}_{n,t}^{Test,agg,adj}\end{aligned} \tag{16}$$

Here, ω_n is determined in the previous ensemble stage.

IV. Competing Methods

The proposed method is compared with the similar day, exponential moving average, and other regression-based methods. The details of the comparison methods are as follows.

A. Similar Day-based Method

HighXofY, MidXofY, and LowXofY are the three well-established methods based on similar day. For example, HighXofY estimates the baseline load as the average load of *X* highest consumption days within *Y* non-DR days preceding the DR day. The *Y* non-DR days have the same day type as the target DR day. For day h, define the set satisfying the requirement as $High(X,Y)_h$. $\forall t \in c_h^e$, the estimated ABL is:

$$\hat{y}_t = \frac{1}{\left|High(X,Y)_h\right|} \cdot \sum_{d \in High(X,Y)_h} \left(\sum_{i=1}^{M} y_{d,t}^{i} \right) \tag{17}$$

where $y_{d,t}^{i}$ is the load of the i_{th} customer on the day $d \in High(X,Y)_h$ at the time stamp t

MidXofY and LowXofY are implemented in the similar way, except that the *X* middle and lowest consumption days are chosen. In this paper, High4of5, Mid4of5, and Low4of5 are used for comparison.

B. Exponential Moving Average

The exponential moving average is the weighted sum of the historical baseline load. Define $D_h = \{d_1,...,d_k\}$ as the set of all the same day type non-DR days preceding the target DR day h. $\forall t \in c_h^e$, the initial average load for the first τ days is,

$$s_{\tau,t} = \frac{1}{\tau} \sum_{j=1}^{\tau} y_{j,t} \tag{18}$$

where $y_{j,t} = \sum_{i=1}^{M} y_{j,t}^{i}$ and $y_{j,t}^{i}$ is the load of the i_{th} customer on the day $j \in D_h$ at the time stamp t.

The exponential moving average for $\tau < j \le k$ is,

$$s_{j,t} = \lambda \cdot s_{j-1,t} + (1-\lambda) \cdot y_{j,t} \tag{19}$$

where $\lambda \in [0,1]$, and the estimated ABL is:

$$\hat{y}_t = s_{k,t} \tag{20}$$

In this paper, τ=5, λ=0.9 are used [7].

C. Regression-based Method

Here, three kinds of regression-based methods are used for comparison, namely the fully aggregated estimation, clustering-based estimation, and clustering-based ensemble estimation methods. The input feature and the regression model of comparison candidates are the same as that of the proposed method. The only difference is that the proposed method treats the clustering and estimation as an integrated model, while those candidates treat two processes as separate parts. For clustering-based estimation, the customers are grouped into 2-7 clusters by three methods, namely k-means (K), hierarchical clustering (H), and GMM (G). The detailed procedure of the clustering-based estimation can be found in [16]. The clustering-based ensemble method is on the basis of the clustering-based method. Then it combines the multiple ABL estimations produced by a specific clustering algorithm with the varying cluster numbers. The combining procedure proposed in [18] is used.

In this paper, the comparison methods are summarized in Table I. For simplicity, the capital letter "X" (X = K/H/G) is used to denote the three clustering methods in the table. For example, 3-K denotes the k-means algorithm with 3 clusters. K-E is the k-means based ensemble estimation on the basis of F-A, 2-K, 3-K, 4-K, 5-K, 6-K, and 7-K.

TABLE I The Summary of The Regression-based Comparison Candidates (X = K/H/G)

Fully aggregated	Clustering-based estimation						Ensemble
F-A	2-X	3-X	4-X	5-X	6-X	7-X	X-E

V. Case Studies

A. Implementation Details

The smart meter data with 30 minutes resolution from the Low Carbon London trail is used [30]. The customers participating in the trail can be divided into two groups, namely the customers who receive flat tariff (non-ToU group), and the customers who receive ToU tariff (ToU group). The non-ToU user receives a flat price, while the ToU user receives a ToU tariff (a high price of 67.2 pence/kWh, a default price of 11.76 pence/kWh, and a low price of 3.99 pence/kWh). The dataset provides the record of the time of DR events. To quantitatively evaluate the performance of the ABL estimation, 441 customers from non-ToU group in the year 2013 are used to evaluate the algorithm. In this way, the actual demand during the event periods can be regarded as the benchmark baseline, and the evaluation metrics can be calculated. Please note that although the price-based DR is studied here, the method is not restricted to it. Since the historical baseline load at the same hour of the day is used as a feature vector for estimation, for the other DR programs such as incentive-based DR, the feature is also available. Therefore, the proposed method can also be applied to other DR programs.

