Bidding Strategy for Two-Sided Electricity Markets: A Reinforcement Learning based Framework

Bala Suraj Pedasingu
TCS Research
Tata Consultancy Services, India
balasuraj.p@tcs.com

Easwar Subramanian
TCS Research
Tata Consultancy Services, India
easwar.subramanian@tcs.com

Yogesh Bichpuriya
TCS Research
Tata Consultancy Services, India
yogesh.bichpuriya@tcs.com

Venkatesh Sarangan
TCS Research
Tata Consultancy Services, India
venkatesh.sarangan@tcs.com

Nidhisha Mahilong
TCS Research
Tata Consultancy Services, India
nidhisha.m@tcs.com

ABSTRACT

We aim to increase the revenue or reduce the purchase cost of a given market participant in a double-sided, day-ahead, wholesale electricity market serving a smart city. Using an operations research based market clearing mechanism and attention based time series forecaster as sub-modules, we build a holistic interactive system. Through this system, we discover better bidding strategies for a market participant using reinforcement learning (RL). We relax several assumptions made in existing literature in order to make the problem setting more relevant to real life. Our Markov Decision Process (MDP) formulation enables us to tackle action space explosion and also compute optimal actions across time-steps in parallel. Our RL framework is generic enough to be used by either a generator or a consumer participating in the electricity market.

We study the efficacy of the proposed RL based bidding framework from the perspective of a generator as well as a buyer on real world day-ahead electricity market data obtained from the European Power Exchange (EPEX). We compare the performance of our RL based bidding framework against three baselines: (a) an ideal but un-realizable bidding strategy; (b) a realizable approximate version of the ideal strategy; and (c) historical performance as found from the logs. Under both perspectives, we find that our RL based framework is more closer to the ideal strategy than other baselines. Further, the RL based framework improves the average daily revenue of the generator by nearly €7,200 (€2.64 M per year) and €9,000 (€3.28 M per year) over the realizable ideal and historical strategies respectively. When used on behalf of a buyer, it reduces average daily procurement cost by nearly €2,700 (€0.97 M per year) and €7,200 (€2.63 M per year) over the realizable ideal and historical strategies respectively. We also observe that our RL based framework automatically adapts its actions to changes in the market power of the participant.

BuildSys '20, November 18–20, 2020, Yokohama, Japan
© 2020 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-8061-4/20/11…$15.00
https://doi.org/10.1145/3408308.3427976

CCS CONCEPTS

• **Applied computing → Operations research**; • **Computing methodologies → Sequential decision making**; **Reinforcement learning**.

KEYWORDS

Electricity Markets, Optimization, Bidding, Forecasting, Reinforcement Learning.

ACM Reference Format:
Bala Suraj Pedasingu, Easwar Subramanian, Yogesh Bichpuriya, Venkatesh Sarangan, and Nidhisha Mahilong. 2020. Bidding Strategy for Two-Sided Electricity Markets: A Reinforcement Learning based Framework. In *The 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys '20), November 18–20, 2020, Virtual Event, Japan.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3408308.3427976

1 INTRODUCTION

Background: Smart grids are an essential part of smart cities. They manage a smart city's electricity demand in a sustainable, reliable and economical way using sensors, smart controls, and renewable energy sources [9, 19]. Power generating companies and electricity distribution utilities are integral participants of a smart grid. Generators are the suppliers that produce and sell power to intermediaries or distributing agencies through auctions in an open electricity market. Auctions offer better price discovery for both buyers and sellers and hence participants get better value for the electricity traded [27]. More importantly, through auctions, electricity markets accommodate demand flexibility and pave way for efficient utilization of distributed renewable generation [14].

There are different kinds of electricity markets such as day-ahead, intra-day, and balancing. Our work's focus is on double-sided, day-ahead, wholesale electricity markets. In these markets, buyers and sellers place their bids and asks with the market operator one day before the actual delivery day. Bids and asks are placed for all time blocks pertaining to the delivery day. Multiple auctions happen for the same delivery day with one auction for each time block. Further, the buyer or seller can place multiple bids or asks per auction. In addition, the total volume of bids (i.e., available generation capacity) and the total volume of asks (i.e., smart city load that has to be met) can vary across auctions pertaining to a delivery day. The market operator matches the bids with the asks using a prescribed market clearing mechanism. The results of auction clearing is advertised to all participants. The participants should honor their cleared commitments on the delivery day.

Scope of work: We consider the problem of optimizing the revenue or purchase cost of a market player who participates in the periodic double auctions of a day-ahead wholesale electricity market [16]. The market player could either be a supplier or consumer. Specifically, we advocate the use of a reinforcement learning (RL) [21] based bidding framework for participation in double auctions. The proposed bidding framework has three sub-systems. The first sub-system is a time-series forecaster based on the attention networks [24]. These networks are specifically trained to forecast the future values of different market state variables such as market demand, supply, clearing prices and clearing quantities. The second sub-system is a model that mimics the clearing mechanism of a two-sided electricity market. The third and the main sub-system considers the current market conditions to design the bids (or asks) using approximate Q-learning techniques [26]. The time-series predictor and the market clearing simulator are leveraged by the bid/ask designer to discover better bidding strategies on behalf the specified market participant. In this paper, we focus on simple bids (asks), wherein, the bids (asks) submitted for one time block are *independent* of the bids (asks) submitted for other time blocks [18].

Gaps: Several studies in existing literature offer efficient bidding mechanisms for participants of a double auction in wholesale energy markets [2, 4, 5, 22, 28, 29]. Some of these works even deploy learning framework to arrive at optimal bidding strategies [2, 4, 22, 29]. There are some that take a game theoretic approach and analyze Nash equilibria of the resultant auctions [4, 13, 25]. However, much of these previous works make simplistic assumptions about the nature and mechanism of the auction process. Some such assumptions include, (a) considering auctions which are just a time-step away; (b) the auctions are capable of accepting only one bid per auction, when in reality multiple bids can be submitted; (c) the bidding strategy assumes a constant generation capacity across time. The prime motivation of our work is to propose a learning framework for a bidding agent by relaxing the above assumptions to make the framework more suitable for real-world markets.

Contributions: Specific contributions of our work are as follows: (i) We propose a RL based approach for a market player to optimally participate in two-sided day-ahead wholesale electricity markets. The framework is generic enough to be used by either an electricity buyer or seller; (ii) We adopt attention based deep learning techniques that are traditionally used for predicting natural language sequences for forecasting time series data pertaining to future market states (*viz.* demand, supply and prices). (iii) We describe an operations research based model to mimic the market clearing mechanism of a two-sided electricity market. This model creates an interaction environment for the learning framework to discover efficient market participation strategies.

We measure the efficiency of our RL based bidding framework in terms of the total cost or revenue generated over all auctions pertaining to a delivery day. While buyers need to minimize their procurement costs, sellers need to maximize their revenues. We test the performance of the proposed RL framework using real-world double auction data logs obtained from day-ahead European Power Exchange (EPEX) market. Specifically, we train RL agents on behalf of suppliers (buyers) with varying degree of market dominance and demonstrate the efficacy of the trained agents against several baselines using accumulated daily revenues (costs).

Paper outline: The outline of the paper is as follows. We begin by discussing the relationship of our problem setting with past works in Section 2. Section 3 provides a description of our system set up and an overview of different components of our bidding framework. Section 4 discusses the design of reinforcement-learning based bid/ask designer. Section 5 explains our forecasting model and Section 6 describes our market clearing model in detail. In Section 7, we outline our experimental setup and illustrate the efficacy of the bidding agent by comparing its performance with several baselines. In Section 8, we discuss extensions to the proposed framework. Finally, in Section 9, we present our conclusions.

