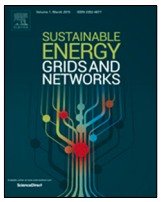
Sustainable Energy, Grids and Networks 34 (2023) 101023

Contents lists available at ScienceDirect

Sustainable Energy, Grids and Networks

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



Statistical arbitrage trading across electricity markets using advantage actor–critic methods☆

Sumeyra Demir *, Koen Kok, Nikolaos G. Paterakis

Department of Electrical Engineering, Eindhoven University of Technology, The Netherlands

ARTICLE INFO

Article history:
Received 27 July 2022
Received in revised form 6 February 2023
Accepted 22 February 2023
Available online 26 February 2023

Keywords:
Algorithmic trading
Day-ahead market
Deep reinforcement learning
Electricity price forecasting
Energy trading
Intraday market
Machine learning

ABSTRACT

In this paper, risk-constrained arbitrage trading strategies that exploit price differences arising across short-term electricity markets, namely day-ahead (DAM), continuous intraday (CID) and balancing (BAL) markets, are developed and evaluated. To open initial DAM positions, a rule-based trading policy using DAM and CID price forecasts is proposed. DAM prices are predicted using both technical indicator features and data augmentation methods, such as autoencoders and generative adversarial networks. Meanwhile, CID prices are predicted using novel features that are engineered from the limit order book. Using the forecasts, the direction of price movements is correctly predicted the majority of the time. To manage open DAM positions while optimising the risk-reward ratio, deep reinforcement learning agents trained using the advantage actor–critic algorithm (A2C) are employed. Evaluated across Dutch short-term markets, A2C yields profits surpassing those obtained using A3C and other benchmarks. We expect our study to benefit electricity traders and researchers who seek to develop state-of-art intelligent trading strategies.

1. Introduction

1.1. Motivation and background

Arbitrage opportunities frequently arise when prices deviate from their long-term means. Arbitrage traders try to profit from such opportunities without exposing themselves to any long term physical commitments. Arbitrage trading offers financial incentives to traders and increases market liquidity and efficiency [1,2].

In this paper, we evaluate the profitability of statistical arbitrage trading (SAT) strategies, which leverage intelligent learning methods. SAT strategies employing deep neural networks, technical indicators, data augmentation and the synchronous advantage actor–critic (A2C) algorithm are developed and analysed across short-term electricity markets, namely the day-ahead market (DAM), continuous intraday market (CID) and real-time balancing market (BAL). Our autonomous agents first open a position on the auction-based DAM, before closing this position on the continuous-based CID or the BAL.

To ensure the profitability of our autonomous agents, as a first step accurate forecasts of DAM prices must be obtained. To attain accurate DAM forecasts, we employ neural networks, which have been shown to outperform statistical models in DAM forecasting [3–5]. Additionally, we propose using technical indicators and data augmentation. Demir et al. [5] highlighted how the use of technical indicator features, such as moving averages and Bollinger bands, can improve the accuracy of machine learning models. Technical features assist in the identification of behavioural biases of DAM traders. Meanwhile, [6] highlighted how data augmentation methods using autoencoders and generative adversarial networks boost forecast accuracies. To the best of our knowledge, we are the first to employ both technical indicator features and data augmentation to forecast DAM prices.

As a second step, we employ deep reinforcement learning (DRL) to optimise the decision-making process of placing trades across the CID. DRL is selected because of its numerous documented successes, such as [7,8], in solving sequential decision problems. Among DRL algorithms, advantage actor–critic algorithms have proven particularly adept at solving sequential decision problems because of their ability to reduce the volatility of gradient updates [9,10]. Yang et al. [11,12] employed A2C in optimising trading decisions across equity markets. Meanwhile, [13] applied A2C to optimise the decision making of a retail electricity trader. To the best of our knowledge, we are the first to employ A2C in the context of SAT across the CID.

1.2. Relevant literature

SAT studies have thus far predominately analysed and developed trading strategies that open a position on the DAM before

☆ This research project is funded by Scholt Energy, The Netherlands.
* Corresponding author.
E-mail address: s.demir@tue.nl (S. Demir).

https://doi.org/10.1016/j.segan.2023.101023

Nomenclature

Arbitrage Trading

σ	Standard deviation of *PnL* distribution
C^{bal}	Cash made on the balancing market
C^{cid}	Cash made on the continuous intraday market
C^{dam}	Cash made on the day-ahead market
d	Day
h	Delivery hour index
K	Total number of CID transactions for a contract.
N	Total number of contracts in the test set
PnL	Profit, agent's cash portfolio
PnL^{high}	Upper profit limit for agent's cash portfolio
PnL^{low}	Lower profit limit for agent's cash portfolio
TC	Trading costs

Prices

$\hat{p}_{d+1}$	Day-ahead price forecast
p	Traded price for a transaction
$p^{a_{high}}$	The highest ask price across a trading session
$p^{a_{low}}$	The lowest ask price across a trading session
p_t^a	The best ask price at time step t
$p^{b_{high}}$	The highest bid price across a trading session
$p^{b_{low}}$	The lowest bid price across a trading session
p_t^b	The best bid price at time step t
p^{dam}	Day-ahead market price
p^{feed}	Settled balancing market price for long positions
p^{take}	Settled balancing market price for short positions
p^{vwap}	Volume-weighted average price of trades for the continuous intraday market

Quantities

$\sum q$	Total arbitraged quantity
q	Traded quantity for a transaction
q^a	Available quantity for the best ask order
q^b	Available quantity for the best bid order
q^{high}	Maximum allowed total traded quantity
v	Position, agent's volume portfolio
v^{max}	Maximum long position
v^{min}	Minimum short position

Reinforcement Learning

β	Learning rate for actor–critic algorithms
$\mathbf{s}_t$	State at time step t
ϵ	Epsilon index
γ	Discount factor
π	Policy
τ	Threshold for rewards
τ^B	Threshold for buy rewards
τ^S	Threshold for sell rewards
θ_π	Parameters for the global policy
θ'_π	Parameters for a local policy
θ_V	Parameters for the global value function
θ'_V	Parameters for a local value function
A_π	Advantage for the policy
a_t	Action at time step t
B	Action buy
e	Episode index
e^{max}	Predefined maximum number of episodes
f^B	Function for scaling buy rewards
f^S	Function for scaling sell rewards
H	Action hold
J	Total number of states
L	Number of hidden layers for deep neural networks of actor–critic algorithms
l	Hidden layer index for deep neural networks of actor–critic algorithms
n_l	Number of neurons for the hidden layer l
$P(B)$	Probability for the buy action
$P(H)$	Probability for the hold action
$P(S)$	Probability for the sell action
R	Return, total reward
r_t	Reward at time step t
r_t^B	Reward at time step t for a buy action
r_t^H	Reward at time step t for a hold action
r_t^S	Reward at time step t for a sell action
S	Action sell
T	Number of trading periods, terminal state
t	Trading period index
t^{max}	Number of time steps to update the global network
V	Value function
W	Number of agents
w	Agent index

closing it out on the BAL. Detailing some of these studies: a stochastic optimisation approach was implemented by [14,15], a min–max two-level optimisation model was analysed by [16], a data-driven approach was explored by [17], a machine-learning approach was implemented by [18], and an online-learning algorithm was analysed by [2]. All of the above studies, evaluated across US markets, identified profitable trading strategies.

Analysing some of the few SAT studies which have developed trading strategies for the CID and BAL, [19] evaluated a rule-based trading agent that used forecasts of demand and imbalance volume to place trades on the CID. By making a decision every 30 min, with a fixed 2 MW order quantity, and closing out any outstanding positions on the BAL, [19] obtained positive profits across British CID and BAL markets. Recently, [20] also developed an intelligent CID trading agent; trained using the asynchronous advantage actor–critic (A3C) algorithm, i.e. the asynchronous version of A2C. While DRL trading applications frequently rely on a pre-defined set of states, [20] developed state engineering and selection methods to identify and select states with the greatest

explanatory power. By making a decision after every impactful limit order book update, with a maximum 1 MW order quantity, and closing out any outstanding positions on the BAL, [20] successfully obtained significantly positive profits across Dutch markets.