Since the specific estimation model is not the main concern of this work and many advanced regression models, such as deep learning methods, can be used here as the estimation approach in the proposed algorithm, in the case study, we apply the classic SVR model from Python package "scikit-learn" as the estimation model. Also, to make fair comparison, the estimation model for the regression-based candidates listed in Table I is also the SVR model. Therefore, the performance of ABL estimation is not determined by the usage of a particular regression model.

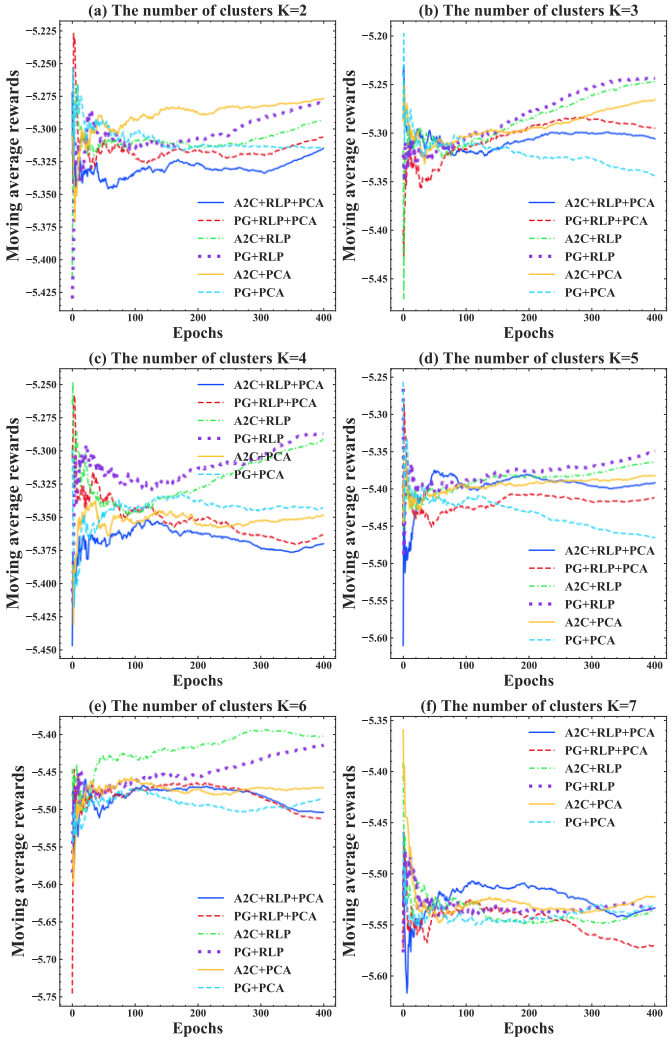

Fig. 5. Comparison of moving average rewards during the training process with different cluster numbers.

To demonstrate the advantage of the proposed method, its dynamic learning process is compared with another five contextual bandit-based ABL estimation methods. The implementation procedure is similar to the proposed method discussed above.

M1) PG+ RLP: The proposed method.

M2) A2C+RLP: Same as the proposed method, RLPs of customers are used as the input state for the DNN. However, instead of using the policy gradient, the advantage actor-critic (A2C) algorithm [31], which is another kind of policy-based method, is adopted for training the agent.

M3) PG+RLP+PCA: the policy gradient is used. The principal component analysis (PCA) is applied to extract features from customers' RLPs. And the extracted features are used as the input state.

M4) A2C+RLP+PCA: M4 is similar to M3 except that A2C is used for contextual bandit.

M5) PG+PCA: Instead of using RLP, the PCA is utilized to extract features directly from the yearly load profile of customer which has 17520 time slots. The extracted features are then used as the state of contextual bandit with policy gradient.

M6) A2C+PCA: M6 is similar to M5 except that A2C is used for contextual bandit.

To make fair comparison and select the proper number of clusters, we experiment with various cluster numbers from 2 to 7, and the moving average rewards of the six methods during the training process are compared. The results are shown in Fig. 5. The larger the average rewards, the better the performance. It can be seen that the proposed method has relatively stable and good performance under experiments with various cluster numbers. And in all experiments, at the end of the training, it achieves the highest average rewards when the number of clusters equals three. Also, its obtained average rewards increase during the training process, which indicates that the agent gradually learns the reasonable customer partition way to improve the ABL estimation accuracy. Therefore, the results prove the superiority of the proposed method. Also, as [32] suggested, the determination of cluster numbers should fit the practical purposes. Since the aim of the clustering is to improve the estimation accuracy, the number of clusters is determined according to the estimation performance. So, the number of clusters and output neurons are chosen as three.