2 RELATED WORK

Existing literature on optimizing the buy/sell strategy in wholesale electricity markets can be classified under two categories: (a) research where the market participant is a price taker [7, 8, 11] and (b) research where the market participant is a price maker. If the market participant is a price taker, the underlying assumption is that the buy/sell bids placed by the participant cannot influence the market clearing price. On the other hand, if the market participant is a price maker, then the buy/sell bids placed by the participant can affect the market clearing price. In our work, we assume that the market participant is big enough to be a price maker. Hence, we focus on research under this category.

Attarha et. al.[1] have proposed a bidding strategy for generators (including storage devices and distributed energy resources) participating simultaneously in one-sided day-ahead and frequency markets. They consider the ramping up/down constraints associated with the generator and use an optimization framework to determine the optimal offers to be placed for the next time slot. Our work is in the context of two-sided markets and uses an RL based approach. An one-sided market is a special case of a two-sided market wherein all the buyers have their bid price set to infinity. Therefore, our work can be considered to be done in a more generic setting. Further, we determine the bids/asks for all the time slots in the delivery day as opposed to only one time slot. Also, the authors of [1] do not model the market clearing process explicitly which we do in our work. Subramanian et. al [20] propose a RL based bidding strategy for generators participating in one-sided day-ahead markets. They determine the asks for all the time slots in the delivery day. In our work, we focus on two sided markets, which is more generic than one-sided markets. Also our proposed framework can not only be used by generators but also by consumers.

Shafiee et. al [17] propose an approach for determining the asks and bids placed by a energy storage operator in a two-sided market. They propose a robust optimization based technique that is cognizant of the uncertainties in forecasting the aggregate market supply and demand curves. However, their approach assumes that the supply and demand curves have a finite number of steps which is known apriori to the storage operator which may not be possible.

Further, their computational complexity increases with the number of steps in the supply and demand curves. This inhibits their real-world application to markets where the supply and demand curves can be quite smooth with arbitrary number of small steps.

There are also other works that investigate the use of RL for bidding in electricity markets. Naghibi et. al [13] show that when two generators use RL to place asks in a power pool market, their asks ultimately converge to a Nash equilibrium. Krause et. al [10] show that when all the generators use RL with Q-learning for placing their asks in a day ahead market, the generator asks can oscillate between different Nash equilibria (if multiple equilibria exist). Another work by Wang et. al [25] shows that when multiple Nash equilibria exist, a set of cooperative generator RL agents may not be able to learn any coordination strategy. While these works are very useful in their own way, they cannot be applied in real world electricity markets due to their simplistic assumptions. Such assumptions include: (i) placing an ask for only the next time step as opposed to multiple time steps; (ii) an ask can be submitted with only one price as opposed to many price bands; (iii) the market demand and generator output remaining constant across time as opposed to being dynamic; to name a few. Our proposed work, on the other hand, does not have these assumptions and hence is more closer to real world settings.

3 SYSTEM SETUP

We begin by introducing the system setting, notations and then provide an overview of different components of our framework.

Electricity Market Ecosystem: We depict a simplistic two-sided, day-ahead electricity market ecosystem in Figure 1. The buyers and sellers submit their bids and asks for each auction pertaining to the next delivery day. For example, there could be twenty-four auctions in a delivery day corresponding to each of the 24 hours – the buyers and sellers would then place their 24 bids and asks simultaneously, one for each hour. The market participants could belong to different geographical regions. There are inter-regional transmission lines that connect these regions and transfer electricity between them. Typically, buyers and sellers of an electricity market, place their bids and asks within their corresponding regions. The market operator matches the buy bids with the sell asks. This matching determines the market clearing price and quantity for all auctions of the delivery day for each market participant. If need be, the market operator can also communicate with the transmission system operator to understand the network constraints before clearing the bids and asks. The clearing price and the respective cleared quantities are communicated to all the participating buyers and sellers. The buyers and sellers then prepare their consumption and generation accordingly.

Notations: Denote $\mathcal{B}$ and $\mathcal{S}$ as the set of buyers and sellers respectively. We define the set of bids placed by a buyer $b \in \mathcal{B}$ at time n for a future time $n+h$ as $\Psi_{b,n+h} = \{< p_{b,n+h}, q_{b,n+h} >\}$. The future time $n+h$ is one of the auction times on a delivery day. The tuple $< p, q >$ is the price-quantity pair of a bid. Similarly, let $\Phi_{s,n+h} = \{< p_{s,n+h}, q_{s,n+h} >\}$ denote the asks that a seller $s \in \mathcal{S}$ placed in market at time n for a future time $n+h$. Note that our definition of $\Psi_{b,n+h}$ and $\Phi_{s,n+h}$ allows for more than one bid/ask to be placed in an auction. In general, there could be $m \in \mathbb{N}$ bids (or asks) per auction.

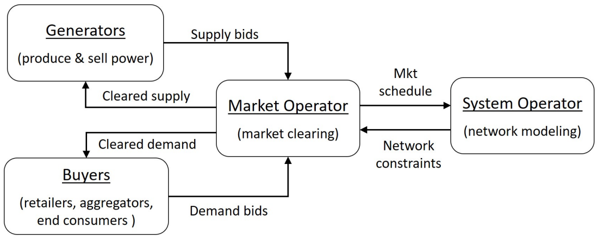


Figure 1: An electricity market ecosystem consisting of buyers, sellers, market operator & system operator.

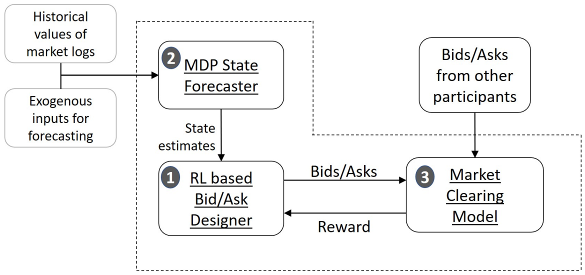

Historical values of market logs
Exogenous inputs for forecasting
2 MDP State Forecaster
State estimates
Bids/Asks from other participants
1 RL based Bid/Ask Designer
Bids/Asks
Reward
3 Market Clearing Model

Figure 2: Overview of bidding framework

All buyers and sellers submit their bids and asks simultaneously for all time blocks of a delivery day. For each time block of the delivery day, the market operator then matches the bids with the asks through a predefined clearing methodology. The clearing process discovers a market clearing price p^*_{n+h} and total cleared volume q^*_{n+h} for each auction time $n+h$ of the delivery day. All asks submitted by a supplier $s \in \mathcal{S}$ with $p_{s,n+h} < p^*_{n+h}$ and all bids of the buyer with $p_{b,n+h} > p^*_{n+h}$ are cleared. Bids and asks with price equal to clearing price p^*_{n+h} are partially cleared. The supply commitment for a supplier $s \in \mathcal{S}$ for time $n+h$, denoted by, $C_{s,n+h}$ is calculated as

$$C_{s,n+h} = \sum \mathbb{1}_{p_{s,n+h} < p^*_{n+h}} \left[q_{s,n+h}\right] + c_{n+h} \tag{1}$$

where the summation is over all the asks submitted by seller s for delivery slot $n+h$. Similarly, the commitment, $D_{b,n+h}$ for a bidder $b \in \mathcal{B}$ at a time $n+h$ is computed as

$$D_{b,n+h} = \sum \mathbb{1}_{p_{b,n+h} > p^*_{n+h}} \left[q_{b,n+h}\right] + d_{n+h} \tag{2}$$

The values of c_{n+h} and d_{n+h} are determined, by the market operator, from the asks (bids) submitted for time $n+h$ at the clearing price. These are referred to as marginal clearings. The clearing mechanism ensures that,

$$\sum_{s \in \mathcal{S}} C_{s,n+h} = \sum_{b \in \mathcal{B}} D_{b,n+h} = q^*_{n+h}. \tag{3}$$

Overview of Bidding Framework: Our RL based electricity bidding framework, as shown in Figure 2 consists of three main modules: (i) a bid/ask designer, (ii) a forecasting module, and (iii) a market clearing module.