Because forecasts of DAM and CID prices are used in this study to identify profitable trades on the DAM, we also describe DAM and CID forecasting studies below. Lago et al. [3] found that machine learning forecasting models outperform statistical models for the French, Belgian, German, Nordic and American DAM markets. Similar results were obtained for the Belgian and Dutch DAM markets by [4–6]. Lago et al. [4] further found that neural networks outperform long-short-term-memory networks and gated recurrent units. Several studies, such as [5,21,22], investigated the importance of various features. They found that some of the most important predictors for forecasting DAM prices were neighbouring country DAM prices. Lago et al. [21], for instance, tested this across the Belgian and French markets, and [5] for the Belgian and Dutch markets.

In [23–25], it was demonstrated that hybrid models outperformed individual benchmark models. In [24], a wavelet transformation, an autoregressive moving average model, a kernel extreme learning machine (KELM), and self-adaptive particle swarm optimisation (SAPSO) were used to harmonise the hybrid model. Meanwhile, deep belief networks, SAPSO, SARIMA, and variational mode decomposition were used to build the hybrid model in [24]. In [24], forecasting model accuracies were evaluated for three separate DAMs. However, exogenous variables were not examined in this study. In [25], a local forecasting paradigm, a generic regression neural network, coordinate delay, and a harmony search method were used to form the hybrid model. Exogenous variables were used in this work, unlike [24]. However, in [25], model accuracies were only evaluated for one DAM. Applications of spike forecasting were examined in more detail in [26]. [26] improved the DAM predicting accuracy by balancing the number of samples in the various target classes, i.e. by increasing the amount of spike samples in the training data, using the Borderline-SMOTE approach.

Examining price forecasting studies focusing on the CID, [27] compared neural network-based forecasting models for the Turkish market and [28] utilised principal component analysis for the German market. Narajewski et al. [29–32] investigated various features, such as forecast errors and seasonal dummy variables. Uniejewski et al. [30] discovered that the most recent CID prices and the DAM price are the most important predictors of forecasting CID prices for the German market. Hagemann [31] found wind power forecast errors as the most important for the same market.

1.3. Contributions and organisation

While SAT studies have thus far focused on exploiting opportunities arising between the DAM-BAL or CID-BAL, to the best of our knowledge no study has yet developed or analysed a trading strategy capable of simultaneously exploiting arbitrage opportunities arising across all short-term electricity markets. Given the benefits SAT confers on markets and traders, we investigate the profitability of SAT across the DAM-CID-BAL.

We propose employing a rule-based trading agent, which uses forecasts of DAM and CID prices to open a position on the DAM. To predict DAM prices, we propose using both technical indicator features and data augmentation methods, such as autoencoders and generative adversarial networks, together. While these methods have been applied separately by [5,6] to the forecasting of DAM prices, to the best of our knowledge they have yet to be used together in, for instance, ensemble models.

To predict CID prices, we propose using novel features, engineered from the limit order book, as inputs. To date, readily available features, such as lagged CID prices and seasonal features, have been used as inputs in CID price forecasting studies. However, detailed statistical information from the limit order book has not been tested.

For the CID and BAL, we propose developing an agent trained utilising A2C. The asynchronous version of A2C, A3C, has already been employed by [20]. Using A2C, however, is more cost-effective. To the best of our knowledge, we are the first to employ A2C in the context of SAT across the CID and BAL.

Note that to improve training stability further, unlike [20], we additionally use gradient clippings, different reward thresholds, additional behaviour cloning methods, more goal-based explorations, more flexible constraints and no early episode terminations. Moreover, our A2C agents start trading on the CID with a volume position of $\neq 0$ which is opened earlier on the DAM and a profit-and-loss of $\neq 0$ which is paid or received earlier on the DAM. This is in contrast to [20] which did not allow for DAM trading.

Summarising the contributions of the study, to the best of our knowledge, we are the first to:

- investigate the profitability of SAT across the DAM, CID, and BAL for a purely financial trader,
- predict DAM prices by combining technical indicators and data augmentation methods,
- utilise A2C to develop a risk-constrained arbitrage trading algorithm for the CID and BAL.

Outlining the structure of this paper, short-term electricity markets and arbitrage trading are described in Section 2. In Section 3, our rule-based trading method for the DAM is outlined. Section 3 details proposed forecasting methods for DAM and CID prices. In Section 4, our advantage actor–critic trading methods for the CID are introduced. The case study and results are presented in Section 5. Finally, in Section 6 we conclude and discuss potential methods for extending our research.

2. Arbitrage trading for electricity markets

2.1. The short-term electricity markets

There are a variety of ways to trade electricity: ranging from week-ahead or year-ahead maturities on futures markets to day-ahead maturities on the DAM to hour-ahead maturities on the CID. Fig. 1 presents trading timelines of the hourly electricity contracts, $h \in \{h_0, h_1, \dots h_{23}\}$, for the DAM and the CID. To participate in an auction-based DAM, orders must be placed before noon on day d. Hourly DAM clearing prices (in €/MWh) are calculated using a matching engine that aggregates all submitted orders, setting a price where supply and demand curves intersect. As a result, a single DAM price, p^{dam}, is set for each hourly electricity contract. More information about the DAM can be found in [3,33].

Starting from 15:00 on day d to one hour prior to physical deliveries on day $d+1$, orders can be continuously submitted to and cleared by the CID. The CID matching engine prioritises the best price and early submission. The best ask price p_t^a (in €/MWh) is the lowest price among available ask orders and the best bid price p_t^b (in €/MWh) is the highest price among available bid orders at time step $t \in [1, T]$, where T is the last time step of the CID trading session. Available quantities of the best ask order and the best bid order are q_t^a and q_t^b (in MWh) respectively. When a new ask (or bid) order is submitted, the CID engine matches orders immediately if the submitted ask (or bid) price is higher (or lower) than the best bid (or ask) price. Transaction price and quantity, which is the lower quantity between matched orders, are announced immediately. The remaining quantity of one of

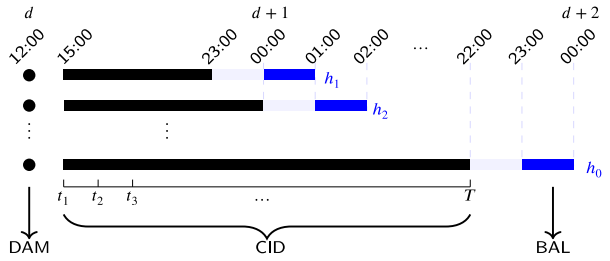


Fig. 1. The trading timelines for all hourly contracts. Blue bars are delivery periods. Black circles represent the auction-based DAM. Black bars are continuous-based CID trading sessions, starting at 15:00 on day $d+1$.

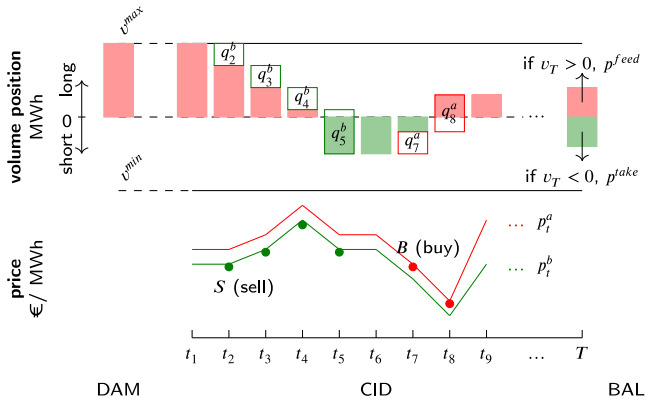


Fig. 2. An example of arbitrage trading for an example hourly contract h. The continuous trading timeline of CID is split into discrete time steps $t \in [1, T]$.

the matched orders is stored as an active order in the limit order book (LOB). If the submitted order has no match, it is stored as an active order in the LOB until it is matched or cancelled. The LOB consequently constitutes all active orders at time t.