TABLE II MAPE AND RMSE ON THE VALIDATION SET UNDER DIFFERENT VALUES OF N

	$N=10$	$N=20$	$N=30$	$N=40$	$N=50$
MAPE	4.91	4.92	4.9	4.87	4.94
RMSE	7.76	7.78	7.66	7.62	7.68

For the number of saved DNN's weights during the training process, we experiment with several values of it. The MAPE and RMSE scores calculated by the weighted sum of ABL on the validation set under different values of N are shown in Table II. When the value of N equals 40, there are the lowest MAPE and RMSE scores on the validation set. Therefore, we choose N equaling 40 in the following analysis.

Moreover, to demonstrate the superiority of the feature selection method described in Section II, it is compared with the control group-based feature selection procedure. The control group is formed by the other 440 non-ToU customers. The Y non-ToU customers in the control group are selected, whose load patterns in non-event hours are the most similar with that of the target ToU customers. And their synchronized loads are used as the features. The detailed feature selection procedure is described in Appendix *A*. The results on the validation and test sets are shown in Table III. The second and third columns are the results of the proposed method, while the last two columns are that of the comparison method. The only difference between the two methods is the feature selection procedure. It is observed that the comparison candidate has larger MAPE and RMSE values on the validation set. Moreover, its performance on the test set is much worse than the proposed method. This can be because there is smaller correlation between the input features and the target estimation variable on test set for the comparison method. Also, the distributions of the training and test set features are less similar. So the method has worse performance on the test set. The results prove the

superiority of the feature selection method using the historical baseline load.

TABLE III MAPE AND RMSE OF TWO FEATURE SELECTION METHODS ON VALIDATION AND TEST SETS

	Validation set (P)	Test set (P)	Validation set (C)	Test set (C)
MAPE/%	4.87	4.59	6.01	17.54
RMSE/kW	7.62	5.99	7.93	19.59

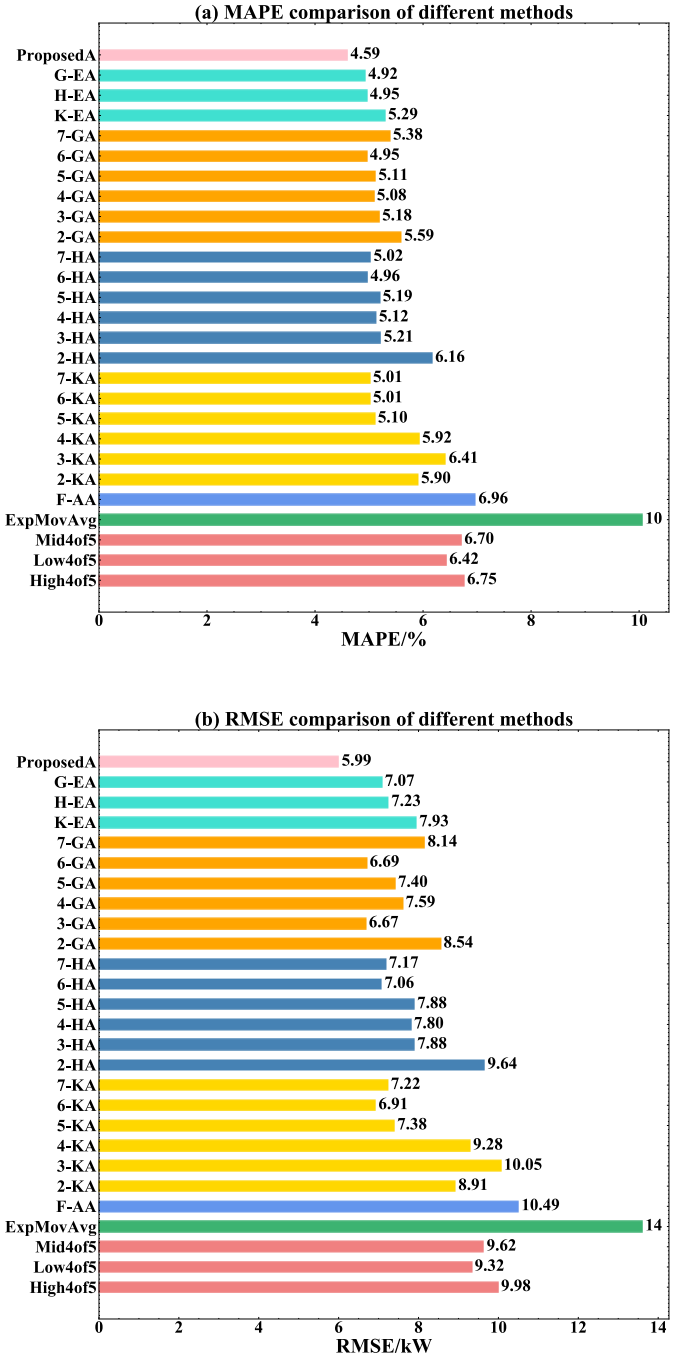

Fig. 6. MAPE and RMSE comparisons of different methods.