At the core of framework is the bid/ask designer, which as the name suggests, is responsible for generating bids or asks on behalf of a particular buyer or seller, respectively. This module learns an optimal strategy to generate bids/asks using reinforcement learning (RL) [21]. Specifically, we formulate a suitable Markov decision process (MDP) [15] to model the bid/ask generation problem and solve the MDP approximately using Q-learning [26]. Since all the state variables of our MDP are not directly observable, we devise a mechanism to estimate those state variables of the MDP via a forecasting module which is attention based RNN time-series predictor. These forecasted values are fed to the bid/ask designer that solves the MDP. The market clearing module is a part of the external environment with which the bid/ask designer interacts. This module mimics the clearing mechanism of the two sided auctions in day-ahead electricity markets. In addition to the bids (asks) received from the bid/ask designer, the market clearing module also obtains inputs from other buyers and sellers in the electricity market to simulate the market clearing process. While the behavior of other participants can also be modeled, in this paper, we use data feeds pertaining to these participants from available historical logs. The next three sections describe each of the three modules in detail.

4 BID/ASK DESIGNER

We use reinforcement learning to train the bid/ask designer to place price and quantity bids (asks) in the market on behalf of a buyer (seller). The bids (asks) are determined so as to minimize (maximize) the procurement cost (revenue) accumulated over a performance horizon of one day. We model this module's interaction with the market as a Markov decision process (MDP) [15].

Synopsis: At each time block n, the bid-ask designer observes a state $\mathbf{s}_n$ and suggests an action a_n. The action a_n constitutes the set of bids $\Psi_{b,n+h}$ (or asks $\Phi_{s,n+h}$), where $n+h$ is a time block of the delivery day. Recall that we allow for more than one bid (or ask) to be placed by a participant. In response to the bids and asks placed for time block $n+h$, the market clearing model returns the market clearing price p^*_{n+h} and aggregated market cleared quantity q^*_{n+h} for the auction (Section 6 describes how to determine p^*_{n+h} and q^*_{n+h}). Then, as described in Equations (1) and (2), the quantity cleared by the market for a specific buyer b and seller s can be determined. The result of the clearing mechanism is then used to compute a reward $r_n = \mathcal{R}(\mathbf{s}_n, a_n)$ (the reward function is discussed later in this section). The objective of RL is then to find a bidding strategy that maximizes the discounted sum of rewards over a horizon of one trading day.

State Space: The state variables that are made available to the bid/ask designer at time block $n \in N$ to place bids for time block $n+1$ are as follows: (i) the capacity (consumption or generation) of a given participant at $n+1$ ($\mathcal{U}_{n+1}$); (ii) the expected total market supply ($\tilde{G}_{n+1}$) and demand ($\tilde{D}_{n+1}$) at $n+1$; (iii) the expected market clearing price and quantity at $n+1$; (iv) the actual capacity (consumption or generation) of the participant that was available at $n-23$ and $n-167$; (v) the actual total market supply and demand at $n-23$ and $n-167$; (vi) the actual market clearing price and quantity at $n-23$ and $n-167$.

We note that the state variables corresponding to instants $n-23$ and $n-167$ are those that precede the auction time by exactly a day and week respectively. Out of the state variables listed above, only the values are instants $n-23$ and $n-167$ can be observed accurately (since they are historical). The total capacity (consumption or generation) of a given participant available to place a bid/ask at $n+1$ and other future time slots can be assumed to be observable – since this corresponds to the estimate of the participant's *own internal* future state. Apart from $\mathcal{U}_{n+1}$, other state variables at $n+1$ are un-observable since these correspond to the future state of the environment.

Since we are focusing on day ahead markets, at a given time slot n, the bid/ask designer should place bids/asks for slots $n+1$ to $n+H$ as a batch, where H refers to the number of time slots in the delivery day. We use the MDP state forecaster module (discussed in Section 5) to estimate all the un-observable state variables from $n+1$ until $n+H$.

Action Space: The action of the bid/ask designer constitutes placing m bids (asks) for future time blocks $n+h$, $h \in \{1, \cdots, H\}$ based on the state information at time n. For a market participant e, each of its m bids (asks) is of the form $< p_{e,n+h}, q_{e,n+h} >$, $e \in \{b, s\}$. The bid/ask generator designs the m possible bids (asks) for a delivery slot $n+h$ in the following way. We take m equally spaced price values from the interval $[0, p_{\max}]$ where $p_{\max}$ is the maximum possible limit price. The value of $p_{\max}$, which is a hyper-parameter, can be estimated using historical market clearing prices. Next, we distribute the available generation or needed consumption for delivery at time slot $n+h$ among the m price bands using nine possible *distribution profiles*. These distribution profiles, shown in Figure 3, are obtained using a family of chi-squared and Gaussian distributions. The first four profiles are skewed to the lower price bands. From a supplier's perspective, these are more conservative actions, since these actions tend to push the market clearing price lower. The last four are skewed to the higher price bands and hence are more aggressive from the point-of-view of a supplier. The fifth profile is a Gaussian that allocates much of the quantity to the price bands at the center. The labeling of aggressive and conservative actions will get reversed when viewed from the perspective of a buyer. Our action space design of having to choose between nine distribution profiles is aimed at achieving computational tractability while solving the MDP. The action space can be further enriched by having more generic profiles, e.g., multi-modal distributions.

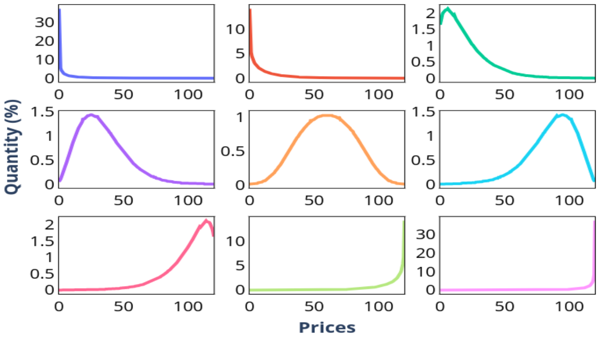


Figure 3: MDP Action Space Design : Distribution profiles that spreads the participant's capacity across m price bands.

State transitions: The underlying state evolution of the MDP depends on (i) market supply or demand transitions, and (ii) participant specific generation or consumption profiles.

Reward Formulation: For each auction pertaining to time-block $n+h$, the reward is the revenue generated or the cost incurred by the participant as a result of the ask/bid placed by it. The reward function for a supplier s is defined as follows:

$$r_{n+h} \quad = \quad \beta_1 \times p^*_{n+h} \times C_{s,n+h} \;\; + \;\; \beta_2 \times p^*_{n+h} \times U_{s,n+h} \qquad (4)$$

For a buyer b, the reward function is defined as

$$r_{n+h} \quad = \quad \beta_1 \times p^*_{n+h} \times D_{b,n+h} \;\; + \;\; \beta_2 \times p^*_{n+h} \times U_{b,n+h} \qquad (5)$$

In Equations 4 and 5, β_1 and β_2 are hyper-parameters which can be tuned as per the specific objective of the supplier or buyer. The first term involving β_1 refers to the actual revenue or cost that results from the market participation. The second term involving β_2 is a penalty levied for the quantum of generation that is left unsold ($U_{s,n+h}$) or demand that is left unmet ($U_{b,n+h}$).