When the above-mentioned CID engine matches orders, the trade book (TB) stores the traded price, quantity and timestamp for each hourly contract. At the end of a trading session, we can extract the volume weighted average price of trades, p^{vwap} (in €/MWh) calculated by $\sum_{k=1}^{K}(p_k \times q_k)/\sum_{k=1}^{T} q_k$, where K is the total number of transactions for a contract, and q_k and p_k are traded quantity and price for a transaction k. More information about the CID can be found in [34–36].

2.2. Arbitrage trading

By trading across the DAM, CID and BAL, arbitrage traders aim to maximise profits, *PnL*, without taking or unwinding long-term physical commitments. Diving into the decisions an arbitrager faces, a trader can open a DAM position v_0 (in MWh) either by buying of $v^{max} > 0$ or short selling $v^{min} < 0$, where v^{min} and v^{max} are pre-defined limits for short and long positions respectively. A trader pays $-v^{max} \times p^{dam}$ for a long position and receives $-v^{min} \times p^{dam}$ for a short position. This represents the cash received or spent on the DAM: C^{dam}.

Having opened a DAM position, an arbitrager can subsequently fully or partially close out the position on the CID. Elaborating, a trader can buy the best ask order (B), sell the best bid order (S), or hold (H) at any time step $t \in [1, T]$ during the CID trading session. Fig. 2, for example, shows a trader who starts with $v_1 = v^{max}$ - the open position brought forward from the DAM - and continuously trades on the CID, performing $\{H, S, S, S, S, H, B, B, H, \dots\}$. This trader pays $p_t^a \times q_t^a$ for buy decisions and receives $p_t^b \times q_t^b$ from sell decisions. The total cash paid or received by trading on the CID is C^{cid}. Note that the total arbitraged quantity is: $\sum_{t=1}^{T} q_t = min\{\sum_{t=1}^{T} q_t^a, \sum_{t=1}^{T} q_t^b\}$. Meanwhile, the outstanding quantity and final position is $\sum_{t=1}^{T} q_t^a - \sum_{t=1}^{T} q_t^b = v_T$.

When the CID trading window closes at time step $t = T$, any outstanding open position the arbitrager holds is automatically settled on the BAL. The hourly BAL purchase and selling prices, p^{take} and p^{feed} respectively, are computed by averaging four quarter-hourly BAL purchase and selling prices. The cash made on the BAL, C^{bal}, is calculated as $v_T \times p^{take}$ or $v_T \times p^{feed}$. More detailed information about the BAL can be found in [37,38].

Summarising the above, the profit *PnL* of the arbitrage trader is calculated according to (1).

$$PnL = C^{dam} + C^{cid} + C^{bal} - TC, \tag{1}$$

where

$$C^{dam} = \begin{cases} -v^{max} \times p^{dam}, & \text{if } v_0 > 0 \\ -v^{min} \times p^{dam}, & \text{otherwise} \end{cases}$$

$$C^{cid} = \sum_{t=1}^{T} p_t^b \times q_t^b - \sum_{t=1}^{T} p_t^a \times q_t^a$$

$$C^{bal} = \begin{cases} v_T \times p^{take}, & \text{if } v_T < 0 \\ v_T \times p^{feed}, & \text{if } v_T > 0 \\ 0, & \text{otherwise} \end{cases}$$

$$TC = 0.116 \times \left(\sum_{t=1}^{T} q_t^a + \sum_{t=1}^{T} q_t^b\right)$$

Note that TC is the trading cost assumed to be charged by a market operator. We assume that the arbitrage trader is charged €0.116 for trading 1 MWh of electricity by, for example, Nord Pool, a nominated electricity market operator. This parameter could be adjusted in future research to reflect higher or lower trading costs across other market operators.

3. Rule-based approach to day-ahead market trading

To open a position on the DAM, we implement a rule-based trading agent using forecasts of p^{dam} and p^{vwap}. If the forecast of p^{dam} is lower than the forecast of p^{vwap}, we open a long position, i.e buy v^{max} MWh. Otherwise, we open a short position, i.e. sell v^{min} MWh. The methods employed to predict DAM prices and CID prices, i.e. the volume-weighted average price of trades (vwap), are described below.

3.1. Forecasting day-ahead market prices

Following [5,6], two deep DAM forecasting models are evaluated. Elaborating, the forecasting accuracies of a two-layer neural network (2NN), consisting of two intermediate fully connected layers, and a joint three-layer network (2CNN_NN), consisting of two intermediate convolutional layers and a single intermediate fully connected layer, are assessed. Parameters and features, such as neighbouring country prices, are used as in [5,6]. L2 regularisation, ReLU activation functions and Adam are employed to improve training stability.

The aforementioned forecasting models are evaluated with technical indicator (TI) [5] feature inputs, and with the addition of augmented data [6]. Following [6] three augmentation methods using autoencoders (AE), variational encoders (VAE) and Wasserstein generative adversarial networks with a gradient penalty (GAN) are evaluated. Note that ensemble forecasts, obtained by averaging the forecasts from multiple methods are assessed as well. Eq. (2) is an example ensemble forecast:

$$\hat{p}^{2NN}_{AE+TI} = 1/2(\hat{p}^{2NN}_{AE} + \hat{p}^{2NN}_{TI}), \tag{2}$$

where $\hat{p}^{2NN}_{TI}$ is the 2NN forecast using TI features as inputs, and $\hat{p}^{2NN}_{AE}$ is the 2NN forecast obtained using augmented data generated by an autoencoder. To acquire the final DAM price forecasts, the model and method, yielding the best result, are selected.

3.2. Forecasting the volume-weighted average price of trades

CID prices, p^{vwap}, are predicted using machine learning models that take novel features as inputs. The features are generated by extracting statistical information from the LOB and TB.

3.2.1. Feature engineering

Information that captures significant price drivers for continuous CID trading can be extracted using the LOB of each hourly contract. Features, such as the total number of submitted ask/bid orders, the lowest best ask price, the highest best bid price, the first quantile of best ask/bid prices, the total ask/bid quantities, the first quantile of cumulative ask/bid quantities and the average mid-price for hour index-1/2/3/4, should capture relevant information required to accurately forecast CID prices.[1]

Using the TB of each hourly contract, information about trades can be extracted. Detailing example features, the highest traded price (high price), the lowest traded price (low price), the first traded price (open price), the last traded price (close price), the standard deviation of p^{vwap}, p^{vwap} of imports, p^{vwap} of exports, a binary feature showing whether a country heavily imports or exports (import), and a binary feature showing whether a country trades during hour index-1/2/3/4 (trade-1/2/3/4) can all be extracted.

Additionally, other possible price drivers can be obtained by adding exogenous and seasonal features to the input space. Exogenous features, for example, are forecasts of p^{dam}, wind speed and temperature. Seasonal features meanwhile include days of the week, delivery hours of the day, holidays, etc. Such categorical seasonal features are processed using one-hot encoding.

3.2.2. Feature selection

Feature selection commences by removing highly correlated variables. Specifically, features with a Pearson correlation higher than 0.8 are removed. This initial step speeds up run time and reduces the potential impacts of multicollinearity.

Known electricity market characteristics are accounted for by adding day-lagged and hour-lagged features. For instance, to forecast p^{vwap} of h^{d+1}_{16}, a fourteen-day-lagged period $h^{d-13}_{16}, \dots h^{d}_{15}$, h^{d}_{16}, h^{d}_{17} is considered for each feature.

The feature selection process is finalised by removing features with the lowest explanatory power from the data set using the least absolute shrinkage and selection operator (LASSO) [39].