Also, for the proposed method, the whole process takes 16min and 18s on the laptop with Intel®CoreTM i5-10210U 1.6 GHz CPU, and 8.00 GM RAM, which demonstrates the computational efficiency of it. The DNN's parameters of the proposed method are summarized in Table IV.

TABLE IV SUMMARY OF THE DNN'S PARAMETERS

Item	Value
Iteration epochs	400
No. of neurons in each layer	64
No. of hidden layers	2
No. of neurons in input layer	48
No. of neurons in output layer	3
Optimizer	Adam

B. Estimation Results

Fig. 6 shows the MAPE and RMSE of different methods on the test set, where different color is used to indicate the methods belonging to different groups. The labels on y-axis are the acronyms of methods which are listed in Table I. For the regression-based methods, the capital A is at the suffix, which indicates that they are all processed by the proposed pre- and post-event adjustment.

The exponential moving average method has the worst performance. Even less sophisticated, similar day-based methods display better estimation accuracy than it. And Low4of5 has the best accuracy among them. Low4of5 can exclude the unusually high consumption day from the baseline computation, while Mid4of5, High4of5, and exponential moving average methods take this day into account.

The regression-based methods perform better than the other two kinds of methods. Although they also consider the unusually high consumption day, the fitting models can learn this exception by assigning less weight on it. Among them, all clustering-based methods have better performance than the fully aggregated method. So, by utilizing sub-profiles provided by smart meters, the similarity of consumption patterns among customers can be understood. Therefore, customers are aggregated in a more reasonable way. And better estimation performance is achieved.

Usually, clustering-based ensemble method produces better result than an individual model. The G-EA and H-EA achieve lower MAPE than corresponding clustering-based estimations. Among all the candidates, the proposed method obtains the lowest MAPE and RMSE scores. Compared with the best performance comparison candidate, the proposed method has an improvement of 7% in terms of the MAPE, and 11% in terms of the RMSE. Also, even without the weight selection, for the 40 individual contextual bandit-based models with a particular DNN's weight, the largest values of MAPE and RMSE on the test set are 4.83 and 6.35, respectively. They are smaller than that of the comparison candidates as shown in Fig. 6. This is because either the ensemble or the clustering-based methods treat customer segmentation and estimation as two separate problems with different goals. The customer segmentation aims to minimize the dissimilarity among customers, while the estimation aims to improve the estimation accuracy. Without the closed-loop feedback, the clustering cannot learn the best way to divide customers for improving the estimation performance. In the proposed closed-loop method, the two sub-problems are optimized toward a common goal, and the exploration and exploitation mechanism of the

proposed method enables the agent to find the optimal way to divide customers for improving ABL estimation accuracy. Apart from the evaluation metrics reflecting accuracy (MAPE and RMSE), the bias [13] of the proposed method is calculated. Bias is defined as the mean error between the estimated loads and true loads [13]. For the proposed method, the bias in the low-price event is 0.58, while the bias in the high-price event is -0.26. Therefore, in the low-price event, the estimated ABL is higher than the actual one overall. Since the DR capacity in the low-price event is calculated by the reduction between the actual response load and the estimated ABL. Therefore, the estimated DR capacity of DRA is lower than the actual one. Likewise, in the high-price event, the DR capacity is calculated by the reduction between the estimated ABL and the actual response load. Since the ABL is estimated lower than the real one overall, the estimated DR capacity of DRA is lower than the real one. So, the system operator gives lower incentive to the DRA, which is beneficial to the system operator.

Moreover, we compare the computation time of the comparison candidates in Fig. 7. It is observed that the exponential moving average and similar day-based methods take the smallest computation time. The computation time of the k-means, hierarchical clustering, and GMM-based estimations are similar. With the increase of the cluster numbers, the number of regression models increases. Therefore, the computation time increases correspondingly. Also, since the clustering-based ensemble method combines the results of individual models, it is obvious that it takes the largest time among the comparison candidates.