The objective of the RL based bid/ask designer is to place a sequence of bids (asks) that maximizes the objective function

$$J(\cdot) = \sum_{h=1}^{H} \mathbb{E}\left(\gamma^h r_{n+h} | s_n\right)$$

Here γ is a discount factor of the MDP, s_n is given state at time block n and H is the number of auctions in a trading day. Since RL is cast as a reward maximization problem, for a supplier, typically $\beta_1 > 0$ and $\beta_2 \leq 0$. In the case of a buyer, $\beta_1 < 0$ and $\beta_2 < 0$. If the participant is interested in optimizing the profit, then the reward function can be modified by deducting the cost associated with action a_n.

Our state and action space design does not consider the ramping constraints of a market participant. This is not an unreasonable assumption, as real world entities such as a gas turbine supplier, have very high ramping rates due to which ramping constraints can be overlooked. However, we do discuss in Section 8, a more generic state and action space formulation that models ramping constraints.

Training Methodology The MDP formulation presented above can be solved using Q-learning [26]. Note that the state space of the above MDP is continuous and the action set is discrete. Hence, to approximately solve this MDP, we deploy the Deep-Q-Network (DQN) algorithm [12]. At the core of the DQN algorithm, lies a neural network that approximates the optimal state-action value function $Q^*(s, a)$. The state-action value function is defined to be the total expected sum of discounted rewards obtained in state s by taking a and then subsequently applying optimal actions until the end of the horizon. The optimal state-value function is the one which yields maximum value compared to all other action-value functions. Solving an MDP is akin to finding the optimal state-action value function. The DQN algorithm aims to learn an approximation $Q^*_\theta(s, a)$ to the optimal action-value function $Q^*(s, a)$ where θ is a vector that represents the weights of a neural net.

While training, the agent observes a state s at time n to place bids (asks) for a delivery time slot $n + h, h \in \{1, \cdots, H\}$. It selects one of nine distribution profiles as action a. Once a profile is chosen, m bids (asks) gets placed for time slot $n + h$. Note that some state variables like expected market demand, market supply for $n + h$ needs to be predicted which are taken care by the MDP state forecaster sub-module. The bids or asks of other participants are chosen from historical values[1]. The auction is then cleared by our market clearing model which outputs the clearing price, cleared quantity and a next state s'. The market outputs are used to derive the reward for choosing action a in state s. The quadruple (s, a, r, s') is an experience and several such experiences are used to update the parameter θ of the neural net according the DQN algorithm. The pseudo code of the DQN algorithm for training the network is given in Algorithm 1.

```
Q-learning using DQN : Learn Action Value Function $Q$

Input: $s_n$ = {State variables enumerated in Section 4} ;
Input: A = {Discrete action space consisting of 9 profiles } ;
Input: Neural network $Q_\theta$ with weights $\theta = \theta_{\text{initial}}$
Input: Target network $\widetilde{Q}_{\tilde{\theta}}$ with weights $\tilde{\theta} = \theta_{\text{initial}}$ ;
Input: Discount factor $\gamma$
Input: Replay buffer capacity $R$
Input: Batch size $B$
Input: Episode length $T$
Input: Number of episodes $M$
Input: Exploration parameter $\varepsilon$
Input: $\varepsilon$-decay size $\Delta\varepsilon$
Input: Hyper-parameter $C$;
Output: Neural network $Q_\theta$ with learnt weights $\theta = \theta_{\text{final}}$
For episodes = $1 \cdots M$ do ;
    For time-steps $n = 1 \cdots H$ do ;
        Fetch state $s_n$
        With probability $\varepsilon$ select random action $a_n$ or ;
        select $a_n = \arg\max_a Q_\theta(s_n, a)$
        Execute action $a_n$; Observe reward $r_n$ and next state $s_{n+1}$;

        Store transition $(s_n, a_n, r_n, s_{n+1})$ in buffer $R$
        Sample $B$ random mini batch of transitions from $R$ ;
        For each transition $(s_j, a_j, r_j, s_{j+1})$ in $B$, calculate target
            $y_j = \begin{cases} r_j, \text{ if } s_{j+1} \text{ is terminal state} \\ r_j + \gamma \max_a \widetilde{Q}_{\tilde{\theta}}(s_j, a_j), \text{ otherwise} \end{cases}$
        Perform gradient descent on $\sum_{j=1}^{B} \left(y_j - \widetilde{Q}_{\tilde{\theta}}(s_j, a_j)\right)^2$
        Every C steps set $\tilde{\theta} = \theta$
    End For
    $\varepsilon \leftarrow \max\{0.01, \varepsilon - \Delta\varepsilon\}$.
End For
```

Algorithm 1: Pseudocode for DQN algorithm

5 MDP STATE FORECASTER: RCAN

In this section, we describe our deep neural network architecture, depicted in Figure 4, called Regression Coupled Attention Network (RCAN) to forecast the future state variables of the MDP for time steps $n+1$ through $n+H$ using information at time step n. Recall that the state variables include market supply, market demand, market clearing price and market clearing quantity. The network contains

[1] Note that the RL training episode length is one day. The bids/asks for a single delivery day are placed simultaneously by all the market participants. Therefore, historical values of bids/asks placed by other market participants for any single day can be used for training as they are not influenced by the RL agent's bids/asks for the same day.

a latent embedding layer that produces a latent representation of the input data, which is periodically sliced and concatenated to feed into multiple regression coupled attention blocks (RCA). Then, the output of final RCA block is fed into feed forward network to produce the desired forecast. We adopt a multi-head attention mechanism, popularly used in natural language synthesis [24]. The components of the our RCAN architecture are described below.

Let $\mathbf{x} = (x_{t-B}, \cdots, x_t)$ denote an input time sequence of length B with each $x_{t-k} \in \mathbb{R}^d, k \in \{0, \cdots, B\}$, with d being the input dimension. The primary objective of a forecasting model is predicting H future points of a given time series. In simple terms, a neural network characterized by a non-linear function $f^{\text{net}}(\cdot)$, to predict $\hat{\mathbf{y}} = (\hat{y}_{t+1}, \cdots, \hat{y}_{t+H})$ given the input sequence $(x_{t-B}, \cdots, x_t)$.

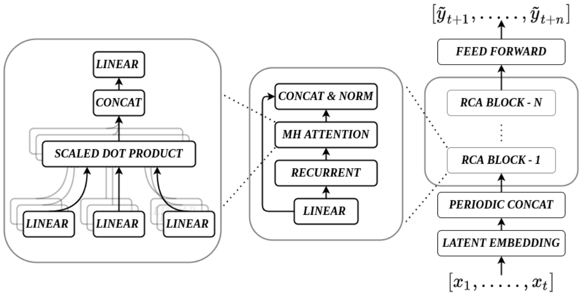


Figure 4: Architecture of the MDP state forecaster (RCAN)

Latent Embedding Module (LEM): Latent embedding module is a two layered linear network. The first layer operates on different features of the input such as calendar, weather or time series by imposing feature specific constraints. Specifically, let x_k, the k-th component of input vector $\mathbf{x}$, be partitioned into three features as $x_k = [z_k, e_k, c_k]$ where z_k denotes the corresponding (input) time series, c_k represents the calendar information and e_k represents the exogenous components. Specifically, we use the following parameters as exogenous variables in forecasting market demand, market supply, market clearing price & market clearing quantity: temperature, wind speed, solar insolation, and humidity. Each feature of the input is now fed into a latent embedding network to get a suitable latent representation as in

$$Z_k = f_z(z_k), \quad E_k = f_e(e_k), \quad C_k = f_c(c_k)$$

where $[f_z, f_e, f_c]$ are the set of latent transformation functions specific to the variable type. The resultant latent representation for x_k is then concatenated and passed into the second linear layer of the LEM and the final output is denoted as $X_k = [Z_k, E_k, C_k]$. The latent representation for the input sequence $\mathbf{x}$ is then denoted as $\mathbf{X} = [\mathbf{Z}, \mathbf{E}, \mathbf{C}]$ where $\mathbf{Z} = (Z_{t-B}, \cdots, Z_t), \mathbf{E} = (E_{t-B}, \cdots, E_t)$ and $\mathbf{C} = (C_{t-B}, \cdots, C_t)$.