3.2.3. Forecasting

The ability of regression models, such as LASSO, random forest (RF) [40], gradient boosting (GB) [41] and deep neural networks (DNN), to predict p^{vwap} is evaluated. Each model has a unique hyperparameter set to be optimised by minimising the forecast error across the validation set. For example, LASSO has the alpha and tolerance, RF has the number of estimators, and GB has the maximum depth. Similarly, DNN has the number of hidden layers, the size of these layers and the learning rate. Note that DNN models are constructed with fully connected layers, ReLU activation functions and Adam optimisers. Additionally, L2 regularisation is implemented to reduce the possibility of overfitting.

[1] Hour indices refer to specific time intervals of the trading session. For instance, hour index-1 is the last hour of the trading session, i.e. $[T-1\ h,\ T]$. Hour index-i is $[T-ihour,\ T-i+1hour]$. Moreover, the mid-price is calculated by averaging the best ask price and the best bid price. The average mid-price for hour index-i is calculated by averaging all mid-prices of hour index-i.

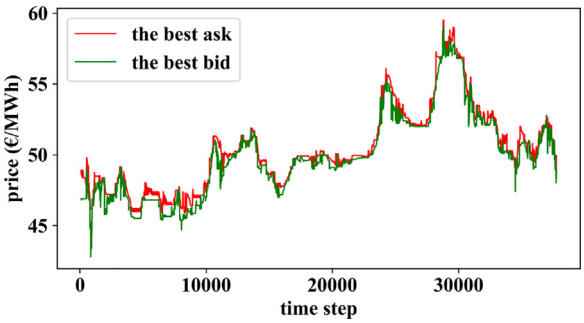


Fig. 3. The best ask/bid prices for h_{16} on 10/08/2020.

An ensemble method, which averages all evaluated forecasts, as shown in (3), is further considered.

$$\hat{p}^{ENSEMBLE} = 1/4(\hat{p}^{LASSO} + \hat{p}^{GB} + \hat{p}^{RF} + \hat{p}^{DNN}) \tag{3}$$

4. Advantage actor–critic approach to continuous intraday market trading

4.1. Background: Reinforcement learning

Following [42], below, the fundamentals of reinforcement learning (RL) are introduced. In the context of RL, a decision-maker is an agent, and a decision is an action. The state is a vector that encapsulates all available information about an environment. After obtaining a state measurement $\mathbf{s}_t$, an agent performs an action a_t at time step $t \in [1, T]$. The agent receives instantaneous scalar feedback in the form of a reward $r_{t+1} \in \mathbb{R}$. Starting from the initial state s_1 the above-described process is repeatedly performed until the termination state s_T is reached. This loop constitutes one episode (e).

A policy function defines the agent's behaviour, mapping states to actions: $\pi(a|\mathbf{s}) = \mathbb{P}[a_t = a \mid \mathbf{s}_t = \mathbf{s}]$. A stochastic policy function outputs a probability for each action. The agent decides whether to exploit — by choosing the action with the highest probability - or explore — by choosing a random action. An optimal policy is identified by maximising the total expected reward across a predefined number of episodes (e^{max}). The value function V_π for policy π is the expected total reward. Formally, under the policy π, $V_\pi(\mathbf{s}) = \mathbb{E}[R_t \mid \mathbf{s}_t = \mathbf{s}]$, where $R_t = r_{t+1} + \gamma r_{t+2} + \dots + \gamma^{T-1} r_T$ is the return and $\gamma \in [0, 1]$ is the discount factor. The policy yielding $\arg\max_\pi V_\pi$ is considered optimal.

4.2. Discrete time steps

In the existing literature, the interval Δt between t and $t+1$ is frequently defined using a fixed time window. In [43] a fixed 15-minute interval was used to discretise the continuous intraday trading period, while in [44] a 1-minute interval was used. In contrast with these studies, we avoid discretising $t \in [1, T]$ into fixed intervals. Instead, we follow a more flexible approach we previously developed in [20], using revision numbers from the LOB to discretise the trading period. A revision number is updated when the LOB changes, e.g. a new order is received. To reduce model training time, we only use revision numbers that change either the best ask price or the best bid price. Fig. 3 visualises the evolution of the best ask and bid prices for an example contract.

4.3. Actions

Actions a_t are defined over a discrete set. The action set is constrained to minimise risk. Firstly, the volume position v_t is restricted by the maximum allowed positions for short and long sides, v^{min} and v^{max} respectively. Following this strategy, if B is chosen, the agent buys $q_t^a \leq v^{max} - v_t$ MWh of electricity at time step $t \in [1, T]$. If S is chosen, the agent sells $q_t^b \leq -v^{min} + v_t$ MWh of electricity at time step t. Additionally, similar to [20], the total bought quantity until time step t, i.e. $\sum_{i=1}^{t} q_i^a$, and the total sold quantity until time step t, i.e. $\sum_{i=1}^{t} q_i^b$, are restricted by the maximum allowed total quantity q^{high}. Eq. (4) summarises our constrained action set.

$$a_t \in \begin{cases} \{H\}, & \text{if } \sum_{i=1}^{t} q_t^b = q^{high} \text{ and } \sum_{i=1}^{t} q_i^b = q^{high} \\ \{B, H\}, & \text{if } v_t = v^{min} \text{ or } \sum_{i=1}^{t} q_i^b = q^{high} \\ \{S, H\}, & \text{if } v_t = v^{max} \text{ or } \sum_{i=1}^{t} q_i^a = q^{high} \\ \{B, S, H\}, & \text{otherwise} \end{cases} \tag{4}$$

4.4. Rewards

Negative rewards motivate agents to learn how to avoid punishment. Similarly to [20], we employ negative reward functions to ensure the stability and efficiency of the learning algorithm. Note, however, that different reward thresholds are used. Additionally, we avoid using the end of contract reward as in [20]. Only three different reward functions are needed to span the action space of our agent.

Focusing on rewards for buying electricity, the immediate buy reward function is calculated by measuring the distance between a buy action – buying at p_t^a - and both the best and worst possible buy actions across a trading session – buying at the lowest and highest prices $p^{a_{low}}/p^{a_{high}}$. The buy reward function is bounded: $r_t^B \in [-2, 0]$. Eq. (5) formalises the buy reward function employed:

$$r_t^B = 1/2 \left(f^B(p^{dam}, p_t^a) + f^B(p^{b_{high}}, p_t^a) \right), \tag{5}$$

where

$$f^B(\tau^B, p_t^a) = \begin{cases} -1 - \left((p_t^a - \tau^B)/(p^{a_{high}} - \tau^B)\right), & \text{if } p_t^a > \tau^B \\ -\left((p_t^a - p^{a_{low}})/(\tau^B - p^{a_{low}})\right), & \text{otherwise} \end{cases}$$

and τ^B is the buy threshold separating gains and losses. A reward of $r_t^B = -1$, attained at τ^B, marks where profit can at best equal 0 over a trading session. When an agent purchases electricity for less than τ^B positive profits are attainable and $r_t^B > -1$. Equally, when the agent purchases electricity for more than τ^B a loss is guaranteed and $r_t^B < -1$. To separate the gains and losses of trading between the DAM and CID, we firstly consider $\tau^B = p^{dam}$. To separate the gains and losses of trading within the CID, we secondly consider $\tau^B = p^{b_{high}}$.