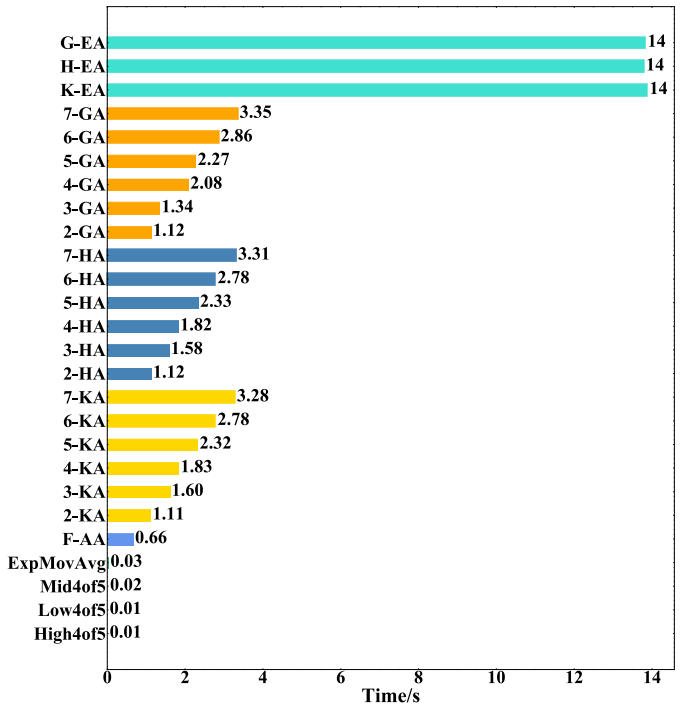

Fig. 7. Computation time comparisons of different methods.

C. The Effect of DNN's Weight Selection and Pre- and post-event Adjustment

To demonstrate the effectiveness of the proposed method, following the "principles of controlling variables", the effect of the weight selection method and pre- and post- event adjustment method is demonstrated separately. Firstly, to prove the effect of the DNN's weight selection method, we compare the MAPE on validation and test sets of the proposed method with that of individual contextual bandit-based method with a particular DNN's weight. Here, the individual methods are the selected N methods which have the first N lowest MAPE scores on the validation set. Note that, in this case, whether using the ensemble technique is the controlling variable.

The histograms in Fig. 8 shows the distributions of MAPE scores on validation and test sets, and the vertical red lines show the MAPE scores of the proposed method on those two sets. The proposed method has the lowest MAPE score on the validation set. Due to the generalization error, the proposed method does not have the best performance on the test set. However, it is still better than 90% individual methods in terms of the MAPE score. Moreover, for the individual model producing the lowest MAPE score (4.88) on the validation set, the MAPE score of it on the test set is 4.68, which is worse than 50% individual methods. So, relying on a single model with a particular weight cannot ensure the best performance on the unseen data. Due to the generalization error, the ensemble method may not produce the best result on the unseen test set data. However, since the ensemble method takes the advantage of all individual methods, it is more robust and reliable than the best performance model on the validation set.

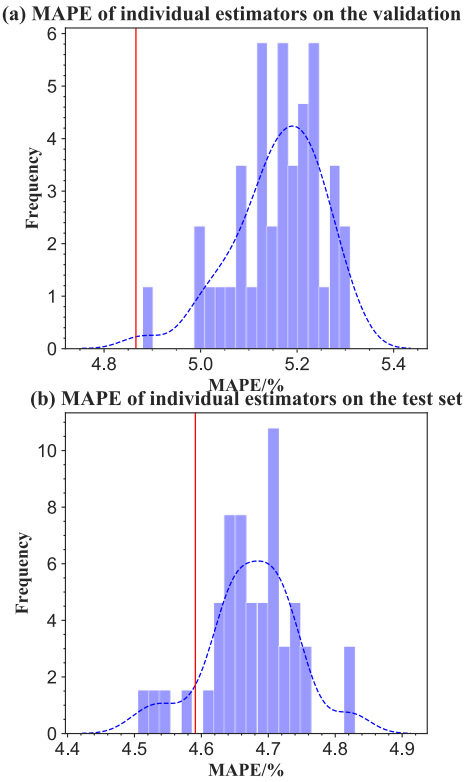

Fig. 8. Histograms of MAPE scores of individual methods on validation and test sets.

Moreover, to prove the effectiveness of the proposed pre- and post-event adjustment, the proposed method is compared with its counterparts without the pre- and post-event adjustment.

For the proposed method, the pre- and post-event adjustment can reduce the MAPE by 10% (5.06 vs. 4.59). It demonstrates that the proposed adjustment method can cope with the daily variation of load profile. The method utilizes the ratio of true load to the estimated load during non-event hours in the DR day, and therefore, the estimation error due to the model is eliminated to some extents.

Fig. 9 shows the actual ABL and estimated ABL obtained by the proposed method with and without pre- and post-event adjustment. It shows different price events, event durations, and start time. For the event happening in the early morning when the load pattern is less variable (Fig. 9 (b)), the two methods can well capture the trend. However, the proposed method with pre- and post-event adjustment shows better performance and it almost overlaps with the true load. During day time and evening (Fig. 9 (c), Fig. 9 (d)), the load is more variable and sudden change can be observed, which increases the difficulty for accurate estimation. It shows that the proposed method with pre- and post-event adjustment can learn the sudden change in the consumption pattern. Therefore, through the adjustment, the proposed method can better follow the change and produce relatively accurate estimation.

VI. Conclusion

In this paper, a novel approach for the ABL estimation is proposed. Compared with the existing literature, the ABL estimation is conducted in a closed-loop fashion. The contextual bandit is utilized to link the customer segmentation with the estimation. The estimation performance is used as the reward for guiding the segmentation, and the segmentation further improves the estimation performance in return. An ensemble method for combining various DNN's weights is proposed. Moreover, a pre- and post-event adjustment method is developed to further improve the estimation accuracy.