Periodic Concatenation (PC) Module: The PC module slices the latent representation of the input sequence with slice length L and combines all latent representations of same time-step to produce a concatenated output.

Regression Coupled Attention (RCA) Module : The RCA module is a special stack of layers namely, linear regression layer, recurrent neural layer and multi-head attention layer, backed with skip connection and layer normalization. The first component in RCA module is a linear layer, which performs regression on the concatenated sequences of the latent representation to produce a hidden representation of the forecast sequence. The second component in RCA module is a recurrent layer backed by GRU/LSTM implementation. For each future time step, the GRU/LSTM block processes the combination of hidden representation from the regression layer and a memory state from all past time-steps to produce a dense representation with sequential relationship.

The third component in RCA module is the multi-head attention layer. In literature, multi-head attention mechanism was invented to improve the long-term dependency in learning and identifying relevant feature sub spaces [24]. The attention mechanism gives the flexibility to focus on significant time steps located at different positions in the observable window. The multi-head comes with an idea that rather than computing the attention on the whole set of features, we can compute multiple set of scores by attending information from different sub-spaces at different time steps. These independent attention outputs can then be concatenated and linearly transformed into suitable dimensions in an ensemble fashion. Finally, in order to reduce the forecast uncertainty, we take aid from dropout layers used in training for regularization. During the prediction phase, stochastic nature of these layers is used to obtain multiple predictions for the same input sequence, which are then averaged to generate the forecast. The details of the attention mechanism can be found in [24].

6 MARKET CLEARING MODEL

Recall that $\Phi_{s,n+h}$ denotes the set of m asks submitted by a supplier $s \in \mathcal{S}$ at time n for auction at time block $n + h$. One can then rearrange the m asks in $\Phi_{s,n+h}$ of a supplier s in increasing order of the prices such that $p^k_{s,n+h} \geq p^{k+1}_{s,n+h}, k \in \{1, \cdots, m-1\}$. Denote $q^k_{s,n+h}$ as the quantity that is part of the supplier ask at price $p^k_{s,n+h}$. Similarly, the m' bids supplied by the buyer for auction time slot $n+h$ are arranged in decreasing order of price. The market regulator gathers the bids and asks pertaining to an auction and performs a merit order dispatch. That is, the cheaper asks are cleared before the expensive ones and costlier bids are cleared before the inexpensive ones until the demand constraint is met. The dispatch mechanism also outputs a market clearing price and cleared quantity for every ask and bid submitted by a market participant. A unique market clearing price ensures uniform market clearing process. The formulation given below mimics the aforesaid process.

$$\max_{\mathbf{q_b}, \mathbf{q_s}} \left(\sum_{b \in \mathcal{B}} \sum_{k=1}^{m'} p^k_{b,n+h} \times q^k_{b,n+h} - \sum_{s \in \mathcal{S}} \sum_{k=1}^{m} p^k_{s,n+h} \times q^k_{s,n+h} \right) \tag{6}$$

$$0 \leq q^k_{b,n+h} \leq q^{\max}_{b,n+h} \quad \forall \quad b, k \tag{7}$$

$$0 \leq q^k_{s,n+h} \leq q^{\max}_{s,n+h} \quad \forall \quad s, k \tag{8}$$

$$\left(\sum_b \sum_k q^k_{b,n+h} - \sum_s \sum_k q^k_{s,n+h} \right) = 0 \tag{9}$$

The optimization objective in Equation (6) maximizes the *social welfare* criterion [18] where the optimization variables $\mathbf{q_b}$ and $\mathbf{q_s}$ represent vector of cleared quantities for bids and asks submitted in the auction. This is a fair way to match bids with asks which ensures that the clearing process is optimal to both the supply and demand side. Equations (7) and (8) indicate that the maximum quantity ($q^{\max}$) that a market participant is willing to buy or sell at a given price. Equation (9) ensures a lossless system in which the net imbalance between supply and demand at all times is zero. The Lagrange multiplier of the demand-supply balance constraint – the equality constraint in (9), is the market clearing price p^*_{n+h} for the auction at time slot $n + h$. The solution procedure can also be used to work out the cleared quantities for each bid (ask) and total cleared quantity (q^*_{n+h}) for the market. Although, the above model considers auctions pertaining to only one region, it is possible to extend the formulation to include multiple regions.

7 EXPERIMENTAL ANALYSIS

We evaluate the performance of the proposed RL based bidding framework using real world market data obtained from the European power exchange (EPEX).

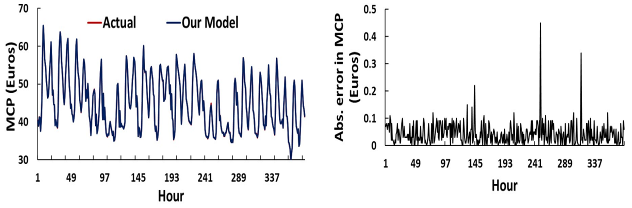
EPEX Market and Data Availability: We obtained logs of daily market operations from the SPOT day-ahead electricity market in France, operated by European Power Exchange (EPEX)[3]. Though this market supports both simple and block bids/asks, we consider data pertaining to the simple bids/asks alone since this is the focus of our work. This market is two-sided with blind auctions in which electricity sellers and buyers participate anonymously. The participants submit multiple bids for each time block of a day in advance as per the timeline of day-ahead market. EPEX collects the bids and asks from all participants and applies its market clearing algorithm to publish market clearing price (MCP) and market clearing price quantity (MCQ). This market follows an uniform market pricing mechanism. EPEX's market clearing process does not consider network power flows for the day-ahead market.

We obtained daily logs of EPEX's day-ahead market for the years 2016-2019. The logs consist of the aggregated bid curves and ask curves obtained from all the buyers and sellers respectively for each hour of a day. We could not obtain the individual bid/ask curves submitted by individual market participants since such data is considered confidential by EPEX and is not shared. However, we could obtain the generation capacity curve for one supplier (referred to as $\mathcal{G}$ henceforth) at a daily level and information about its 'asks'. The daily generation capacity of $\mathcal{G}$ varies between 4900 MW to 5150 MW. We note that this supplier $\mathcal{G}$ could also be a group of individual generation plants acting in unison. $\mathcal{G}$ has very fast ramping rates – its generation could be ramped up or down across its full dynamic range within 10 minutes, which is quite small in comparison to the overall market auction slot duration of 60 minutes. This allows us to use our RL framework to design asks on behalf of $\mathcal{G}$.

We initially present detailed results on our framework's performance from the perspective of $\mathcal{G}$ as a seller. We then consider $\mathcal{G}$ to be a buyer with matching attributes and discuss sample results.

