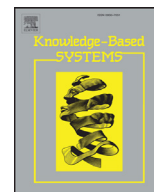




ELSEVIER

Contents lists available at ScienceDirect

## Knowledge-Based Systems

journal homepage: [www.elsevier.com/locate/knosys](http://www.elsevier.com/locate/knosys)

# A hybrid-learning based broker model for strategic power trading in smart grid markets

Xishun Wang\*, Minjie Zhang, Fenghui Ren

School of Computing and Information Technology, University of Wollongong, Australia

## ARTICLE INFO

### Article history:

Received 20 June 2016

Revised 5 December 2016

Accepted 7 December 2016

Available online xxx

### Keywords:

Smart grid market

Broker model

Data-driven

Hybrid learning

## ABSTRACT

Smart Grid markets are dynamic and complex, and brokers are widely introduced to better manage the markets. However, brokers face great challenges, including the varying energy demands of consumers, the changing prices in the markets, and the competitions between each other. This paper proposes an intelligent broker model based on hybrid learning (including unsupervised, supervised and reinforcement learning), which generates smart trading strategies to adapt to the dynamics and complexity of Smart Grid markets. The proposed broker model comprises three interconnected modules. Customer demand prediction module predicts short-term demands of various consumers with a data-driven method. Wholesale market module employs a Markov Decision Process for the one-day-ahead power auction based on the predicted demand. Retail market module introduces independent reinforcement learning processes to optimize prices for different types of consumers to compete with other brokers in the retail market. We evaluate the proposed broker model on Power TAC platform. The experimental results show that our broker is not only is competitive in making profit, but also maintains a good supply-demand balance. In addition, we also discover two empirical laws in the competitive power market environment, which are: 1. profit margin shrinks when there are fierce competitions in markets; 2. the imbalance rate of supply demand increases when the market environment is more competitive.

© 2016 Elsevier B.V. All rights reserved.

## 1. Introduction

Smart Grid markets are complex and dynamic. The complexity is ascribed to the various participators and interactions between multiple stakeholders. There are various participants, including large energy generators, general consumers, interruptible consumers and storage consumers, and even renewable energy producers, such as solar power systems and wind turbines. In such a two-way power flow system, there are multiple interactions between consumers, prosumers<sup>1</sup> and energy suppliers. The dynamics are caused by the varying energy demands, changing prices and customer migrations. The energy demands can vary from time to time because of the fluctuations of energy needs by time and the mutable weather conditions. Prices may also change according to the designed pricing mechanisms [11], or the energy supply and demand status. The autonomous end users may switch tariffs

based on their own utilities. Due to the complexity and dynamics, it is of great challenge to manage Smart Grid markets.

To ameliorate the management of such complex markets, brokers are widely employed. Brokers, who buy energy from the wholesale market and sell it to the retail market, form a decentralized mode to enhance the efficacy of Smart Grid markets. In the power trading, brokers simultaneously interact with the wholesale and retail markets. In the wholesale market, brokers buy energy 24 h ahead through auctions. There are different energy suppliers, such as thermal power generators, hydropower generators and wind power generators. The various energy suppliers exhibit different prices, quantities and stabilities. In the retail market, brokers try to attract more customers and sell out their energy to make more profit. Different consumers have different requirements on the price, quantity, quality and time of energy usage. A successful broker should not only maximize his own profit, but also keep a good supply-demand balance in the two markets to improve the energy efficiency. However, the excellent broker has to cope with the omnifarious challenges. For the wholesale market, there are dynamics in energy price, quantity and stability because of the various energy suppliers. To purchase a proper amount of energy for the each of coming 24 h, a customer demand prediction is needed, but is a very challenging issue in Smart Grid due to

\* Corresponding author.

E-mail addresses: [xw357@uowmail.edu.au](mailto:xw357@uowmail.edu.au) (X. Wang), [minjie@uow.edu.au](mailto:minjie@uow.edu.au) (M. Zhang), [fren@uow.edu.au](mailto:fren@uow.edu.au) (F. Ren).

<sup>1</sup> Prosumer refers to the market participant as both a producer and a consumer, first used by Alvin Toffler in 1970.

the various behaviors of different customers and customers' migrations between brokers. Moreover, a bidding strategy is required to optimize the bidding prices in auctions, with the considerations of competitions among other brokers. For the retail market, there are variety of consumers with various behaviors. An excellent broker should consider the different types of consumers and their different power usages. Besides, the broker should also deliberate on the dynamics and uncertainty of customers' behaviors in power usage. Moreover, the broker also needs an efficient strategy to compete with other brokers to attract customers in the retail market.

To construct a systematic broker model and to efficiently surmount the challenges in Smart Grid markets, three goals are established in our designation of a good broker model: **Goal 1** is to efficiently predict the energy demand of customers, so as to keep a good balance of supply and demand; **Goal 2** is to obtain energy as the required demand in the wholesale market with a lower price; and **Goal 3** is to sell energy to customers with a proper price, which can ensure a good profit and attract customers. This paper proposes a broker model with effective methods to achieve the above three goals. For Goal 1, a data-driven method is proposed to first cluster various customers according to their energy consumption patterns, and then predict the one-day-ahead hourly demand of subscribed customers. For Goal 2, the Markov Decision Process (MDP) is employed for energy auctions in the wholesale market. For Goal 3, independent reinforcement learning processes are introduced to optimize prices for different types of customers. The proposed broker model is evaluated on the platform of Power Trading Agent Competition [8] (Power TAC), which supplies a practical simulation of the complex Smart Grid markets based on real data.

The proposed broker model contributes to the research in Smart Grid markets in four aspects. 1) A systematic framework of broker model is designed to simultaneously balance supply and demand and optimize energy prices in both the wholesale and retail markets. This framework will enlighten the designations of future broker models in Smart Grid markets. 2) Hybrid learning is introduced as an effective way to adapt to the dynamics in markets. Unsupervised, supervised and reinforcement learning approaches are integrated to construct a systematic model, which can efficiently adapt to the dynamics in Smart Grid markets. The experiments have demonstrated that the proposed broker model works better than the previous models. 3) A new data-driven customer demand prediction method is proposed. This method deeply explores the energy consumption patterns of various customers, and then integrates supervised learning to predict the future energy demand. The proposed method provides an efficient way to predict the energy demand for a variety of customers, and can be extended to demand prediction in a market level. 4) Independent SARSA processes [2] are used for different consumers in the retail market, and the experimental results have demonstrated that it is an effective way to compete with other brokers.

## 2. Related work

Broker modeling in Smart Grid markets is an emerging research field and there have not been much literatures. In 2012, Power TAC started and supplied a simulated real-world Smart Grid market environment. Some broker models have been developed since then, but there have been not many available literatures. AstonTAC team [9] introduced MDP approach for auctions in the wholesale market, and employed different HMMs to predict the price of energy and the customer demand. The AstonTAC can keep a good supply-demand balance, but it does not take effective strategy to attract customers in the retail market. Urieli and Stone [22] developed a broker model called TacTex and won the Power TAC in 2013. They decomposed the global optimization into sub-optimizations in the

wholesale and retail markets. Locally weighted linear regression was introduced to predict if the customers would subscribe his tariffs. The TacTex wins in profit making, but it does not make much effort on supply-demand balance. The CwiBroker team [10] used game theories in both wholesale and retail markets to maximize the profit. In contrast, our broker model takes both profit making and supply-demand balance into considerations, resulted in a more comprehensive method in coping with the complexity and dynamics in Smart Grid markets.

Demand prediction has been intensively studies in Smart Grid [19,23]. A variety of models have been proposed, including time series models[6], ARIMA [17], neural networks [4] and so on. Recently, some novel learning-based methods have been proposed. Srinivasan [18] introduces a group method of data handling (GMDH) neural network for mid-term energy demand prediction. In his method, six categories of consumers are predicted respectively, yet the customer groups are stipulated manually. Amjady et al. [3] uses a bilevel method, which is composed of a feature selection technique and a forecasting engine, to predict the demand of a single micro-grid. Their method has been tested on the demand prediction of a campus. However, a large scope of customers, especially for the market level, has not been demonstrated in their method. Motamedi et al. [12] combine a multi-input multi-output forecasting engine for joint price and demand prediction with data association mining algorithms, through which the relationship of demand and price is extracted. This method is applied to a macro scope, regardless the types of customers.