Similarly to the above, the immediate sell reward function is calculated by measuring the distance between a sell action – selling at p_t^b - and both the best and worst possible sell actions across a trading session – selling at the most expensive and least expensive prices $p^{b_{high}}/p^{b_{low}}$. The sell reward function is intrinsically bounded: $r_t^S \in [-2, 0]$. The more expensive the bid price at which the agent sells electricity the higher the sell reward. Eq. (6) formalises the sell reward function employed:

$$r_t^S = 1/2 \left(f^S(p^{dam}, p_t^b) + f^S(p^{a_{low}}, p_t^b) \right), \tag{6}$$

Table 1
Features encoding the state.

minutes to end of trading session
spread between p_t^b and p_t^a
spread between $(p_t^a + p_t^b)/2$ and its average forecast
spread between p_t^b and p^{dam}
spread between p_t^b and average p^b forecast
spread between p_t^b and p^{feed} forecast
spread between p_t^b and $1/6 \sum_{d-3}^{d-1} \sum_{h-2}^{h} p^{take}$
spreads between p_t^b and its lags $[p_{t-8}^b : p_{t-1}^b]$
best bid quantity q_t^b
number of bid orders
third (upper) quantile of cumulative bid quantities
best ask price p_t^a
first (lower) quantile of ask prices
second quantile (median) of ask prices
third (upper) quantile of ask prices
spread between p_t^a and $1/6 \sum_{d-3}^{d-1} \sum_{h-2}^{h} p^{take}$
best ask quantity q_t^a
total ask quantities
number of ask orders
categorical trade rule
scaled total bought quantity $\sum^t q_t^a / q^{high}$
scaled total sold quantity $\sum^t q_t^b / q^{high}$
scaled volume position $(v_t + v^{max})/(2 \times v^{max})$
scaled profit $(PnL - PnL^{low})/(PnL^{high} - PnL^{low})$

where

$$f^S(\tau^S, p_t^b) = \begin{cases} -2 + \left((p_t^b - p^{b_{low}})/(\tau^S - p^{b_{low}})\right), & \text{if } p_t^b < \tau^S \\ -1 + \left((p_t^b - \tau^S)/(p^{b_{high}} - \tau^S)\right), & \text{otherwise} \end{cases}$$

Sell thresholds of $\tau^S = p^{dam}$ and $\tau^S = p^{a_{low}}$ are considered to separate the gains and losses of trading between the DAM and CID, and within the CID.

Finally, the hold reward function is determined by quantifying the opportunity cost of a buy/sell action. Eq. (7) formalises the hold reward function.

$$r_t^H = \begin{cases} 0, & \text{if } r_t^B \,\&\, r_t^S < -1 \\ -1 - max\{r_t^B,\ r_t^S\}, & \text{otherwise} \end{cases} \tag{7}$$

Observe that the agent receives a hold reward of $r_t^H = 0$ when buy and sell actions lead to losses. The agent receives a hold reward $r_t^H \in [-1, 0)$ if either a buy action or a sell action is profitable. The more profitable the buy or sell action is, the lower the hold reward is.

4.5. States

The features presented in Table 1 are used to encode the state space. They are comparable to the features used in [20]. The categorical trade rule feature, in contrast with [20], uses p^{dam} in place of CID and BAL price forecasts. Formally, when $p_t^a < p^{dam}$ the feature takes the value buy. When $p_t^b > p^{dam}$ the feature takes the value sell. Otherwise, the feature takes the value hold.

4.6. Advantage actor–critic algorithms

In the context of actor–critic (AC) algorithms, a DRL agent is an AC worker, the value is updated by the critic and the policy is updated by the actor using the critic's feedback [42]. The AC algorithm uses a single global network but multiple AC workers, i.e. $w \in [1, W]$, where W is the number of local networks. AC workers update the global network asynchronously in A3C, whereas synchronously in A2C. Each AC worker collects different experiences by independently interacting with the environment. Using multiple workers results in a greater exploration of the state space.

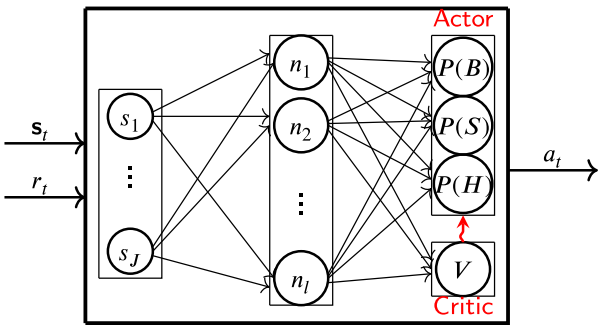


Fig. 4. An example AC worker, $w \in [1, W]$.

In this study, each actor is stochastic and approximated by deep neural networks that use softmax functions in the output layer. Fig. 4 shows an example AC worker. For simplicity, a one-layer neural network is visualised. Note that in practice, hyper-parameters, such as the number of hidden layers L and the number of neurons n_l for each hidden layer $l \in [1, L]$, are optimised.

Algorithm 1 Pseudocode for each AC worker.

Require: hyper-parameters, γ_{start}, γ_{end}, ϵ_{start}, ϵ_{end}, t^{max}, e^{max}, β; global parameters, θ_π, θ_v; thread parameters, $\theta_\pi^{'}, \theta_v^{'}$; global counter, $e \leftarrow 1$; and $t_1 \leftarrow 1$
Ensure: $d\theta_\pi$, $d\theta_v$
1: **while** $e < e^{max}$ **do**
2: reset gradients $d\theta_\pi \leftarrow 0$, $d\theta_v \leftarrow 0$
3: synchronise thread parameters $\theta_\pi^{'} = \theta_\pi$, $\theta_v^{'} = \theta_v$
4: calculate episode index $e_i = 1 - (e/e^{max})$
5: calculate discount $\gamma = (\gamma_{start} - \gamma_{end}) * e_i + \gamma_{end}$
6: calculate epsilon $\epsilon = (\epsilon_{start} - \epsilon_{end}) * e_i + \epsilon_{end}$
7: $t = t_1$
8: get state $\mathbf{s}_t$
9: **while** $\mathbf{s}_t \neq \mathbf{s}_T$ **and** $t - t_1 < t^{max}$ **do**
10: constrain the action space
11: **if** $e = 1$ **then**
12: clone behaviour a_t
13: **else**
14: **if** $\epsilon_{random} < \epsilon$ **then**
15: **if** $t_{random} < t < t_{random} + t^{max}$ **then**
16: clone behaviour a_t
17: **else**
18: explore: *random* a_t according to $\pi(a_t|\mathbf{s}_t; \theta_\pi^{'})$
19: **end if**
20: **else**
21: exploit: *max* a_t according to $\pi(a_t|\mathbf{s}_t; \theta_\pi^{'})$
22: **end if**
23: **end if**
24: receive reward r_{t+1} and new state $\mathbf{s}_{t+1}$
25: $t \leftarrow t + 1$
26: **end while**
27: $R_\pi = \begin{cases} 0, & \text{for terminal } \mathbf{s}_T \\ V_\pi(\mathbf{s}_t, \theta_v^{'}), & \text{for non-terminal } \mathbf{s}_t \end{cases}$
28: **for** $i \in \{t-1, \dots t_{start}\}$ **do**
29: $R_\pi \leftarrow r_i + \gamma R_\pi$
30: $A_\pi \leftarrow R_\pi - V_\pi(\mathbf{s}_i, ; \theta_v^{'})$
31: $d\theta_\pi \leftarrow d\theta_\pi + \beta \ \nabla_{\theta_\pi^{'}} \log \pi(a_i \,|\, \mathbf{s}_i \,; \theta_\pi^{'})(A_\pi)$
32: $d\theta_v \leftarrow d\theta_v + \beta \ \partial(A_\pi)^2 / \partial\theta_v^{'}$
33: **end for**
34: $e \leftarrow e + 1$
35: update θ_π using $d\theta_\pi$, and θ_v using $d\theta_v$
36: **end while**

Algorithm 1 formalises the update procedure of each AC worker. Describing the algorithm, firstly, the AC worker resets gradients and synchronises thread parameters (lines 2 and 3). The worker then calculates the episode index (line 4), ascending discount rate (line 5) and descending epsilon (line 6). Subsequently, it receives its first state (line 8). Next, the worker interacts with the environment and collects experiences (lines 9 to 26). The action space is constrained following Section 4.3 (line 10). Each worker clones behaviours during the first episode. It also clones behaviours t^{max}/T of the time over the rest of the episodes (lines 12 and 16). Note that to spur exploration, each worker clones different behaviours. Elaborating, the first worker, w_1, clones (8), to learn to effectively arbitrage between the DAM and the CID.