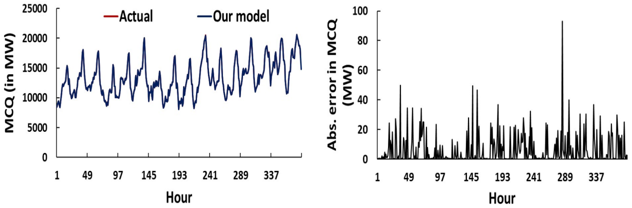
7.1 Market clearing & Forecasting modules

Efficacy of Market Clearing Model: We tested the accuracy of our market clearing model described in Section 6 against the real world data obtained from EPEX. The accuracy of the market model is quite critical since this is the basis for deriving the reward values for the RL agent during the learning and testing phases. We evaluate the accuracy of our market model with respect to two parameters: (i) ability to estimate market clearing price, and (ii) ability to estimate market clearing quantity. The performance of our market model in estimating MCP and MCQ on the EPEX market data over a period of two weeks in February 2019 is depicted in Figures 5 and 6 respectively. We found that the mean absolute error in estimating the MCP over a wider duration (between Feb 2019 to Sep 2019) to be €0.06 which amounts to a mean relative absolute error of 0.16%. For the same period, the mean absolute error in estimating the MCQ on EPEX data between Feb 2019 to Sep 2019 came out to be 3.82 MW. In relative terms, our market model gives an error of 0.02% when estimating the MCQ. These results convey that our proposed market model is good enough to be used as a proxy for the actual EPEX's market clearing mechanism for simple bids.



(a) Time series of actual & estimated values (b) Absolute value of estimation error

Figure 5: Market clearing model: MCP estimation accuracy.



(a) Time series of actual & estimated values (b) Absolute value of estimation error

Figure 6: Market clearing model: MCQ estimation accuracy.

Efficacy of Forecasting Module: The forecasting module is also critical for the bidding framework since this gives forecasts of the unobserved state variables for the underlying MDP. Following EPEX data format, our forecast module estimates with a look-ahead horizon of 24 hours (time steps). Specifically, we predict four parameters namely, MCP, MCQ, volume of supply traded in the market, and volume of demand traded in the market. For the sake of brevity, we show the forecast performance on two of the four variables – namely market demand and market clearing price in Table 1. We compare our RCAN prediction with two popular baselines, namely, stacked GRU and stacked GRU encoder-decoder (ED) using popular metrics such as mean-absolute error (MAE) and root mean-square error (RMSE) [6]. To obtain these results, all models were trained on EPEX market data from 2016 to 2018. Data from the year 2019 was used for testing the performance. The input features to the training process include past historical lags of the time series, weather and calendar features. The table entries show the forecast error that has been averaged across the 24 look-ahead values for the time slots in the entire year of 2019.

Model	Market Demand (MW)			Market Clearing Price (€)		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE
RCAN	928.9	1474.5	7.86%	2.02	3.48	12.6%
Stacked GRU	1480.2	2005.4	8.56%	2.96	4.32	14.58%
Stacked ED	1108.9	1690.3	8.07%	3.77	5.15	13.79%

Table 1: Performance of RCAN, Staked GRU and ED models.

As can be seen from table 1, the proposed forecaster RCAN performs better than the other two techniques and the error obtained is reasonable enough to be used as part of the bid/ask designer. The improved performance of our RCAN model can be attributed to the multi-head attention layer which is not a part of vanilla RNN based forecasters. The attention aspect helps RCAN identify important time steps in the observable window. The multi-head mechanism computes multiple such attention outputs which are then combined to form an ensemble prediction.

7.2 Efficacy of Bid/Ask Designer

We discuss the performance of our bid/ask designer module which is at the core of our proposed RL framework. We measure the performance based on the revenue earned by placing the asks suggested by the designer on behalf of generator $\mathcal{G}$.

We compare our bid/ask designer's performance against three baselines, namely (i) *Ideal*: a baseline that knows the actual behavior of all other market participants (i.e., their bid/ask curves) without any error and determines the best action through an exhaustive evaluation of all actions available; (ii) *Exhaustive Action based on Forecasts (EAF)*: a baseline that estimates the behavior of other market participants through a simple moving average estimator before determining the best action in exactly the same manner as the Ideal baseline; (iii) *Historical*: a baseline that derives historical revenue of $\mathcal{G}$ from market logs. Note that while the Ideal method is an upper bound on the performance that can be achieved, it is not realizable in practice. However, EAF can be realized in practice.

We note here that EPEX's day-ahead market uses uniform pricing strategy. For generator $\mathcal{G}$, we had the historical logs that provide the historical market clearing quantity as well as the market clearing price. Multiplying the two gives the historical revenues for $\mathcal{G}$. Using our RL agent, we design bids for $\mathcal{G}$. These RL bids are merged with the historical bids from the competition to get the new overall supply side bids for the day-ahead market. The new aggregate supply side bids and the historical demand side bids are then fed to our market clearing model to obtain the new market clearing price and the new market clearing quantity for $\mathcal{G}$. Multiplying these two gives the new revenue obtained using RL. Through a similar process, the revenues for $\mathcal{G}$ under *Ideal* and *EAF* are determined.

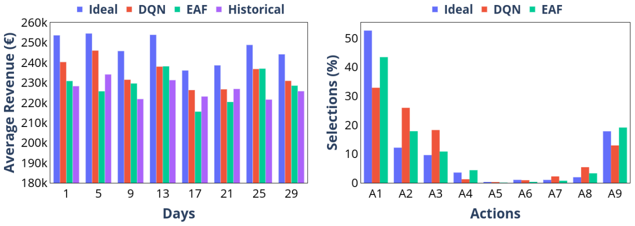
Experimental Setup: As mentioned earlier, we had EPEX market logs for the years 2016-2019. We use data from the years 2016 and 2017 to train the RL agent. We test the performance using data from the years 2018 and 2019. We use each day's log as an independent episode for training and testing. Our market logs had the overall market supply curve, which is the aggregate of ask curves submitted by $\mathcal{G}$ and the curves submitted by $\mathcal{G}$'s competitors. We obtain the aggregated ask curve of the competitors by subtracting the $\mathcal{G}$'s ask curve from the overall market supply curve. This competition ask curve and the market demand curve are fed to our market clearing module along with the asks supplied by the DQN (from our bid/ask designer) to obtain the reward during both training and testing.

Gen. Capacity	Ideal	Proposed	EAF	Historical
Original	249.51	237.04	229.80	228.05
Original - 10%	175.98	171.21	164.26	171.04
Original + 10%	337.58	317.74	311.99	285.06

Table 2: Avg. daily revenue across test days for different generation capacities (× 1000 €).

The DQN is designed with 2 hidden layers with 128 units each, an input layer with number of state variables (15), and an output layer with number of discrete actions (9). Hidden layers are activated by the Rectified Linear Unit (ReLU), and the output layer is activated with linear function. During training, batch gradient descent is performed using Adam optimization. We used a variant of the DQN algorithm called Double DQN [23]. The following values are used for the hyper-parameters in Algorithm 1 : i) Replay buffer capacity $R = 10000$; ii) Batch size $B = 64$; iii) Episode length $T = 24$ time slots (1 day); iv) Number of Episodes $M = 730$; v) Explore-exploit trade off parameter $\varepsilon = 0.9$; and vi) ε-decay step size is set as $\Delta\varepsilon = (0.9 \times M)^{-1}$; vi) For the reward function r_n, we use $\beta_1 = 1.0$ and $\beta_2 = 0$. We set the discount factor $\gamma = 0.1$ to give importance for hourly revenue. We set $p_{max} = 100$ for our action space.