Generally, previous prediction methods either focus on all customers as a whole or a special customer. None of them explores the behaviors of different customers in a market level. The prediction method we proposed is a data driven method, based on the nature of customers. Customers are hierarchically clustered based on their historical usage data. Different usage prediction methods are tailored for customer clusters with different energy consumption patterns. Thus our method is applicable and effective for market level prediction.

Pricing mechanisms have been deeply explored in Smart Grid [5,11,15] with the objective of curtailing peak load, while literatures on pricing strategy on the stand of a broker are rare. Recent work in [13] by Peters et al. used reinforcement learning with function approximation to adapt to the economic signals from the retail market. In their work, a range of market features were studied and effective features are selected. Their work gives a good hint to this research. Different from their work, independent SARSA processes are introduced for different types of customers in our method. The independent SARSA processes can be easily implemented with parallel computing technology for efficiency.

## 3. Definitions and framework design

In this section, the terms that are used in the rest of the paper are defined, and the framework of the proposed broker model is described.

### 3.1. Definitions

**Definition 1** (Bootstrap data). Bootstrap data  $\mathbf{B}_D$  are the historical data of customer usages in the retail market. It is represented as the following matrix,

$$\mathbf{B}_D = \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1T_b} \\ u_{21} & u_{22} & \cdots & u_{2T_b} \\ \vdots & \vdots & \vdots & \vdots \\ u_{N1} & u_{N1} & \cdots & u_{NT_b} \end{pmatrix}, \quad (1)$$

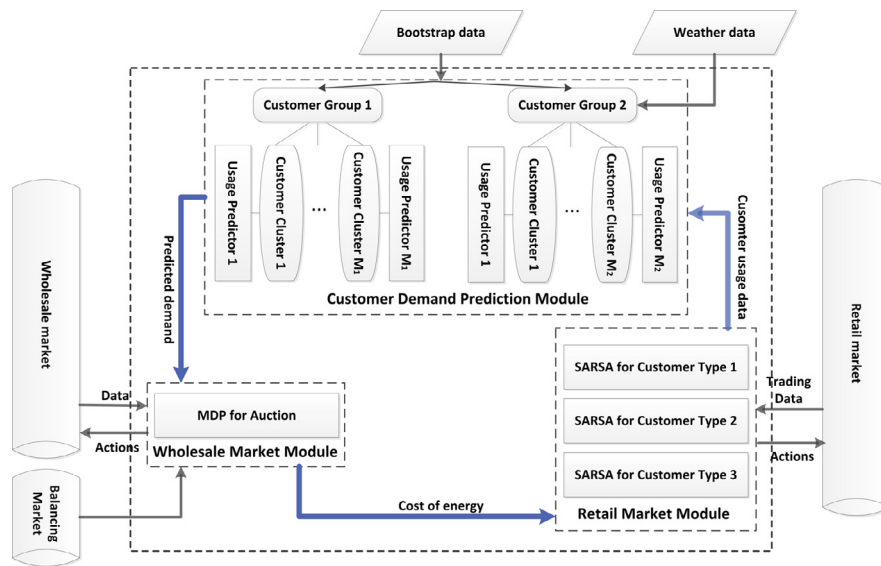


Fig. 1. The framework of the proposed broker model.

where  $u_{ij}$  is the hourly usage of customer  $i$  in hour  $j$ ,  $N$  is the total number of customers, and  $T_b$  is the length of a usage sequence of customer  $i$ .

**Definition 2** (Usage gradient). Usage gradient  $\mathbf{G}_i = [g_1, g_2, \dots, g_{24}]$ , is a 24-dimension vector that reflects how the hourly usage amount of customer  $i$  changes in 24 h. Each element  $g_k$  is calculated in a usage sequence of customer  $i$ .

Assuming there is a  $N$  day length usage sequence  $\{u_k^n\}$ , where  $u_k^n$  is the usage of hour  $k$  in day  $n$ .  $g_k$  is the average of  $g_k^n$ ,  $g_k = \frac{1}{N} \sum_{n=1}^N g_k^n$ , where  $g_k^n$  is calculated by  $g_k^n = (u_k^n - u_{k-1}^n) / \bar{u}^n$  (for  $u_0^n$  the usage of hour 0 is the usage of hour 24 of last day), where  $\bar{u}^n$  is the average usage of the 24 h in day  $n$ .

**Definition 3** (Usage variance). Usage variance  $\mathbf{V}_i = [v_1, v_2, \dots, v_{24}]$ , is a 24-dimension vector regarding customer  $i$ . Each element  $v_k$  reflects how much the usage in hour  $k$  varies in different days.  $v_k$  is calculated in a usage sequence of customer  $i$ .

In a  $N$  day length usage sequence  $\{u_k^n\}$ ,  $v_k$  is a standard variance, calculated by  $v_k = \sqrt{\frac{1}{N} \sum_{n=1}^N (u_k^n - \bar{u}_k)^2}$ , where  $\bar{u}_k$  is the average of usage in hour  $k$  of  $N$  days.

**Definition 4** (Energy consumption pattern). Energy consumption pattern  $\mathbf{P}_i = \{\mathbf{G}_i, \mathbf{V}_i, T\}$  for customer  $i$ , is defined as a collection of usage gradient  $\mathbf{G}_i$ , usage variant  $\mathbf{V}_i$  and time period  $T$ . It indicates how a customer consumes energy in a certain time period.

3.2. Framework design

The broker model is designed to efficiently adapt to the dynamics in both the retail and wholesale markets. To achieve the three goals set in Section 2, we design three modules in our broker model, which are: 1) customer demand prediction module, 2) wholesale market module, and 3) retail market module. The three modules incorporate to cope with market dynamics. The customer demand prediction module predicts one-day-ahead customer demand, and outputs the predicted demand to the wholesale market module. The wholesale market module aims to obtain such amount of energy with a low price in auctions. Then energy cost is calculated and passed to the retail market module. Based on the known energy cost, the retail market module makes decisions to attract customers and gain profit. The framework of the proposed broker model is shown in Fig. 1.

The three modules work as follows.

Customer demand prediction module

The customer demand prediction module makes use of the data from two stages: the bootstrap stage and the dynamic market stage. It first explores the retail market based on bootstrap data, from which the energy consumption pattern for each customer is calculated. According to the different usage patterns, all the customers are clustered into clusters through a hierarchical clustering process. For each customer cluster, a usage predictor, which is learned from the bootstrap data, is used to predict the future usage. In dynamic market environments, the retail customers who subscribe our tariffs are maintained, and then these customers are assigned to the corresponding customer clusters with respect to their energy consumption patterns. Using the learned usage predictor for each cluster, our broker predicts the energy usage for each customer cluster, and accumulates all the predicted usages to obtain total demand.

Wholesale market module

Given the predicted demand, the wholesale market module employs a Markov Decision Process (MDP) to bid for the predicted amount of energy in one-day-ahead auction in the wholesale market. If the obtained energy amount is less than the actual usage, the imbalance amount will be compensated by the balancing market with a balancing price,<sup>2</sup> which might be much higher than the market price. After the auction, the cost of the obtained energy is calculated and passed to the retail market module.

Retail market module

The retail market module calculates the profit based on the given energy cost from the wholesale market module. To compete with other brokers, this module uses three independent SARSA control processes to optimize the tariff prices for three types of customers, which are prosumers, general consumers (customers who require consistent power supply, such as householders and office users) and interruptible consumers (customer with energy storage capacity, such

<sup>2</sup> The balancing price is for the power from high-level grid, for example, the state grid. As the delivery distance is long, balancing price may be much higher than the market price.

as cold storage companies). The calculated profit is regarded as the immediate reward in the SARSA process.