Extensive comparisons are conducted. It is shown that the proposed method has stable and good dynamic learning process compared with the other five models. Compared with the similar day-based, exponential moving average, and regression-based methods, the proposed method achieves the lowest MAPE and RMSE scores on the test set, mainly because of the closed-loop feedback mechanism under the contextual bandit's framework. Compared with individual contextual bandit-based method with a particular DNN's weight, the proposed weight selection method further improves the robustness against the generalization error. Therefore, the proposed method has lower MAPE than most individual models on test set. Moreover, compared with the method without the adjustment, the proposed adjustment method further reduces the MAPE by 10%.

Future work can be conducted from two sides. Firstly, the current method performs the deterministic estimation. In order to better quantify the uncertainty, further research can focus on extending the proposed method to the probabilistic estimation. Secondly, since the aggregated load forecasting also involves the customer segmentation and forecasting processes, it is also interesting to explore whether the closed-loop method can improve the aggregated load forecast accuracy.

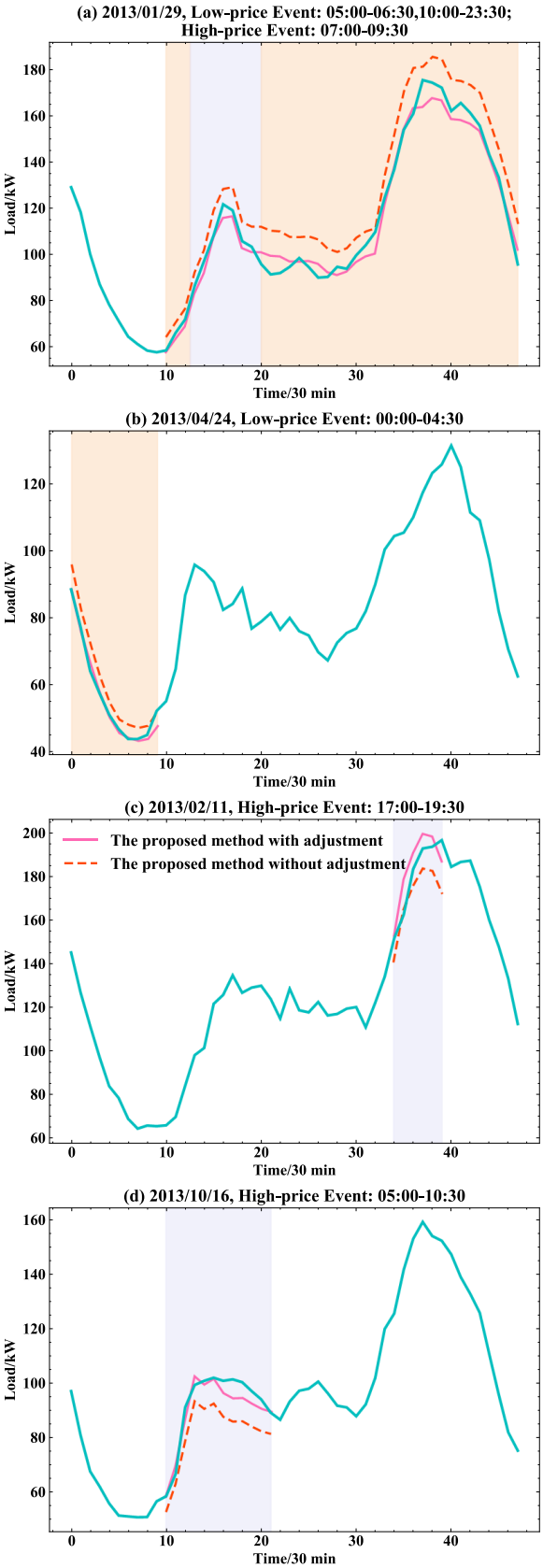

Fig. 9. Actual and estimated baseline loads by the proposed method with and without the adjustment (The blue line: actual ABL; orange area: low-price event; purple area: high-price event).