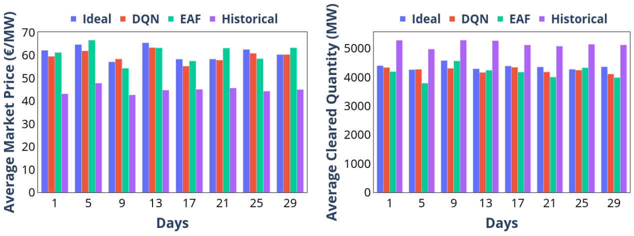
Comparison against baselines: The revenue accumulated using the asks generated by our RL framework during the testing period is compared with the three baselines mentioned above in Figure 7 (a). For ease of viewing, the average revenue across several sample day-types is shown. Also, the average revenue accumulated per day across the entire test period of 2018-19 is given in table 2. Based on these results, we infer the following. First, the proposed RL framework does improve the revenue accumulated in comparison to historical values by nearly €9,000 per day or €3.2 M per year (4% increase) which is a considerable amount. Second, among other baselines, the proposed RL framework is the one that comes closest to the performance of the Ideal strategy. Third, the performance of our RL framework is consistently better than EAF. We also observe that the revenue yield of EAF is lower than the historical values at times. The poor performance of EAF could be attributed to the erroneous predictions of the bid/ask curves of other market participants (buyers and competitive suppliers). *This shows that in the absence of accurate forecasts of the behavior of other market participants (for an explicit optimization), a RL based framework is a better alternative.*



(a) Daily revenue (b) Actions chosen

Figure 7: Performance of different ask generation schemes

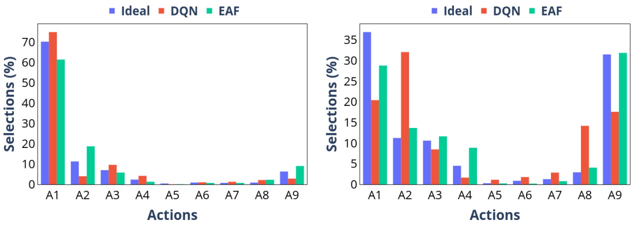
Figure 7(b) also shows the histogram (in percentage) of various actions picked by different ask generation schemes. The action profiles are arranged in increasing order of their aggressiveness – action A1 being the most conservative and A9 the most aggressive. We note that the relative proportion of conservative and aggressive actions varies across the schemes which leads to their performance difference. Figure 8 gives some insights as to how the actions taken by different schemes affect the overall market operations. We find that in their quest to push the earnings higher, almost all the non historical schemes design their asks in such a way which pushes the market clearing price higher than that achieved by the historical asks. Consequently, the quantum of generation cleared by the market under all these schemes fall lower than under the historical asks. Nevertheless, Ideal and the proposed RL framework end up earning higher revenues consistently.



(a) Avg. Market Clearing price (b) Avg. Market clearing quantity

Figure 8: Avg. MCP and Avg. MCQ under various ask generation schemes.

Impact of generation capacity: We study the ability of the proposed RL framework to learn the optimal ask strategy under different generation capacities. To do this, we both decrease and increase the baseline generation capacity of $\mathcal{G}$ by 10% and re-train the RL framework. From table 2, we see that even when the generation capacity is changed, the proposed RL framework works reasonably well. With a lower capacity, the amount of market power that $\mathcal{G}$ has in influencing the market operations decreases. Consequently, the RL agent takes more conservative actions to make sure that its earnings are not affected. As the capacity of $\mathcal{G}$ increases, it gains more market power. Hence our RL framework learns to pick aggressive actions more frequently to boost the revenue. This learning is clearly seen in Figure 9 which shows the action profiles chosen by various schemes under different generation capacities.



(a) Decrease by 10% (b) Increase by 10%

Figure 9: Action profiles chosen by RL framework under different generation capacities.

Performance from a buyer's perspective: For this study, we consider the daily capacity curve specified for the given generator $\mathcal{G}$ to be the aggregate demand that a buyer b should procure from the market. The hyper parameters used in training of the DQN remain same as before. The values of β_1 and β_2 used in the reward function are −1 and −2.5 respectively. A non-zero negative value for β_2 encourages the RL agent to buy most of the required power from wholesale market rather than from balancing market. The goal of the bid/ask designer is to place bids such that the overall cost of energy required to meet the demand is reduced.

Buy Capacity	Ideal	Proposed	EAF	Historical
Original	209.94	219.8	222.44	227.01
Original - 10%	167.64	172.29	175.44	170.26
Original + 10%	236.27	246.64	248.96	283.77

Table 3: Avg. daily purchase cost across test days (× 1000 €)

We consider the same baselines as before (Ideal, EAF, and Historical) against which we compare the performance of the our bid/ask designer. Table 3 presents the average daily cost of meeting the daily demand through the bids placed by various schemes across the entire test period of 2018-19. We infer that the Ideal strategy has the lowest procurement cost followed by our proposed bid/ask designer. In comparison to historical performance, our RL based strategy reduces the daily procurement cost by nearly €7,200 (€2.63 M per year). Results are also shown for the cases when the daily demand requirement is increased and decreased by 10%. As in the case of generators, the performance of EAF remains inconsistent.

8 MODELING RAMPING CONSTRAINTS

State Space: Recall that at time step n, the state space described earlier in Section in 4 contained the capacity available for the participant at time step $n+1$. In order to model the ramping constraint, this state variable needs to be replaced with the following three state variables: (a) the utilized capacity (consumption or generation) of a given participant at n ($\widehat{\mathcal{U}}_n$); (b) the amount by which the utilized capacity can be increased at n over one time slot (Δ_n^+); and (c) the amount by which the utilized capacity can be decreased at n over one time slot (Δ_n^-). Other state variables remain unchanged. We also note here that the ramping in either direction can be asymmetric, i.e., Δ_n^+ need not be equal to Δ_n^-. Since we are focusing on day ahead markets, at a given time slot n, the bid/ask designer should place bids/asks for slots $n+1$ to $n+H$ as a batch. The values of $\widehat{\mathcal{U}}_{n+1}$ through $\widehat{\mathcal{U}}_{n+H}$ can be estimated sequentially based on the action suggested by the RL agent at instants n through $n+H-1$. We use the MDP state forecaster to estimate the other state variables.

Action Space: The action space in the presence of ramping constraints can be designed by observing that the maximum and minimum amount of generation/consumption that can be achieved at instant $n+1$ would now depend on the utilized capacity $\widehat{\mathcal{U}}_n$ at time slot n. Hence, the participant has one of the following choices with respect to its ramping operation: (a) ramp up so that its available capacity at $n+1$ increases to $\widehat{\mathcal{U}}_n + \Delta_n^+$; (b) available capacity remaining unchanged between n and $n+1$; and (c) ramp down so that the available capacity at $n+1$ decreases to $\widehat{\mathcal{U}}_n - \Delta_n^-$. Once the ramping operation is determined, the actual bids/asks can be determined in same way as in Section in 4 using the nine capacity distribution profiles. Thus for each state transition from n to $n+1$, the overall action space consists of 27 actions – nine capacity distribution profiles for each ramping operation. Once the action a_n for instant n has been determined, the MDP transits to another state at time $n+1$. At time $n+1$, the best action a_{n+1} is determined and the MDP moves to another state at time $n+2$ and so on until $n+H$. It is possible that in spite of modeling the ramping constraints when placing the bids/asks, the actual quantity cleared by the market may violate the ramping constraints. This can be accommodated by adding an additional penalty term to the reward function for those instants in which the constraints are violated.

9 CONCLUSION & LIMITATIONS

In this paper, we proposed a holistic RL based framework to increase (decrease) the revenues (cost) of a participant in a two-sided, day-ahead, wholesale electricity market. Our framework is generic enough to be used by a buyer as well as a seller. We tested the performance of our framework on real world market data obtained from the European Power Exchange (EPEX). We found that the proposed framework was able to improve the historical revenues of a generator by nearly 4% (€3.28 M per year). We observe that when our framework is used from a buyer's perspective, it reduces the daily procurement cost by nearly 3.2% (€2.63 M per year). We also observed that our RL based framework automatically adapts its action set as per the capacity of the market participant.