$$a_t = \begin{cases} B, & \text{if } p_t^a < p^{dam} \ \& \ v_t < v^{max} \ \& \ \sum_{i=1}^{t} q_i^a < q^{high} \\ S, & \text{if } p_t^b > p^{dam} \ \& \ v_t > v^{min} \ \& \ \sum_{i=1}^{t} q_i^b < q^{high} \\ H, & \text{otherwise} \end{cases} \tag{8}$$

The second worker, w_2, meanwhile clones (9) to learn to arbitrage well within the CID.

$$a_t = \begin{cases} B, & \text{if } p_t^a < p^{b_{high}} \ \& \ v_t < v^{max} \ \& \ \sum_{i=1}^{t} q_i^a < q^{high} \\ S, & \text{if } p_t^b > p^{a_{low}} \ \& \ v_t > v^{min} \ \& \ \sum_{i=1}^{t} q_i^b < q^{high} \\ H, & \text{otherwise} \end{cases} \tag{9}$$

The third worker, w_3, clones (10) to arbitrage well between the CID and the BAL.

$$a_t = \begin{cases} B, & \text{if } p_t^a < p^{feed} \ \& \ v_t < v^{max} \ \& \ \sum_{i=1}^{t} q_i^a < q^{high} \\ S, & \text{if } p_t^b > p^{take} \ \& \ v_t > v^{min} \ \& \ \sum_{i=1}^{t} q_i^b < q^{high} \\ H, & \text{otherwise} \end{cases} \tag{10}$$

Finally, the rest of the workers, $w \in [4, W]$, clone (11) with different reward thresholds $\tau \in [-1, 0)$ to learn to reach higher rewards.

$$a_t = \begin{cases} B, & \text{if } r_t^B > \tau \ \& \ v_t < v^{max} \ \& \ \sum_{i=1}^{t} q_i^a < q^{high} \\ S, & \text{if } r_t^S > \tau \ \& \ v_t > v^{min} \ \& \ \sum_{i=1}^{t} q_i^b < q^{high} \\ H, & \text{otherwise} \end{cases} \tag{11}$$

Following the decayed epsilon greedy exploration method, the worker explores $\epsilon - (t^{max}/T)$ of the time and exploits $1 - \epsilon$ of the time (lines 18 and 21). The worker collects experiences until the number of time steps to update the global network t^{max} or the last state in the training set $\mathbf{s}_T$ is reached. Using collected experiences, the critic estimates the value of a state (line 27). The advantage function [10] A_π is calculated by the difference between the estimated value of this state and the value of the state–action (line 30). The actor and critic are updated using A_π (lines 31 and 32). Consequently, the global network is updated asynchronously for A3C and synchronously for A2C (line 35). Note that the smooth L1 loss and gradient clipping are implemented to avoid exploding gradients. Updating process continues until a predefined maximum number of episodes e^{max} is reached.

Table 2
Summary statistics for DAM prices in €/MWh.

	Mean		Standard deviation	
	Train	Test	Train	Test
Belgian DAM	43.95	31.04	22.93	16.38
Dutch DAM	41.31	31.46	14.67	15.08

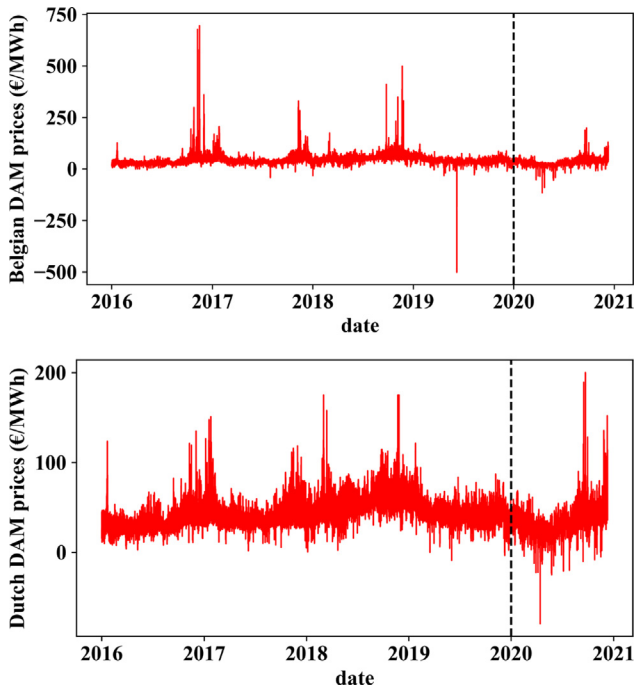


Fig. 5. Historic DAM prices, covering the training and test periods. The dashed black line represents the training/test split.

5. Numerical case study

5.1. Forecasting day-ahead market prices

The data gathering and processing steps employed in the forecasting of DAM prices are outlined below. Additionally, the evaluation procedure is described, and the DAM forecasting results are analysed.

5.1.1. Data

Dutch and Belgian DAM prices, spanning from 01/01/2016 to 11/12/2020, are collected together with load and generation day-ahead forecasts from the ENTSO-E Transparency Platform [45]. This data is subsequently split into training and test data. Data from 01/01/2016 to 01/01/2020 are used for training, while data from 01/01/2020 to 11/12/2020 are used for testing. Note that hyperparameter optimisation is not performed. Instead, features and optimised parameters from [5,6] are used directly. No validation set is thus required.

Training and testing DAM prices are presented in Fig. 5. Summary statistics of DAM prices for the training and test sets are shown in Table 2. From Table 2 we discern that Belgian DAM prices in the test set have a lower standard deviation than historic DAM prices. A lack of outliers across the Belgian DAM test set, as Fig. 5 highlights, explains this observation.

Analysing train and test set differences further, note that Table 2 additionally highlights differences in mean DAM prices. The means of Belgian and Dutch DAM prices are higher across the training set than the test set.

Table 3
MAE results on the train set for DAM price forecasting methods.

		AE	VAE	GAN	TI
Dutch	2NN	9.84	9.79	9.86	5.28
	2CNN_NN	10.18	10.21	10.04	5.31
Belgian	2NN	13.68	13.83	13.60	8.32
	2CNN_NN	13.52	13.51	13.06	7.93
average		11.81	11.84	11.64	6.71

5.1.2. Data processing

Min–max scaling is applied across a 30-day rolling window to scale features. A 30-day window is selected to counteract market seasonalities.

5.1.3. Modelling

Per Section 3.1, the forecasting accuracy of two neural network architectures (2NN and 2CNN_NN) is evaluated. The models are trained with TI features, and with augmented data separately.

Detailing the TI training process, the best-performing TIs, as identified by [5], are assessed. Specifically, the exponential moving average (EMA) indicator, with a span of 22 days, is used as a 2NN input. Meanwhile, the rate of change (ROC) indicator, with a 9-day lag factor, is used as a 2CNN_NN input. Specifying model hyperparameters, following [5], a 2NN with [500, 250] neurons is assessed. A 2CNN_NN employing 32 filters, a (1, 3) kernel, and 123 neurons in its NN layer, is additionally evaluated. A dropout rate of 0.25 and a learning rate of 0.001 are also employed in both the 2NN and 2CNN_NN.

Detailing the augmentation training process, the best-performing augmentation models, as identified by [6], are evaluated. Note that hyperparameters are set separately for each DAM contract. For instance, when forecasting the Belgian 20 h, a 2NN with [128, 128] neurons is assessed. Across all hourly forecasting models, on average a 2NN with [270, 254] neurons, a dropout rate of 0.01, and a learning rate of 0.001 is assessed. Similarly, on average a 2CNN_NN, with 40 filters and a (1, 3) kernel in its first layer, 52 filters and a (4, 4) kernel in its second layer, and 140 neurons in its final layer, is assessed. On average a learning rate of 0.0009 and a dropout rate of 0.01 are used with the 2CNN_NN.

5.1.4. Software

Several Python libraries are used in the forecasting of DAM prices. All deep neural networks, for instance, are implemented using `Keras`.