From the perspective of an integrated system, the three modules are connected and can exchange information. The information flow is highlighted in Fig. 1. The customer demand prediction module gathers customer usage data from retail market module and outputs the predicted energy demand to the wholesale market module. The wholesale market module tries to obtain the predicted amount of energy in the auction process, and passes the energy cost to the retail market. The retail market module uses the cost of energy in reinforcement learning to calculate rewards and makes decisions to different customers. The retail market module also maintains the data of customers for the customer demand prediction module. Through the communications and cooperations of the three modules, the broker model can efficiently response to the dynamics in markets, so as to make more profit and keep supply-demand balance. In the following section, technical designs of the three modules are introduced in details.

#### 4. The detail design of three modules

This section introduces the three modules of the proposed broker model in details.

##### 4.1. The detail design of customer demand prediction module

In this module, a data-driven method is proposed to predict customer demand, which is critical for supply-demand balancing. The data-driven method hierarchically clusters the customers based on their energy consumption patterns, and then different prediction methods are introduced to predict the energy demand for different customer clusters 24 h ahead. The procedure of customer demand prediction is described in Algorithm 1.

---

**Algorithm 1** A data-driven method for customer demand prediction.

---

**Input:** Bootstrap data  $\mathbf{B}_D$ ;

**Output:** Predicted energy demand  $U_t, t = 1, \dots, 24$ ;

- 1: **Initialize:**  $U_t = 0, t = 1, \dots, 24$ ;
  - 2: Calculate customer energy consumption pattern  $\mathbf{P}_i$  for each customer based on  $\mathbf{B}_D$ ;
  - 3: Cluster the customers into customer group  $C_1$  and  $C_2$  according to customer usage variance  $\mathbf{V}_i$ ;
  - 4: Cluster the customers in  $C_1$  and  $C_2$  according to customer usage gradient  $\mathbf{G}_i$ , and  $M_1$  and  $M_2$  customer clusters are obtained, respectively;
  - 5: Assign the subscribed customers into the  $M$  customer clusters;
  - 6: **for** each hour  $t \in [1, 24]$  **do**
  - 7:   **for** each customer cluster  $c_i \in C_1$  **do**
  - 8:     Predict future usage  $U_t^i$  based on  $\mathbf{B}_D$ ;
  - 9:      $U_t = U_t + U_t^i$ ;
  - 10:   **end for**
  - 11:   **for** each customer cluster  $c_i \in C_2$  **do**
  - 12:     Predict  $U_t^i$  with the regressor learned from  $\mathbf{B}_D$ ;
  - 13:      $U_t = U_t + U_t^i$ ;
  - 14:   **end for**
  - 15: **end for**
- 

In Algorithm 1, The broker model predicts next day energy demand  $U_t (t = 1, \dots, 24)$  according to the input bootstrap data  $\mathbf{B}_D$ . In Line 1, the energy demands to be predicted are initialized to 0. Line 2 calculates the energy consumption pattern  $\mathbf{P}_i$  of each customer. Then customers are clustered according to  $\mathbf{P}_i$  in Lines 3–4. Line 3 clusters the customers into customer group  $C_1$  and  $C_2$  according to customers' usage variances, and Line 4 further clusters

the customers in  $C_1$  and  $C_2$  according to customers' usage gradients, respectively. The customers who subscribe our tariffs are assigned into the customer clusters in Line 5. In Lines 7–10, the energy demand for each customer cluster in customer group  $C_1$  is predicted and accumulated to the total energy demand. In Lines 11–14, similar routine is executed for each customer cluster in customer group  $C_2$ . The details of clustering customers (Lines 3–4), prediction method in  $C_1$  (Line 8), and prediction method in  $C_2$  (Line 12) are further introduced as follows.

**Clustering customers:** According to pattern  $\mathbf{P}_i$  obtained in Line 2, a hierarchical clustering method is used to cluster all the customers. There are two layers in our cluster method. For the first layer, customers are clustered according to usage variance  $\mathbf{V}_i$ , and two customer groups are obtained, i.e.  $C_1$  and  $C_2$  (see Line 3). For the second layer, customers in  $C_1$  and  $C_2$  are further clustered according to usage gradient  $\mathbf{G}_i$ , and  $M_1$  and  $M_2$  clusters are formed, respectively (see Line 4).

In the first layer, it is aimed to cluster all the customers into two groups, where one group of customers ( $C_1$ ) shows stable hourly usages in different days, and another group ( $C_2$ ) has variable hourly usages. A hard threshold is used to cluster all the customers in the first layer. For customer  $i$ , if any element  $v_k$  in his usage variance  $\mathbf{V}_i$ , satisfies  $v_i \leq \Delta_1$ , the customer is assigned into  $C_1$ . Otherwise, the customer will be assigned into  $C_2$ . A strict threshold  $\Delta_1$  is used to ensure that the hourly usage of customers in  $C_1$  is stable. The cluster procedure in the first layer makes sense for the reason that we can tailor two prediction methods for the customers in  $C_1$  and  $C_2$ . As customers in  $C_1$  show stable hourly usages everyday, the future hourly usages can be predicted according to the historical usage data.

In contrast, the hourly usages of customers in  $C_2$  change with respect to other factors, thus supervised learning can be introduced to learn predictors for those customers. In the second layer, we aim to find customers sharing similar usage gradients.  $K$ -means [7] with a Euclidean distance criterion  $\Delta_2$  is used to cluster the customers in  $C_1$  and  $C_2$ , respectively. It is obvious that  $\Delta_2$  affects the numbers of final clusters, i.e.  $M_1$  and  $M_2$ . The parameter  $\Delta_2$  is further analyzed in the experiment section (see Subsection D in Section 4). The reason of further clustering customers and the meaning of clusters in the second layer are explained as follows. Similar usage gradients indicate that customers in the same cluster show similar responses to the outside factors, thus one predictor is adequate for one customer cluster.

**Prediction method in  $C_1$ :** Customers in  $C_1$  show stable hourly usages in different days. For this kind of customers, we simply use accumulated historical data to predict their one-day-ahead usage. For the  $M_1$  customer clusters,  $M_1 \times 24$  predictors are constructed. Eq. (2) is designed to predict one-day-ahead usage  $U_{mt}$  of hour  $t$  for cluster  $m$ .

$$U_{mt}^f = \lambda U_{mt}^c + (1 - \lambda) \overline{U_{mt}^h} \quad (2)$$

In Eq. (2),  $U_{mt}^f$  is the future one-day-ahead usage,  $U_{mt}^c$  is the current usage,  $\overline{U_{mt}^h}$  is the average of historical usage within a time window of one week, and  $\lambda$  is a weight parameter. A large  $\lambda$  indicates a strong emphasis on the current usage. Eq. (2) recurs while time goes forward.

**Prediction method in  $C_2$ :** Customers in  $C_2$  show variable hourly usages, which might be affected by weather conditions and dynamics of the retail market. We first consider the influence of weather conditions. A supervised learning method—LASSO regression [21] is used to estimate the future usage. The regressors are learned from the historical usage data and weather data.  $M_2 \times 24$  predictors are learned for each customer clusters in  $C_2$ , respectively. The predictor, which predicts the hourly usage in hour  $t$  for cluster  $m$ , is illustrated as follows. We extract the customer usage

in hour  $t$  of all the customers in cluster  $m$  from bootstrap data  $\mathbf{B}_D$ . A new data matrix  $\mathbf{B}_d$  is obtained,

$$\mathbf{B}_d = \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1L} \\ u_{21} & u_{22} & \cdots & u_{2L} \\ \vdots & \vdots & \vdots & \vdots \\ u_{K1} & u_{K2} & \cdots & u_{KL} \end{pmatrix}, \quad (3)$$

where  $u_{kl}$  is the hourly usage, and  $K$  is the total number of customers in this cluster,  $L$  is the total number of usage records of hour  $t$  contained in a line vector of bootstrap data  $\mathbf{B}_D$ .  $L$  can also represent the length of bootstrap data measured by days. Adding up each line of  $\mathbf{B}_d$ , a vector  $\mathbf{U}$  is obtained,  $\mathbf{U} = [u_1, \dots, u_L]$ . In  $\mathbf{U}$ , each element  $u_l$  is the  $l$ th day hourly usage of all customers in this cluster.  $u_l$  fluctuates with respect to weather conditions. Following the traditional procedure in data mining, we normalize  $u_l$  by deducting the mean value and then dividing the standard variance. We use  $u'_l$  to denote the obtained elements.  $\varphi_t^l$  is from weather forecast of hour  $t$  of the  $l$ th day.  $\varphi_t$  is a 4-dimension real value vector:  $\varphi_t = [T, W, D, C]$ , where  $T$  refers to temperature,  $W$  refers to wind speed,  $D$  refers to wind direction, and  $C$  refers to cloudiness. Each weather feature is also normalized as  $u_l$ . A LASSO regressor uses the following equation to estimate  $u'_l$  given  $\varphi_t^l$ .