APPENDIX

A. Feature Selection based on Synchronous Information

For a given ToU customer i, firstly, the daily load data $\boldsymbol{d}_{i,h}, h \in \Omega^E$ is normalized into the range of $[0,1]$ by $\hat{\boldsymbol{d}}_{i,h} = \boldsymbol{d}_{i,h} / \max(\boldsymbol{d}_{i,h})$, where $\hat{\boldsymbol{d}}_{i,h}$ is the normalized load profile of customer i in day h. Then, it is divided into non-event data $\hat{\boldsymbol{d}}_{i,h}^{base} \in \mathrm{R}^{1\times|c_h^b|}$ and event data $\hat{\boldsymbol{d}}_{i,h}^{event} \in \mathrm{R}^{1\times|c_h^e|}$, respectively. Similarly, for ToU customer $k\,(k=1,...,N_{control})$ in the control group, the day h's load profile is firstly normalized and then divided into the corresponding two parts: $\hat{\boldsymbol{d}}_{k,h}^{b} \in \mathrm{R}^{1\times|c_h^b|}, \hat{\boldsymbol{d}}_{k,h}^{e} \in \mathrm{R}^{1\times|c_h^e|}$. Based on the similar consumption pattern matching principle, users with similar load profiles tend to have similar living habits, and therefore their synchronous baseline loads are also similar. So, for the ToU customer i, the input feature at the time stamp $t \in c_h^e$ is the same moment' s baseline loads of the selected Y control group's non-ToU customers with similar load profiles. The details of feature selection can be summarized into the two steps:

Step 1: Given a day h and for $k=1,...,N_{control}$, the Euclidean distance is calculated between non-event hours' load profiles of the ToU customer i and the non-ToU customer k in the control group:

$$dist\left(\hat{\boldsymbol{d}}_{i,h}^{b}, \hat{\boldsymbol{d}}_{k,h}^{b}\right) = \sqrt{\sum_{t\in c_h^b} (\hat{d}_{i,h,t}^{b} - \hat{d}_{k,h,t}^{b})^2} \qquad (21)$$

Step 2: The set formed by Y non-ToU customers with the first Y smallest distances is denoted as $\{k_v\}_{v=1}^{Y}$. Therefore, the input feature of the ToU customer i in day h at hour $t \in c_h^e$ is $\left\{\hat{d}_{k_v,h,t}^{e}\right\}_{v=1}^{Y}, t \in c_h^e$.

REFERENCES

[1] M. Shafie-khah, P. Siano, J. Aghaei, et al, "Comprehensive Review of the Recent Advances in Industrial and Commercial DR," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 3757-3771, July 2019.

[2] Z. Yi, Y. Xu, W. Gu, et al, "A Multi-Time-Scale Economic Scheduling Strategy for Virtual Power Plant Based on Deferrable Loads Aggregation and Disaggregation," *IEEE Transactions on Sustainable Energy*, vol. 11, no. 3, pp. 1332-1346, July 2020.

[3] D. Chassin, D. Rondeau, "Aggregate modeling of fast-acting demand response and control under real-time pricing," *Applied energy*, vol. 181, pp. 288-298, Nov 2016.

[4] Q. Shi, C. Chen, A. Mammoli, et al, "Estimating the Profile of Incentive-Based Demand Response (IBDR) by Integrating Technical Models and Social-Behavioral Factors," *IEEE Transactions on Smart Grid*, vol. 11, no. 1, pp. 171-183, Jan. 2020.

[5] V. Azarova, D. Engel, C. Ferner, et al, "Exploring the impact of network tariffs on household electricity expenditures using load profiles and socio-economic characteristics," *Nature Energy*, vol. 3, no. 4, pp. 317-325, March 2018.

[6] S. Fan, Z. Li, L. Yang, et al, "Customer directrix load-based large-scale demand response for integrating renewable energy sources," *Electric Power Systems Research*, vol. 181, 2020. DOI: https://doi.org/10.1016/j.epsr.2019.106175.

[7] T. K. Wijaya, M. Vasirani and K. Aberer, "When Bias Matters: An Economic Assessment of Demand Response Baselines for Residential Customers," *IEEE Transactions on Smart Grid*, vol. 5, no. 4, pp. 1755-1763, July 2014.

[8] Y. Zhang, W. Chen, R. Xu and J. Black, "A Cluster-Based Method for Calculating Baselines for Residential Loads," *IEEE Transactions on Smart Grid*, vol. 7, no. 5, pp. 2368-2377, Sept. 2016

[9] Y. Wi, J. Kim, S. Joo, et al, "Customer baseline load (CBL) calculation using exponential smoothing model with weather adjustment," in *Proc. 2009 Transmission & Distribution Conference & Exposition: Asia and Pacific*, Seoul, South Korea, 2009, pp. 1-4.

[10] J. Priolkar, E. Sreeraj, A. Thakur, "Analysis of Consumer Baseline for Demand Response Implementation: A Case Study," in *Proc. 2020 7th International Conference on Signal Processing and Integrated Networks*, Toronto, Canada, 2020, pp. 89-94.

[11] Y. Chen, P. Xu, Y. Chu, et al, "Short-term electrical load forecasting using the Support Vector Regression (SVR) model to calculate the demand response baseline for office buildings," *Applied Energy*, vol. 195, pp. 659-670, June 2017.