5.1.5. Evaluation

The mean absolute errors (MAE) are used to evaluate forecasting accuracy. Additionally, to facilitate relative evaluation, benchmark forecasts are computed according to (12):

$$\hat{p}_{d+1}^{\text{DAM-BENCH}} = 1/2(p_d^{dam} + p_{d-1}^{dam}). \tag{12}$$

The benchmark is a two-day moving average of DAM prices.

5.1.6. Results and discussion

The forecasting accuracies of evaluated models on the training and test sets are summarised in Table 3 and Table 4 respectively. Data augmentation and TI methods show similar performance across the training and test sets. Note that slight deviations are expected since the summary statistics of historic prices are not the same across the training and test sets, as shown in Table 2.

Analysing the test results from Table 4, the benchmark, DAM-BENCH, is observed to yield the highest average MAE of 8.07. Meanwhile, TI is observed to yield an average MAE of 7.56;

Table 4
MAE results on the test set for DAM price forecasting methods.

		AE	VAE	GAN	TI	AE+VAE	AE+GAN	VAE+GAN	AE+TI	VAE+TI	GAN+TI	AE+VAE+GAN	AE+VAE+GAN+TI	DAM-BENCH
Dutch	2NN	5.70	5.72	5.78	6.54	5.58	5.60	5.61	5.54	5.55	5.54	5.56	5.37	7.75
	2CNN_NN	5.62	5.69	5.65	7.61	5.53	5.52	5.55	6.04	6.12	6.08	5.49	5.62	
Belgian	2NN	7.27	7.04	6.89	8.15	6.71	6.59	6.47	6.55	6.37	6.42	6.40	6.14	8.38
	2CNN_NN	7.28	7.33	7.23	7.96	6.81	6.84	6.80	6.57	6.48	6.56	6.63	6.24	
average		6.47	6.45	6.39	7.57	6.16	6.14	6.11	6.18	6.13	6.15	6.02	5.84	8.07

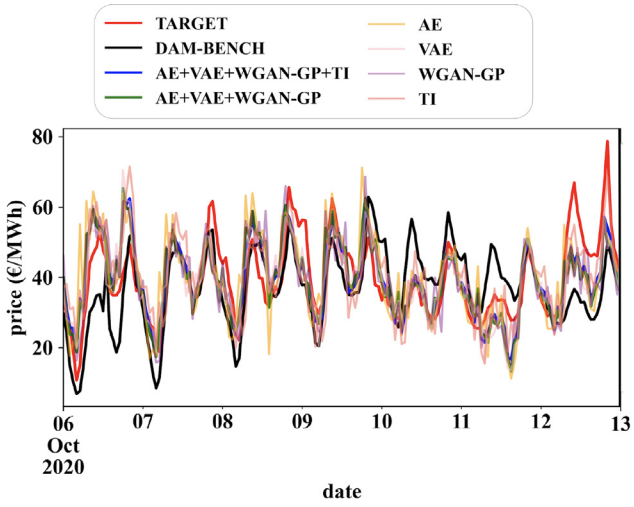


Fig. 6. An example showing Belgian DAM 2NN forecasts spanning a week: from 06/10/2020 to 13/10/2020.

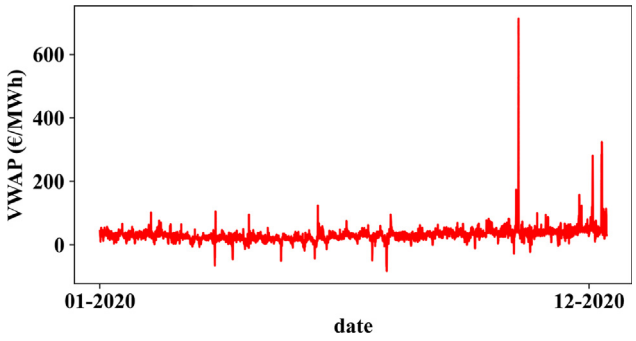


Fig. 7. Historic vwap.

6.20% lower than DAM-BENCH. Improving forecasting accuracies further, data augmentation methods, namely AE, VAE, and GAN, are observed to yield average MAEs of 6.47, 6.45, and 6.39 respectively; up to 20.82% lower than DAM-BENCH. The high performance of data augmentation methods highlights the importance of the training set size in reducing the generalisation error of DAM forecasts.

Combining forecasts from the above methods boosts accuracies even further. For example, AE+GAN, which generates forecasts by averaging AE and GAN forecasts, yields an average MAE of 6.14. This is 5.10% and 3.91% lower than AE and GAN methods respectively. Overall, AE+VAE+GAN+TI is observed to yield the lowest average MAE of 5.84. The ensemble method, using both TI features and data augmentation, outperforms TI by 22.85%, and AE, VAE, and GAN on average by 9.27%.

To better understand the summary results, Fig. 6 displays forecasts generated using the evaluated methods. While DAM-BENCH forecasts are observed to be a lagging indicator, reflecting price patterns and levels from previous days, other evaluated methods are observed to yield more accurate leading forecasts. Fig. 6 highlights how combining predictions, by averaging slightly overestimated and underestimated prices, increases the overall accuracy of DAM forecasts.

Note that, given the findings of [5,6], we postulate that augmentation helps in forecasting the bulk of the distribution, by reducing overfitting. Meanwhile, TI, which introduces additional inputs, helps in forecasting the tails of the distribution by allowing the models to better segment the n-dimensional feature space. Averaging captures both these effects; improving forecast accuracies across the entire distribution.

5.2. Forecasting the volume-weighted average price of trades

The data gathering and processing steps employed in the evaluation of vwap forecasts are outlined below. The feature selection and modelling procedures are described. Finally, the performance of forecasting models is compared and analysed.

5.2.1. Data

Data spanning from 01/01/2020 to 11/12/2020 is gathered for vwap forecasting. LOB and TB data for the European single intraday coupled market is obtained from Scholt Energy [46], an energy supplier. Forecasts, such as forecasts of wind speed and solar irradiance, are also queried from Scholt Energy. Engineered features are generated following the methods described in Section 3.2.

The historic vwap is presented in Fig. 7. The mean and standard deviation of the vwap are found to be 30.45 €/MWh and 24.35 €/MWh respectively.

Describing the training, validation, and testing procedures, given a need to accumulate forecasts for the entire dataset (from 01/01/2020 to 11/12/2020) an iterative procedure is employed to obtain test forecasts for all 49 weeks of inputs. In one iteration, the available data is split into roughly 48 weeks of training and 1 week of test data. 48 fold cross-validation is subsequently employed across the training data; yielding an optimised model, which is used to generate vwap forecasts for the test data. This process is repeated 49 times until forecasts are obtained for the entire dataset.

5.2.2. Data processing

Min–max scaling is employed to ensure the comparability of inputs.

5.2.3. Feature selection

Firstly, highly correlated features with a correlation higher than 0.8 are removed. Next, LASSO is used to select features from the remaining pool of uncorrelated features. On average, 37 features are selected.

Following the above method the most frequently selected LOB features are: the total number of submitted bid orders, the total number of submitted ask orders, the total number of revisions, the lowest best ask price, and the average mid price for hour index-4. The most frequently selected TB features are: the p^{vwap} for hour index-3, the p^{vwap} of exports, the standard deviation of p^{vwap}, the binary feature import, trade-1, trade-2, trade-3, trade-4, the open price, the low price, and the spread between high and low prices. The most frequently selected exogenous features are: the wind speed forecast, the solar irradiance forecast, the precipitation forecast, and the p^{dam} forecast. The most frequently selected seasonal features are dummy variables

Table 5
MAE results on the test set for vwap forecasting methods.

	LASSO	GB	RF	DNN	ENSEMBLE	VWAP-BENCH
Dutch	8.27	8.22	8.58	7.78	7.59	12.25

for holidays, months, days of the week, and delivery hours of the day. Fourteen-day-lagged, seven-day-lagged, two-day-lagged and one-day-lagged LOB, TB, and exogenous features are often selected; combating electricity market seasonality.