$$\tilde{u}'_l = w_t \cdot \varphi_t^l + b \quad (4)$$

In Eq. (4),  $w_t$  is the learned weight vector, and  $b$  is the bias item. A LASSO regressor for time  $t$  is learned as the following process. The weather feature and the corresponding normalized usage at time  $t$  form a pair ( $\varphi_t^l \rightarrow u'_l$ ), which is regarded as a training sample.  $L$  training samples are obtained in the end. We employ LASSO algorithm [21] to minimize the following regularized cost function,

$$J(w_t) = \frac{1}{L} \sum_{l=1}^L (u'_l - \tilde{u}'_l)^2 + \frac{\theta}{2L} \|w_t\|_1 \quad (5)$$

In Eq. (5),  $\theta$  is a parameter to compromise the cost function and the regularization term, and  $\|\cdot\|_1$  is the  $L_1$  norm which can perform feature selection. With the trained regressor, power usage  $u_l$  can be predicted. Finally, all the predicted usages of  $M$  clusters are collected, and the estimated energy demand  $U_t$  for hour  $t$  is obtained.

The performance of the predictors are monitored to adapt to the dynamics in the retail market. If the predicted result differs from the actual usage to more than a certain criterion for a certain time period, the prediction module is updated with the historical data of subscribed customers. For customers in  $C_1$ , the predictors will instantly update their values accordingly with Eq. (2). For customers in  $C_2$ , the average usage  $\bar{u}$  is updated by Eq. (2), and all the predictors for  $C_2$  are re-trained using Eq. (5). In this manner, our predictors can catch up with the dynamics in the retail market.

#### 4.2. The detail design of wholesale market module

In the wholesale market, our design aims to minimize the cost of energy in one-day-ahead auctions. When energy demand  $U_t$  for hour  $t$  is estimated,  $U_t$  is immediately sent to the auction process. Bidding occurs at the beginning of each hour, thus the broker has 24 opportunities to buy energy of amount  $U_t$ . This is a sequential bidding problem. Vytelingum [24] studied continuous auctions in 2008. Tesauro et al. [20] proposed a dynamic programming strategy to optimize the bidding price. In our situation, the auction repeats every 24 h. Thus, it would be beneficial to employ an MDP [14] to model the bidding process. The bidding form is  $bid(Q, p)$ , where  $Q$  is the energy amount and  $p$  is the bidding price. The proposed broker model uses Algorithm 2 for an energy auction process in the wholesale market.

**Algorithm 2** Algorithm for one energy auction in the wholesale market.

**Input:** Historical auction data  $\mathbf{A}_t$ ; Predicted energy demand  $U_t$ ;  
**Output:** Bidding  $bid(Q, p_t)$ ;  
1: **Initialize:**  $Q = U_t$ ;  
2: **if** the length of  $\mathbf{A}_t < l$  **then**  
3:   Set  $p_t$  as the clear price  $p_c$  of the former successful auction  
4: **else**  
5:   Optimize  $p_t$  using MDP [5]  
6: **end if**  
7: **return**  $bid(Q, p_t)$ ;

Algorithm 2 outputs the optimal bidding price according to the historical auction data and predicted energy demand. Line 1 sets the bidding amount  $Q = U_t$ . Lines 2–3 deal with the situation lack of historical data, and the bidding price  $p_t$  is set as the clear price of the former successful auction  $p_c$ . If enough historical data are available, the MDP designed in TaxTac13 [22] is employed for auction in Lines 4–5.

The MDP is defined as a five tuples  $S \times A \times T \times R \times T_S$  as follows.

- **States:**  $s \in \{0, 1, \dots, 24, success\}$ ,  $s_0 := 24$
- **Actions:** bidding price  $p_t \in R^+$
- **Transition:** a state  $s \in \{1, \dots, 24\}$  is transited to the terminal state *success* if a bid is fully or partially cleared. Otherwise, the state  $s$  is transited to state  $s - 1$ . The transition probability is defined by  $P_T(s, p_t) = m'/m$ , where  $m'$  is the total cleared energy amount when the clear price is less than the bidding price, and  $m$  is the total cleared energy amount.  $m'$  and  $m$  are accumulated from the historical data of state  $s$ .
- **Reward:** The reward for states  $s \in \{1, \dots, 24\}$  is 0, and the reward for state  $s = 0$  is the balancing price  $p_b$  (initially unknown, accumulated from historical balancing data). For the state *success*, the reward is the  $p_t$ .
- **Terminal states:**  $\{0, success\}$

The MDP can be solved by a back-sweep, from state 0 to state 24, with the following value function operator:

$$V(s) = \begin{cases} p_b & \text{if } s = 0 \\ \min_{p_t} \{P_T \times p_t + (1 - P_T) \times V(s - 1)\} & \text{if } 1 \leq s \leq 24 \end{cases} \quad (6)$$

At time  $t + 24$ , the bought energy is consumed by customers. Assuming that the actual customer usage is  $Q$ ,  $S_A$  is the cost spent in the auction and  $S_B$  is the cost spent for balancing amount (when the obtained energy in auction is less than the actual customer usage). The per kW energy cost  $W_t$  is computed,  $W_t = (S_A + S_B)/Q$ . The cost  $W_t$  is sent to the retail market module for profit computing.

#### 4.3. The detail design of retail market module

In the dynamic retail market, there are two objectives to be achieved in the competitions, which are: 1) to keep and attract more customers against other brokers; and 2) to make more profit. There are mainly three types of customers, which are prosumers, general consumers and interruptible consumers. Prosumers can produce energy, such as solar systems. Interruptible consumers can tolerate the energy interruptions for a certain period. General consumers require continuous energy supply. As the three types of customers exhibit different attributes in consuming energy, it is reasonable to publish different tariffs for different types of customers. To optimize different tariffs, independent SARSA [2] processes are introduced for different customers. Each SARSA is used to explore the dynamics of its corresponding customers and take

**Table 1**  
Features and feature representations for state  $s$ .

Feature	Definition	Representation
Bias	Constant 1.0	Plain
ActionIndex	Index of selected action	Plain
MarketBreadth	Range from the lowest to highest rate	RBF
MarketShare	Customer percentage of the market	Plain

**Table 2**  
Actions and action definitions.

No.	Action	Definition
$a_1$	<i>Maintain</i>	keep the same price at time $t + 1$ as at time $t$
$a_2$	<i>Max</i>	publish a tariff with price $p_h + \varepsilon$ , where $p_h$ is the highest price, $\varepsilon$ is a small value
$a_3$	<i>Min</i>	publish a tariff with price $p_l - \varepsilon$ , where $p_l$ is the lowest price
$a_4$	<i>Increase</i>	publish a tariff with price $p_c + \delta$ , where $p_c$ is the current price, $\delta$ is the amount to increase price
$a_5$	<i>Decrease</i>	publish a tariff with price $p_c - \delta$
$a_6$	<i>Average</i>	publish a tariff with an average market price $p_a$

proper actions (optimizing energy price). We take one SARSA process as an instance to detail our strategy in the retail market. This process is illustrated in Algorithm 3.