Our proposed RL framework relies on the forecaster module to give accurate forecasts of the state variables that are non observable. Inaccurate forecasts can be detrimental to the performance of our RL based trading strategy. Further studies are needed to make the overall MDP formulation more robust to such forecast errors. Also, in our current work, network congestion is not modeled in the market clearing process. If network congestion has to be modeled in market clearing (as done in North American markets), then additional inputs to estimate the network state are required.

In future, we aim to extend our work to address the above limitations. Also, we plan to investigate the use of RL based trading for other electricity markets such as intra-day and balancing markets.

ACKNOWLEDGMENTS

We sincerely thank all the reviewers and our shepherd Prof. Carol Menassa for their inputs. Without doubt, their comments have greatly improved the paper's quality.

REFERENCES

[1] Ahmad Attarha, Paul Scott, and Sylvie Thiébaux. 2020. Network-aware Participation of Aggregators in NEM Energy and FCAS Markets. In *Proceedings of the Eleventh ACM International Conference on Future Energy Systems, e-Energy 2020, Virtual Event, Australia, June 22-26, 2020*. ACM, Virtual Event, Australia, 14–24.

[2] Moinul Morshed Porag Chowdhury, Christopher Kiekintveld, Son Tran, and William Yeoh. 2018. Bidding in Periodic Double Auctions Using Heuristics and Dynamic Monte Carlo Tree Search. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, Jérôme Lang (Ed.). ijcai.org, 166–172.

[3] EPEX. 2020. European Power Exchange. https://www.epexspot.com/.

[4] Susobhan Ghosh, Easwar Subramanian, Sanjay P. Bhat, Sujit Gujar, and Praveen Paruchuri. 2020. Bidding in Periodic Double Auctions : Theory, Analysis and Strategy. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, Newyork, USA*. AAAI Press.

[5] Dhananjay K Gode and Shyam Sunder. 1993. Allocative efficiency of markets with zero-intelligence traders: Market as a partial substitute for individual rationality. *Journal of Political Economy* 101, 1 (1993), 119–137.

[6] Ian J. Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press, Cambridge, MA, USA. http://www.deeplearningbook.org.

[7] José Iria, Filipe Soares, and Manuel Matos. 2019. Optimal bidding strategy for an aggregator of prosumers in energy and secondary reserve markets. *Applied Energy* 238 (2019), 1361 – 1372. https://doi.org/10.1016/j.apenergy.2019.01.191

[8] J. P. Iria, F. J. Soares, and M. A. Matos. 2019. Trading Small Prosumers Flexibility in the Energy and Tertiary Reserve Markets. *IEEE Transactions on Smart Grid* 10, 3 (2019), 2371–2382.

[9] Wolfgang Ketter, John Collins, and Mathijs Weerdt. 2017. The 2018 Power Trading Agent Competition. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3087096/

[10] T. Krause, G. Andersson, D. Ernst, E. Vdovina-Beck, R. Cherkaoui, and A. Germond. 2004. Nash equilibria and reinforcement learning for active decision maker modelling in power markets. In *6th IAEE EuropeanConference: Modelling in Energy Economics and Policy*.

[11] Sagar Kurandwad, Chandrasekar Subramanian, P. Venkata Ramakrishna, Arunchandar Vasan, Venkatesh Sarangan, VijaySekhar Chellaboina, and Anand Sivasubramaniam. 2014. Windy with a chance of profit: bid strategy and analysis for wind integration. In *The Fifth International Conference on Future Energy Systems, e-Energy '14, Cambridge, United Kingdom - June 11 - 13, 2014*. 39–49.

[12] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (26 02 2015), 529–533. http://dx.doi.org/10.1038/nature14236

[13] M.B. Naghibi-Sistani, M.R. Akbarzadeh-Tootoonchi, M.H.Javidi-Dashte Bayaz, and H.Rajabi-Mashhadi. 2006. Application of Q-learning with temperature variation for bidding strategies in market based power systems. *Energy Conversion and Management* 47, 11 (2006), 1529 – 1538.

[14] Karsten Neuhoff, Nolan Ritter, and Sebastian Schwenen. 2015. *Bidding Structures and Trading Arrangements for Flexibility across EU Power Markets*. EconStor Research Reports 111922. ZBW - Leibniz Information Centre for Economics. https://ideas.repec.org/p/zbw/esrepo/111922.html

[15] Martin L Puterman. 1994. *Markov Decision Processes*.

[16] Daniel L. Rubinfeld Robert S. Pindyck. 2009. *Microeconomics*.

[17] S. Shafiee, H. Zareipour, and A. M. Knight. 2019. Developing Bidding and Offering Curves of a Price-Maker Energy Storage Facility Based on Robust Optimization. *IEEE Transactions on Smart Grid* 10, 1 (2019), 650–660.

[18] Devnath Shah and Saibal Chatterjee. 2020. A comprehensive review on day-ahead electricity market and important features of world's major electric power exchanges. *International Transactions on Electrical Energy Systems* 30, 7 (March 2020). https://doi.org/10.1002/2050-7038.12360

[19] Bethany Speer, Mackay Miller, W Schaffer, Leyla Gueran, Albrecht Reuter, Bonnie Jang, and Karin Widegren. 2015. *Role of smart grids in integrating renewable energy*. Technical Report. National Renewable Energy Lab.(NREL), Golden, CO (United States).

[20] Easwar Subramanian, Yogesh Bichpuriya, Avinash Achar, Sanjay P. Bhat, Abhay Pratap Singh, Venkatesh Sarangan, and Akshaya Natarajan. 2019. lEarn: A Reinforcement Learning Based Bidding Strategy for Generators in Single sided Energy Markets. In *Proceedings of the Tenth ACM International Conference on Future Energy Systems, e-Energy 2019, Phoenix, AZ, USA, June 25-28, 2019*. ACM, Phoenix, AZ, USA, 121–127.

[21] Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction* (second ed.). The MIT Press.

[22] Gerald Tesauro and Jonathan L. Bredin. 2002. Strategic Sequential Bidding in Auctions Using Dynamic Programming. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 2* (Bologna, Italy) *(AAMAS '02)*. ACM, New York, NY, USA, 591–598.

[23] Hado van Hasselt, Arthur Guez, and David Silver. 2015. Deep Reinforcement Learning with Double Q-learning. (2015). http://arxiv.org/abs/1509.06461 cite arxiv:1509.06461Comment: AAAI 2016.

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *NIPS*. 5998–6008.

[25] X. Wang and T. Sandholm. 2002. Reinforcement learning to play an optimal Nash equilibrium in team Markov games. In *Advances in neural information processing systems*. 1571–1578.

[26] Christopher JCH Watkins and Peter Dayan. 1992. Q-learning. *Machine learning* 8, 3-4 (1992), 279–292.

[27] Robert Wilson. 1992. Strategic analysis of auctions. *Handbook of Game Theory with Economic Applications* 1 (1992), 227–279.

[28] Peter R Wurman, William E Walsh, and Michael P Wellman. 1998. Flexible double auctions for electronic commerce: Theory and implementation. *Decision Support Systems* 24, 1 (1998), 17–27.

[29] Y. Ye, D. Qiu, M. Sun, D. Papadaskalopoulos, and G. Strbac. 2020. Deep Reinforcement Learning for Strategic Bidding in Electricity Markets. *IEEE Transactions on Smart Grid* 11, 2 (2020), 1343–1355.