[12] M. Sun, Y. Wang, F. Teng, et al, "Clustering-Based Residential Baseline Estimation: A Probabilistic Perspective," *IEEE Transactions on Smart Grid*, vol. 10, no. 6, pp. 6014-6028, Nov. 2019.

[13] E. Lee, K. Lee, H. Lee, et al, "Defining virtual control group to improve customer baseline load calculation of residential demand response," *Applied Energy*, vol. 250, pp. 946-958, September 2019.

[14] Y. Zhang, Q. Ai, Z. Li, "Intelligent Demand Response Resource Trading using Deep Reinforcement Learning," *CSEE Journal of Power and Energy Systems*, to be published.

[15] F. Wang, B. Xiang, K. Li, *et al*., "Smart Households' Aggregated Capacity Forecasting for Load Aggregators Under Incentive-Based Demand Response Programs," *IEEE Transactions on Industry Applications*, vol. 56, no. 2, pp. 1086-1097, March-April 2020.

[16] Y. Zhang, Q. Ai, Z. Li, " Improving Aggregated Baseline Load Estimation by Gaussian Mixture Model," *Energy Reports*, vol. 6, no. 9, pp. 1221-1225, December 2020.

[17] T. K. Wijaya, M. Vasirani, S. Humeau and K. Aberer, "Cluster-based aggregate forecasting for residential electricity demand using smart meter data," in *Proc. 2015 IEEE International Conference on Big Data (Big Data)*, Santa Clara, CA, 2015, pp. 879-887.

[18] Y. Wang, Q. Chen, M. Sun, et al, "An Ensemble Forecasting Method for the Aggregated Load With Subprofiles," *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 3906-3908, July 2018.

[19] S. Li, L. Goel, P. Wang, "An ensemble approach for short-term load forecasting by extreme learning machine," *Applied Energy*, vol. 170, pp. 22-29, May 2016.

[20] A. Bagherjeiran, C. F. Eick, Chun-Sheng Chen and R. Vilalta, "Adaptive clustering: obtaining better clusters using feedback and past experience," in *Proc. Fifth IEEE International Conference on Data Mining (ICDM'05)*, 2005, pp. 1-4.

[21] Y. Wang, Q. Chen, C. Kang and Q. Xia, "Clustering of Electricity Consumption Behavior Dynamics Toward Big Data Applications," *IEEE Transactions on Smart Grid*, vol. 7, no. 5, pp. 2437-2447, Sept. 2016.

[22] S. Aleksandrs, "Introduction to multi-armed bandits," *arXiv preprint arXiv:1904.07272*, 2019.

[23] X. Chen, Q. Hu, and Q. Shi, et al, "Residential HVAC Aggregation Based on Risk-averse Multi-armed Bandit Learning for Secondary Frequency Regulation," *Journal of Modern Power Systems and Clean Energy*, vol. 8, no. 6, pp. 1160-1167, November 2020.

[24] M. Khodayar, G. Liu, J. Wang and M. E. Khodayar, "Deep learning in power systems research: A review," *CSEE Journal of Power and Energy Systems*, vol. 7, no. 2, pp. 209-220, March 2021.

[25] Y. Li, "Deep reinforcement learning: An overview," *arXiv preprint arXiv:1701.07274*, 2017.

[26] E. Mocanu, D. Mocanu, P. Nguyen, et al., "On-Line Building Energy Optimization Using Deep Reinforcement Learning," *IEEE Transactions on Smart Grid*, vol. 10, no. 4, pp. 3698-3708, July 2019.

[27] A. Agarwal, M. Dudík, S. Kale, et al, "Contextual bandit learning with predictable rewards," *arXiv preprint arXiv:1202.1334*, 2012.

[28] D. Bouneffouf, I. Rish, G. Cecchi, and R. Feraud, "Context attentive bandits: Contextual bandit with restricted context," *arXiv preprint arXiv:1705.03821*, 2017.

[29] Y. Zhang, Q. Ai, and Z. Li, "ADMM-based distributed response quantity estimation: a probabilistic perspective," *IET Generation, Transmission & Distribution*, vol. 14, no. 26, pp. 6594-6602, Dec. 2020.

[30] J. Schofield, S. Tindemans, R. Carmichael, M. Woolf, M. Bilton, and G. Strbac, "Low carbon london project: Data from the dynamic time-of-use electricity pricing trial, 2013," *Tech. Rep*., Jan. 2016.

[31] V. Mnih, P. Badia, M. Mirza, et al, "Asynchronous methods for deep reinforcement learning," *International conference on machine learning*, pp. 1928-1937, June 2016.

[32] Y. Wang, Q. Chen, T. Hong and C. Kang, "Review of Smart Meter Data Analytics: Applications, Methodologies, and Challenges," *IEEE Transactions on Smart Grid*, vol. 10, no. 3, pp. 3125-3148, May 2019.