**Algorithm 3** SARSA control for a type of customer in the retail market.

- 1: **Initialize:**  $Q(s, a) = 0$ ,  $a = a_6$  (See Table 2);
- 2: **Repeat:**
- 3: Take action  $a$ , and observe the reward  $r$  and the next state  $s'$ ;
- 4: Calculate  $Q(s, a)$  according to Equation 7;
- 5: Choose the next action  $a'$  with  $\epsilon$ -greedy policy based on  $Q(s, a)$ ;
- 6: Update:  $a = a'$ ,  $s = s'$ ;

In Algorithm 3, Line 1 initializes  $Q(s, a)$  and  $a$ . Lines 3–6 repeat for the life time of the broker model. Line 3 calculates the immediate reward  $r$  and next state feature  $s'$  after taking action  $a$ . Line 4 calculates  $Q(s, a)$  according to Eq. (7), which will be introduced later. Line 5 chooses next action using  $\epsilon$ -greedy policy based on the value of  $Q(s, a)$  in last step. In Line 6, state and action are updated.

The action-value function  $Q(s, a)$  is calculated by the following equation,

$$Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma Q(s', a') - Q(s, a)], \quad (7)$$

where  $r$  is the immediate reward, which is the profit when taken action  $a$  to reach state  $s'$ ,  $\alpha$  is the learning rate, and  $\gamma$  is a discount parameter. A smaller  $\gamma$  makes stronger emphasis on the current reward.

For state  $s$ , we refer to the work of Peters et al. [13], which has explored effective features in the retail market. With the selected features (and the corresponding presentations), Peter et al. has achieved competitive performance in the retail market, so we inherit these features in this work. The selected features for  $s$  and the corresponding feature representations are listed in Table 1. Actions are designed to effectively response to the dynamic market states [13]. Typical actions include increasing price, decreasing price, maintaining current price etc. The full list of action  $a$  is illustrated in Table 2.

As we are dealing with continuous control problems in the retail market, function approximation [16] is employed. We use a linear function approximation represented in Eq. (8).

$$Q(s, a) = \theta F(s, a)^T, \quad (8)$$

where  $F(s, a)$  is a feature vector, and  $\theta$  is continually updated by the reinforcement learner.

In above, one SARSA control process has been illustrated. For the three types of customers, three SARSA control processes, as illustrated above, are used independently. The three independent SARSA control processes work in parallel, taking the best strategy for each type of customers. We will demonstrate this is a more effective way to compete with other brokers in the experiment section.

## 5. Experiment

The proposed broker model is evaluated by the following four experiments from different perspectives. **Experiment 1** aims to test the performance of the proposed broker model competing with other brokers in Smart Grid markets. **Experiment 2** is to evaluate the on-line performance of the broker model on supply-demand balance. **Experiment 3** aims to evaluate the data-driven method for demand prediction and reveal the interactions between customer clustering and demand prediction. **Experiment 4** is to evaluate the performance of the strategy that our model takes in the retail market. For convenience, we call our implemented broker model **GongBroker**. We experiment GongBroker on the platform of Power TAC, which simulates real-world Smart Grid markets. Power TAC started in 2012 and held every year, and has drawn wide attentions throughout the world. Dozens of brokers have participated in the competition, making Power TAC an authentic platform to evaluate the performances of brokers. This is the reason for us to evaluate GongBroker on the platform of Power TAC.

The game process of Power TAC is briefly described as follows. Power TAC simulates the real-world 1 h with a 5 s timeslot, and it simulates 60 days in one game. At the beginning of the competition, each broker receives the bootstrap data with two weeks' length from the Power TAC server. The server also informs the properties of all consumers in the retail market. Besides, the server sends the hourly weather data with a length of two weeks. In each timeslot, brokers get public information including wholesale market clearing price and trading quantities, total energy production and consumption and weather forecasting data. In every 6 timeslots, brokers get the public information on tariffs in the retail market, including new tariffs, revoked tariffs and superseding tariffs. In each timeslot, each broker also gets the private information, including tariff transaction states of all his customers, the production and consumption of his current customers, the wholesale market transactions and the balancing transaction, and the current bank balance. Basing on the market information, brokers can make decisions on actions, such as bidding or asking in the wholesale market, and publishing or modifying tariffs in the retail market. When a game ends, brokers are evaluated by the amount of profit and the balance of supply and demand.

The Power TAC game is utilized to design our experiments. In Power TAC games, brokers compete in the markets to make more profits. This is a straight way to test the performances of broker models. Experiment 1 is taken in the game mode to compare the performance of GongBroker with others. Experiment 4 also makes use of the game mode to evaluate GongBroker's retail strategy. Besides, there are sufficient logs of historical market data supplied by Power TAC game server to analyze a broker model. The logs are used to analyze the performance of demand prediction module in GongBroker in experiment 2 and 3.

### 5.1. Experiment settings

Four experiments were setup to evaluate and analyze the proposed broker model. **1. Evaluation of competition with other brokers:** GongBroker was put into the Power TAC environment to compete with other excellent brokers. This experiment was a direct evaluation of the overall performance of GongBroker in Smart Grid mar-

**Table 3**

Common parameters for GongBroker in experiments.

Module	Parameter	Value
Customer demand prediction module	clustering criterion $\Delta_1$	0.15
	clustering criterion $\Delta_2$	0.05
	weight parameter $\lambda$	0.70
Wholesale market module	length of time $l$	5
Retail market module	learning rate $\alpha$	0.66
	discount parameter $\gamma$	0.71
	$\epsilon$ in $\epsilon$ -greedy	0.04

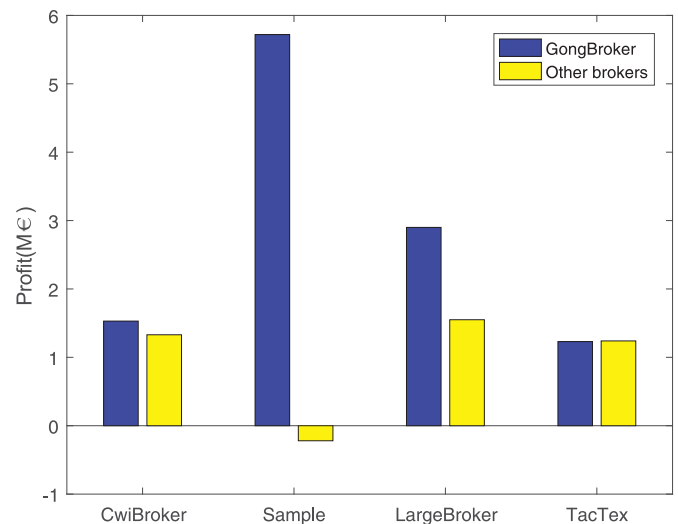
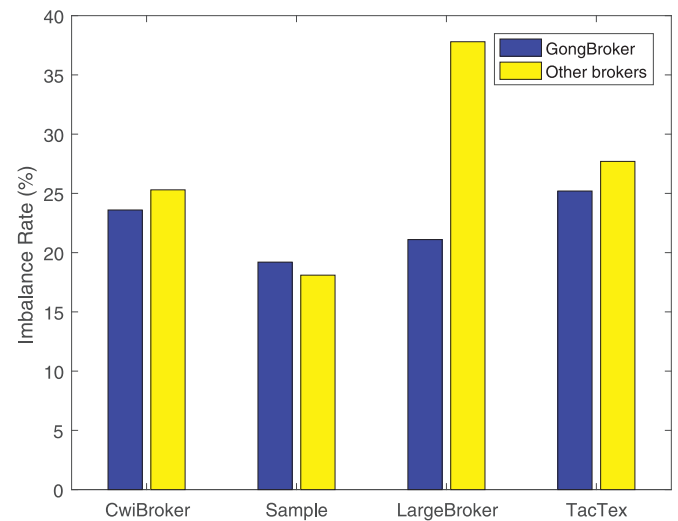
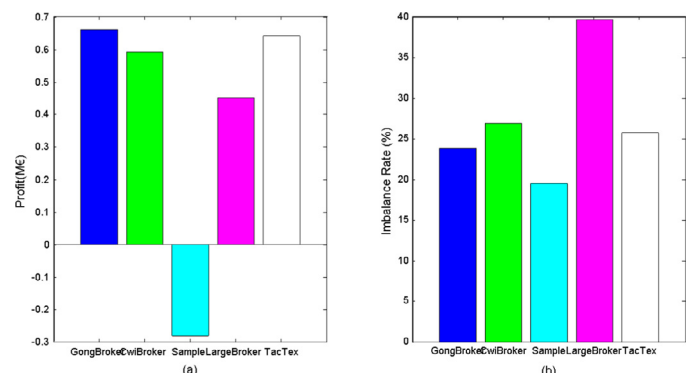
kets. **2. Evaluation of on-line supply-demand balance:** A new broker was implemented by replacing the customer demand prediction module by a baseline prediction method. Then we compared the new broker with GongBroker in Power TAC games. **3. Evaluation of off-line demand prediction:** We let GongBroker dominate the Power TAC markets. Different parameters in customer demand prediction module were set to evaluate the proposed data-driven method according to the logs of Power TAC server. **4. Evaluation of retail market control:** A new broker was constructed by replacing the retail market module by only one SARSA control process. Then we compared GongBroker with the new broker in Power TAC games. The common parameters for the four experiments are set in Table 3.

In the customer demand prediction module, we set the clustering criterion  $\Delta_1 = 0.15$  for the first layer clustering. For the second layer clustering, we set the clustering criterion  $\Delta_2 = 0.05$ . In this setting,  $M_1 = 11$  customer clusters were obtained in  $C_1$ , and  $M_2 = 12$  customer clusters were formed in  $C_2$ . To predict the usage amount for customers in  $C_1$ ,  $M_1 \times 24$  predictors were used, and the weight parameter  $\lambda$  in Eq. (1) was set as 0.70. In the prediction method for customers in  $C_2$ ,  $M_2 \times 24$  LASSO regressors were used. The above settings in customer demand prediction module maintained for experiment 1, 2 and 4. In experiment 3, the parameter  $\Delta_2$  was set to different value to evaluate the data-driven demand prediction method. In the wholesale market module, the length  $l$  was set to 5, which indicated if there was enough historical auction data. In the retail market module, the parameters for SARSA control was set as follows:  $\alpha = 0.66$ , and  $\gamma = 0.71$ . These settings were referred to [13]. For the  $\epsilon$ -greedy policy,  $\epsilon$  was set as 0.04. There are some random factors in the game, but the results are generally consistent. For a fair comparison, we run 10 games and report the average result. In the following subsections, the four experiments are reported and analyzed.

## 5.2. Experiment 1: evaluation of competition with other brokers

In this experiment, we introduced other four brokers<sup>3</sup> to compete with GongBroker in Power TAC games. The four brokers are Sample, LargeBroker, TacTex, and cwiBroker [1]. Sample broker was provided by the Power TAC, and the other three brokers were excellent brokers in former Power TAC competitions. We designed two different game modes to test the performance of GongBroker. In Mode 1, GongBroker competed with each of the other four brokers. Each competition was repeated 3 times to avoid the random factors. The averaged profits and imbalance rates of each broker are shown in Figs. 2 and 3, respectively. The imbalance rate here is defined as  $r = |U_a - U_p|/U_a$ , where  $U_a$  is the actual hourly usage and  $U_p$  is the predicted usage. In Mode 2, all the brokers joined

<sup>3</sup> There are some excellent broker models in [1]. We picked excellent brokers whose relevant literatures can be found, so that we could compare and analyze them. TacTex [22] was the champion broker in Power TAC 2013. CwiBroker [10] got the second places in Power TAC 2013 and 2014. LargeBroker was an excellent broker in Power TAC 2012, and its literature is [13]. Sample broker was a base line version supplied by Power TAC.

**Fig. 2.** Results of averaged profits of 5 brokers in Mode 1.**Fig. 3.** Results of averaged imbalance rates of 5 brokers in Mode 1.**Fig. 4.** Results of averaged profits and imbalance rates of 5 brokers in Mode 2.

the game. Three repeated games were held to avoid the randomness. The average profits and imbalance rates of each broker are illustrated in Fig. 4.

From Fig. 2, we can see that in Mode 1, when competing with other brokers, GongBroker made a profit of 1.53, 5.72, 2.90 and 1.23M€. The other four brokers, CwiBroker had a profit of 1.33M€, Sample broker made  $-0.22$ M€, LargeBroker got 1.55M€, and TacTex

gained 1.24M€. GongBroker made a competitive profit with TacTex, and gained larger profits than the other three brokers. In Fig. 3, the imbalance rates of GongBroker were 23.6%, 19.2%, 21.1%, 25.2%, respectively, while the imbalance rates of CwiBroker, Sample, LargeBroker and TacTex were 25.3%, 18.1%, 37.8% and 27.7%, respectively. GongBroker showed less imbalance rate against other three brokers but Sample. Sample broker kept a good imbalance rate but it owned only a few customers after a short period of the game. In summary, GongBroker shows the excellent performance in both profit making and supply-demand balancing in games in Mode 1.

In games in Mode 2, all the brokers participated in one game. From Fig. 4, we can see that GongBroker made a profit of 0.67M€ with an imbalance rate 23.9%. CwiBroker gained 0.59M€, and its imbalance rate was 26.8%. Sample broker made a profit of -0.27M€, with an imbalance rate 19.3%. LargeBroker had a 0.45M€ profit, and a 39.6% imbalance rate. TacTex gained 0.63M€, and had an imbalance rate of 25.6%. In Mode 2, GongBroker made the most profit and kept a good balance of supply and demand.

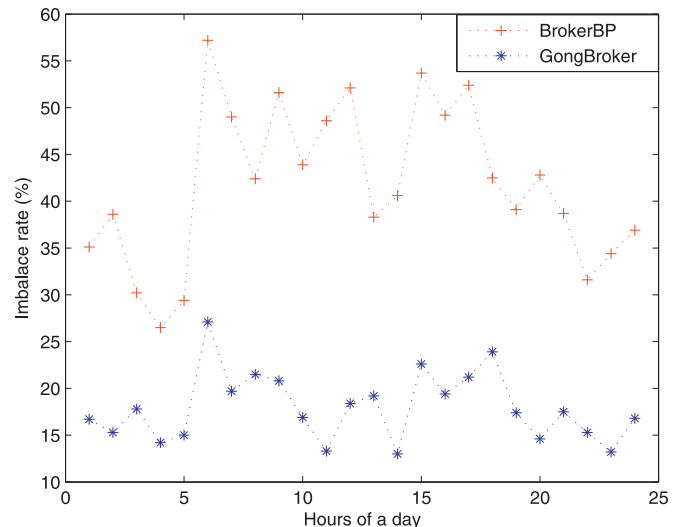
Combining the games in Mode 1 and Mode 2, we can safely draw the conclusion that GongBroker gains advantages in both profit making and supply-demand balance. The performance of the five brokers can be ranked in the following order: GongBroker, TacTex (the first two brokers show competitive performances), CwiBroker, LargeBroker and Sample broker. Moreover, some interesting empirical laws in Smart Grid markets can be drawn combining Mode 1 and Mode 2. One is that the profit margin shrinks when there are fierce competitions in markets. Games in Mode 2 exhibited stronger competitions than games in Mode 1. The total market profit in Mode 2 is 2.07M€, less than the total profit in any games in Mode 1. Taking a further analysis of the market profits in games in Mode 1, we can see that profit margins become smaller when the participated brokers are more competitive. For instance, in the game between GongBroker and TacTex, the total market profit is 2.47M€, while the market profit is 5.50M€ in the game between GongBroker and Sample. In a competitive environment, brokers publish their tariffs with low prices to attract customers from time to time, thus the profit margin shrinks. Another law is that the imbalance rate of supply demand increases when the market environment is more competitive. In Mode 2, the imbalance rates of all brokers slightly increased comparing to those of in games in Mode 1. In games in Mode 1, GongBroker showed a higher imbalance rate when competing with TacTex than the one when competing with Sample broker. In order to attract customers in the competitive retail market, brokers have to publish cheaper tariffs timely. In this situation, customers may migrate frequently among different brokers, which brings an extra difficulty in demand prediction. That is the reason why imbalance rate increases in a more competitive environment.

### 5.3. Experiment 2: evaluation of supply-demand balance

In this experiment, we evaluated the performance of GongBroker on supply-demand balance by a comparison broker. From this evaluation, the contribution of the customer demand prediction module can be demonstrated. The comparison broker retained the other modules of GongBroker, but replaced the customer demand prediction module with a baseline prediction method. The baseline prediction method was designed as follows. For each customer  $i$ , the next-day usage  $u_{it}$  ( $t = 1, \dots, 24$ ) was predicted the same as the hourly usage in current. Then by adding up the results for all the customers, the next-day energy demand  $U_t$ ,  $t = 1, \dots, 24$  was obtained. The modified broker model was called Broker with Baseline Predictions (BrokerBP). In the comparison experiment, the two brokers participated the game. Three games were repeated. The average results are shown in Table 4.

**Table 4**  
Performance comparison of GongBroker and BrokerBP.

Broker	Profit (M€)	Imbalance (%)
GongBroker	2.41	32.8
BrokerBP	0.76	81.4



**Fig. 5.** Results of imbalance rate of GongBroker and BrokerBP in a non-competitive environment.

As shown in Table 4, GongBroker made a profit of 2.41M€, while BrokerBP had a profit of 0.76M€. The imbalance rate of GongBroker was 32.8%, while imbalance rate of BrokerBP was 81.4%. GongBroker beat BrokerBP in both profit and imbalance rate. As GongBroker showed significant advantages over BrokerBP on balancing supply and demand, the contribution of the customer demand prediction module can be demonstrated. This result also reveals that an effective demand prediction method is critical for supply-demand balance. Moreover, though GongBroker and BrokerBP applied the same strategies to the retail market, GongBroker made much more profit than that of BrokerBP. The poor performance of BrokerBP in supply-demand balance results in a smaller profit, which demonstrates that supply-demand balance is also a critical factor for profit making.

To further analyze the customer demand prediction module, we run GongBroker and BrokerBP in two independent games. In each game, there was only one broker, GongBroker or BrokerBP, in Smart Grid markets. In this game setting, there was relatively no competition. Fig. 5 illustrates the hourly imbalance rate of GongBroker and BrokerBP in a non-competitive environment.

Fig. 5 shows that imbalance rates vary in 24 h. According to our commonsense, peak loads occur in 6–8 o'clock in the morning and 6–8 o'clock in the evening. It can be seen that the imbalance rates are higher in the peak hours than those of in other time periods. This phenomenon may attribute to that customers' behaviors are more chaotic in the peak hours. The total averaged imbalance rate of GongBroker is 19.1% and that of BrokerBP is 48.4%. In the environment with no competitions, imbalance rates are less than those in the competitive environment. This result reveals that strong competition brings an extra difficulty in supply-demand balance.

### 5.4. Evaluation of off-line demand prediction

This experiment evaluated the performance of the proposed data-driven method according to the off-line log data. The log data from the server recorded the actual hourly usages of all cus-

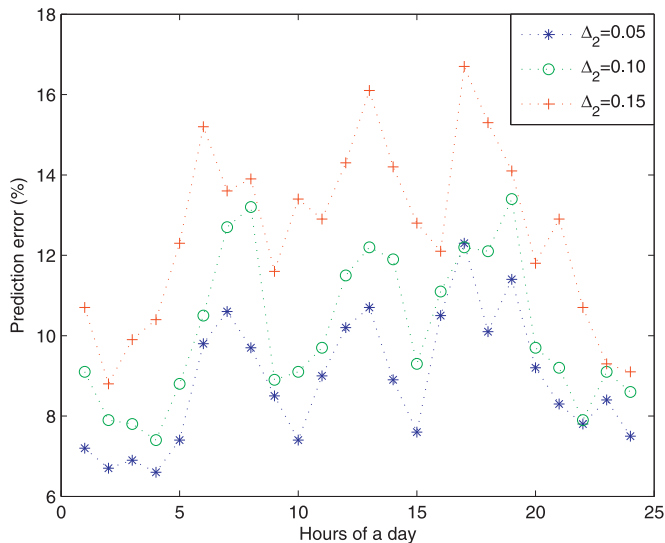


**Table 5**  
The influence of  $\Delta_2$  on the MAPE and the number of clusters.

$\Delta_2$	Cluster no. in $C_1$	Cluster no. in $C_2$	MAPE(%)
0.05	11	12	9.6
0.10	8	10	10.7
0.15	6	6	14.4

**Table 6**  
Performance comparison of GongBroker and BrokerOSC.

Broker	Profit (M€)	Imbalance (%)
GongBroker	2.43	27.3
BrokerOSC	1.70	24.7



**Fig. 6.** Results of prediction errors under different values of  $\Delta_2$ .

tomers, which could be regarded as the ground-truth of customer demand prediction. GongBroker was configured to dominate the whole Smart Grid markets, thus there was no competition in the market. Excluding all the other varying factors, we could analyze the robustness of the data-driven method. There are two parameters in the hierarchical customer clustering process, i.e.  $\Delta_1$  and  $\Delta_2$ . For  $\Delta_1$ , it can be determined by constraining the error between the prediction value and the ground-truth value. But for  $\Delta_2$ , it cannot be determined with ease.  $\Delta_2$  is a critical parameter which influences the granular of final customer clusters and affects the final prediction precision. Therefore, we analyze  $\Delta_2$  in details in the following.

We set  $\Delta_2$  to three different values: 0.05, 0.10, and 0.15. Apparently, the smaller value of  $\Delta_2$  leads to the finer granular of customer clusters. Table 5 summarizes how  $\Delta_2$  influences the number of clusters and the MAPE of 24 h. The prediction precision is measured by Mean Absolute Percentage Error (MAPE). Fig. 6 shows how MAPE of each hour changes with the value of  $\Delta_2$ .

In Fig. 6, when  $\Delta_2 = 0.05$ , the data-driven method shows the best performance. The prediction error increases when  $\Delta_2$  becomes larger. When  $\Delta_2$  increases from 0.10 to 0.15, the MAPE of each hour increases more. When  $\Delta_2$  ranges from 0.05 to 0.10, the prediction precision is acceptable. Table 5 shows how  $\Delta_2$  influences the MAPE of a day and the number of clusters. A larger  $\Delta_2$  results in a less number of customer clusters, thus less predictors are required. However, large value of  $\Delta_2$  causes the declination of prediction precision. On the contrary, a too small  $\Delta_2$  may result in too many customer clusters and the required predictors. The above analysis reveals that  $0.05 \leq \Delta_2 \leq 0.10$  is reasonable to ensure the performance of the data-driven demand prediction method. In practice, the value of  $\Delta_2$  can be compromised between the prediction precision and the number of predictors.

Some profiles of customer clusters are shown when  $\Delta_2 = 0.05$ . Customer clusters in  $C_1$  show stable power usages, and such clusters include cold storage company, householders with power storage and some office users. Cluster in  $C_2$  have unstable power us-

ages. These clusters are householders, office users and renewable energy producers.

#### 5.5. Experiment 4: evaluation of retail market control

This experiment evaluated the retail market control strategy of GongBroker by comparing GongBroker with a newly constructed broker. The comparison broker retained the other modules of GongBroker, but replaced the retail market module with only one SARSA control for all customers. The newly introduced SARSA control shares the same parameters with the three SARSA controls in GongBroker. The modified GongBroker is called Broker with One SARSA Control (BrokerOSC). We compared the performance of the two brokers in three games. The average results are shown in Table 6.

In Table 6, GongBroker made a profit of 2.43M€, while BrokerOSC had a profit of 1.70M€. The imbalance rate of GongBroker was 27.3%, and the imbalance rate of BrokerOSC was 24.7%. It can be seen that imbalance rates of the two brokers were very close, but GongBroker made 0.73M€ profit more than BrokerOSC. Therefore, the profit gap evidences that GongBroker uses better strategies than BrokerBP in the retail market. These comparison games demonstrate that it is an effective method to use independent SARSA controls for different types of customers in the retail market.

## 6. Discussions

We populated competitive market environments to test the performance of the proposed broker model. Through the competitions with other successful brokers, we demonstrated the advantages of the proposed broker model. The results of the games in two modes show that our broker model can make a leading profit and keep a good supply demand balance in different competitive market environments.

More importantly, through the experiments, two empirical laws can be drawn from the competitive market environments. **Law 1:** profit margin shrinks when there are fierce competitions in markets. In the competitive retail market, brokers offer cheap tariffs to attract customers, resulting in small profit margin. **Law 2:** the imbalance rate of supply demand increases when the market environment is more competitive. In the competitive environment, as brokers publish cheap tariffs from time to time, the customers migrate frequently between brokers. The frequent migrations of customers bring extra difficulties in demand prediction, resulting in a high imbalance rate.

A good supply-demand balance can enhance the energy using efficiency. A comparison experiment demonstrated the advantage of the proposed broker model in supply-demand balance. This experiment also verified the effectiveness of the proposed data-driven method for demand prediction. Besides, the comparison experiment showed that supply-demand balance was also a critical factor to gain a high profit for a broker. To further analyze the data-driven method for demand prediction, the key parameter  $\Delta_2$ , which affected the number of predictors and the prediction precision, was deeply analyzed. We have suggested a proper value range for  $\Delta_2$  in practice based on our evaluation results.

It is vital for a broker to keep and attract customers in the retail market. Reinforcement learning can be applied to generate adap-

tive tariff prices based on the changing market prices. Through a comparison experiment, we demonstrated that using independent SARSA controls for different customers was superior than using only one SARSA control for all customers. Independent SARSA controls introduce additional computational cost, but this disadvantage can be easily compensated by parallel computing.

## 7. Conclusions

This paper proposed an intelligent broker model for strategic power trading in Smart Grid markets. In proposed broker design, the challenges that brokers face in Smart Grid markets were comprehensively considered, and an adaptive and systematic model was constructed to surmount the challenges. The proposed broker model was tested and evaluated on the platform of Power TAC. The evaluation results showed the advantages of our broker against other excellent available brokers on both profit making and supply-demand balance. In particular, four experiments were conducted from different perspectives to explore the effectiveness of our methods, used in each module in our broker design. The experimental results demonstrated that our data-driven method for demand prediction was effective in the complex and dynamic market environment due to its consideration of various types of customers and their behaviors. Moreover, the online reinforcement learning method used in the retail market could help the broker to adapt to the changing market environment. The experimental results have also verified that learning various strategies for different types of customers is more competitive than learning towards the whole retail market. Through the experiments, two empirical laws have been discovered i.e. Law 1: profit margin shrinks in a competitive market environment and Law 2: the imbalance rate of supply demand increases when a market environment is more competitive.

Research on broker model for Smart Grid markets is an emerging field. We have made effort to build an autonomous broker model that could take adaptive strategies to cope with the dynamics in the complex Smart Grid environments. We wish that our attempts might enlighten the autonomous broker designs in Smart Grid markets, even though there are some issues worth further explorations.

## Acknowledgment

The authors would like to thank the organizers of Power TAC, which supplies a simulation of real-world smart market. The experiments are done based on that platform. This work is supported by a Discovery Project (DP140100974) from Australian Research Council.

## References

- [1] <http://www.powertac.org/wiki/index.php/form:broker>
- [2] A. G. Barto, Reinforcement Learning: An Introduction, MIT press, 1998.
- [3] N. Amjady, K. F. H. Zareipour, Short-term load forecast of microgrids by a new bilevel prediction strategy, *IEEE Trans. Smart Grid* 1 (3) (2010) 286–294.
- [4] T. Chow, C. Leung, Neural network based short-term load forecasting using weather compensation, *IEEE Trans. Power Syst.* 11 (4) (1996) 1736–1742.
- [5] A. Conejo, J. Morales, L. Baringo, Real-time demand response model, *IEEE Trans. Smart Grid* 1 (3) (2010) 236–242.
- [6] J. Fan, J. McDonald, A real-time implementation of short-term load forecasting for distribution power systems, *IEEE Trans. Power Syst.* 9 (2) (1994) 988–994.
- [7] T. Kanungo, D.M. Mount, N.S. Netanyahu, C.D. Piatko, R. Silverman, A.Y. Wu, An efficient k-means clustering algorithm: analysis and implementation, *Pattern Anal. Mach. Intell.* *IEEE Trans.* 24 (7) (2002) 881–892.
- [8] W. Ketter, J. Collins, P. Reddy, M. Weerd, The 2014 Power Trading Agent Competition, ERIM Report Series Reference No. ERS-2014-004-LIS, 2014.
- [9] R. Kuate, M. He, M. Chli, H. Wang, An intelligent broker agent for energy trading: An mdp approach, in: Proceedings of the Twenty-Third international joint conference on Artificial Intelligence, AAAI Press, 2013, pp. 234–240.
- [10] B. Liefers, J. Hoogland, P. La, A successful broker agent for power tac, in: Agent-Mediated Electronic Commerce. Designing Trading Strategies and Mechanisms for Electronic Markets, Springer, 2014, pp. 99–113.
- [11] A. Mohsenian, A. Leon, Optimal residential load control with price prediction in real-time electricity pricing environments, *IEEE Trans. Smart Grid* 1 (2) (2010) 120–133.
- [12] A. Motamedi, H. Zareipour, W. Rosehart, Electricity price and demand forecasting in smart grids, *IEEE Trans. Smart Grid* 3 (2) (2012) 664–674.
- [13] M. Peters, W. Ketter, M. Saar-Tsechansky, J. Collins, A reinforcement learning approach to autonomous decision-making in smart electricity markets, *Mach. Learn.* 92 (1) (2013) 5–39.
- [14] M. Puterman, Markov Decision Processes: Discrete Stochastic Dynamic Programming, John Wiley & Sons, 2014.
- [15] M. Roozbehani, M. Dahleh, S. Mitter, Dynamic pricing and stabilization of supply and demand in modern electric power grids, in: Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on, IEEE, 2010, pp. 543–548.
- [16] G. Rummery, M. Niranjan, On-Line Q-Learning Using Connectionist Systems, CUED/F-INFENG/TR 166, Cambridge University Engineering Department, 1994.
- [17] S. Saab, E. Badr, G. Nasr, Univariate modeling and forecasting of energy consumption: the case of electricity in Lebanon, *Energy* 26 (1) (2001) 1–14.
- [18] D. Srinivasan, Energy demand prediction using gmdh networks, *Neurocomputing* 72 (1) (2008) 625–629.
- [19] L. Suganthi, A. Samuel, Energy models for demand forecasting—a review, *Renewable Sustainable Energy Rev.* 16 (2) (2012) 1223–1240.
- [20] G. Tesauro, J. Bredin, Strategic sequential bidding in auctions using dynamic programming, in: Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 2, ACM, 2002, pp. 591–598.
- [21] R. Tibshirani, Regression shrinkage and selection via the lasso, *J. R. Stat. Soc. Ser. B (Methodological)* 58 (1996) 267–288.
- [22] D. Urieli, P. Stone, Tactex'13: a champion adaptive power trading agent, in: Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems, 2014, pp. 1447–1448.
- [23] J. Vilar, R. Cao, G. Aneiros, Forecasting next-day electricity demand and price using nonparametric functional methods, *Int. J. Electr. Power Energy Syst.* 39 (1) (2012) 48–55.
- [24] P. Vytelingum, D. Cliff, N. Jennings, Strategic bidding in continuous double auctions, *Artif. Intell.* 172 (14) (2008) 1700–1729.