5.2.4. Modelling

Four individual models, namely LASSO, RF, GB, and DNN, and one ensemble model are evaluated. The hyperparameters of the individual models are optimised using 48-fold cross-validation. On average, cross-validation selects an alpha of 0.21 and tolerance of 0.06 for LASSO. 115 estimators are most frequently selected by RF. A maximum depth of 6 is most frequently selected by GB. A learning rate of 0.007, and 2 hidden layers of size 272 and 408 are on average selected by DNN.

5.2.5. Software

Several Python libraries are used to implement feature selection and obtain vwap forecasts. For instance, the `scikit-learn` library is used to perform LASSO feature selection, and forecast vwap prices using machine learning models.

5.2.6. Evaluation

As with DAM prices, the MAE is used to evaluate the accuracy of vwap forecasts. Furthermore, a two-day moving average, calculated according to (13), is computed to facilitate a relative evaluation of forecast accuracies.

$$\hat{p}_{d+1}^{\text{VWAP-BENCH}} = 1/2(p_d^{vwap} + p_{d-1}^{vwap}) \tag{13}$$

5.2.7. Results and discussion

Table 5 presents the forecast accuracy of the evaluated models. Overall, VWAP-BENCH is found to yield the highest MAE of 12.96. LASSO, GB, and RF meanwhile yield MAEs of 8.27, 8.22, and 8.58 respectively; outperforming VWAP-BENCH by roughly 30.00%. DNN obtains a MAE of 7.78; outperforming VWAP-BENCH by 36.49%. ENSEMBLE, which takes the average of all model forecasts, further improves upon forecast accuracies; yielding the lowest MAE of 7.59.

The results presented above can be further understood by evaluating the example forecasts shown in Fig. 8. From the example, we observe that DNN more successfully approximates the non-linearities in p^{vwap}; capturing the complex signals embedded in the time series.

5.3. Trading on the day-ahead market

AE+VAE+GAN+TI forecasts of p^{dam} are utilised along with ENSEMBLE forecasts of p^{vwap} by our ruled-based DAM trading agent. Defining the agent's strategy, a long position is opened on the Dutch DAM, $v^{max} = 10$, if $\hat{p}^{\text{2NN}}_{\text{AE+VAE+GAN+TI}} < \hat{p}^{\text{ENSEMBLE}}$. Otherwise, a short position is opened: $v^{min} = -10$. Dutch and Belgian AE+VAE+GAN+TI forecasts are averaged to obtain $\hat{p}^{\text{2NN}}_{\text{AE+VAE+GAN+TI}}$. Averaging DAM forecasts of neighbouring countries improves the performance of the rule-based trading agent. We postulate that this occurs because the CID market provides quotes of pan-European electricity prices. Taking an average of DAM prices allows us to identify profitable opportunities arising between neighbouring market areas.

Table 6 presents a confusion matrix summarising the performance results of the DAM trading strategy. Following the rule-based trading strategy, the direction of price movements is correctly predicted 73.97% of the time. As opened DAM positions are used as the initial CID positions, this accuracy is important for the performance of the upcoming CID trading agent and the final profit. The lower the accuracy the harder for the CID agent to yield a profit. For example, assuming no trade is made on the CID and all opened DAM positions are closed out at the BAL, an agent with an accuracy of 0% would yield a loss of €406419.42, while an agent with an accuracy of 100% would return a profit of €231842.07. Our ruled-based DAM trading agent with an accuracy of 73.97% would return a profit of €678.07. Note that this agent will be evaluated as a benchmark, called HOLD, in the next section.

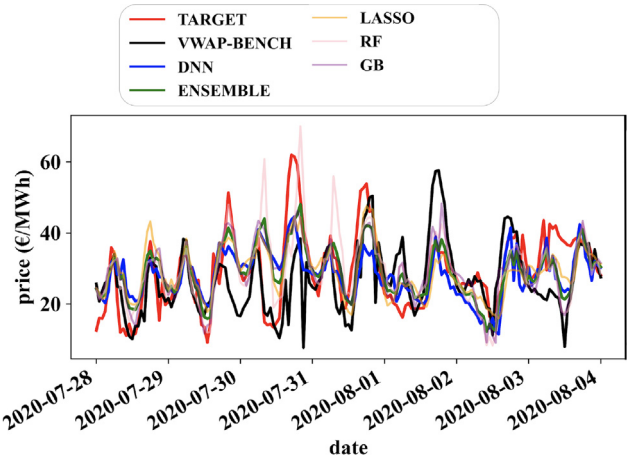


Fig. 8. An example showing Dutch CID vwap forecasts spanning a week: from 28/07/2020 to 04/08/2020.

Table 6
Confusion matrix across 2020 for the ruled-based DAM agent.

		Targets	
		True	False
Predictions	True	1781	665
	False	737	2203

5.4. Trading on the continuous intraday market

In this section, the data gathering and processing steps employed in developing agents for CID trading are outlined. Further, our actor–critic hyperparameters are specified. The evaluation procedure is described. And finally, statistical arbitrage trading results are analysed.

5.4.1. Data and data processing

Data spanning 2020 – from 01/01/2020 to 11/12/2020 – are collected from Scholt Energy [46]. In total the order-books of 8280 contracts are obtained. 4148 contracts are used for training and 1760 contracts for testing. The remaining contracts are excluded from the study for failing data quality tests. The primary test checks whether the data provider stores orders in the limit order book. If a contract has an empty limit order book, it is excluded from our study.

Note that because optimised parameters from [20] are used, no contract is ascribed to a validation set. A rolling window is used to continuously train and evaluate AC workers every month. Finally, min–max scaling is utilised to scale states.

5.4.2. Advantage actor–critic algorithms

The RL environment is configured following Section 4. Using hyper-parameters from [20], we set: $W = 8$, $\epsilon^{start} = 0.9$, $\epsilon^{end} = 0.01$, $\gamma^{start} = 0.29$, $\gamma^{end} = 0.9999$, $t^{max} = 2906$, $\beta = 0.003$, $L = 2$, $n_1 = 216$ and $n_2 = 193$. The chosen neural network architecture is a two-headed shared network. A tanh activation function is used in the hidden layers. Adam is selected as an optimiser.

To scale our states, we set $PnL^{low} = -5000$ and $PnL^{high} = 10000$. To constrain our action space, we use a more flexible approach than [20]. While [20] sets $q^{high} = 40$, $v^{max} = 3$, and $v^{min} = -3$, we set $q^{high} = 50$, $v^{max} = 10$, and $v^{min} = -10$. While [20] additionally constrains the maximum allowed buying/selling quantity at each time step t to 1 MW, we avoid such a fixed constraint.

For every month of the training set, AC workers are trained until $e^{max} = 100$. Eight workers are thus trained across 800 episodes. At the end of each episode e, we calculate the performance: i.e. the total reward accumulated across the training window. Model weights are saved whenever performance improves. The last saved model is used on the test set.

Training times vary across the rolling windows due to varying monthly contract counts. For a single month, $1-2$ days are required to train AC agents using a GeForce GTX 1080. Once trained, the trading agent can however execute a decision immediately.

5.4.3. Software

Several Python libraries are used to train and evaluate our AC workers. For instance, the `gym` library is used to build custom trading environments for reinforcement learning and the `PyTorch` library is used to build and optimise deep neural networks of AC workers, as visualised in Fig. 4.

5.4.4. Evaluation

Both profit (*PnL* and *PT*) and risk-reward (*PD*) metrics are used to evaluate agents' test performances. The *PT* is calculated by dividing the *PnL* by the total traded quantity. Formally, $PT = \sum_{n=1}^{N} PnL_n / \sum_{n=1}^{N}\sum_{t=1}^{T} q_t$, where N is the total number of contracts in the test set. The *PD*, measuring the profit per unit of risk, is calculated as $\sum_{n=1}^{N} PnL_n/\sigma$, where σ is the standard deviation of the *PnL*. The higher *PnL*, *PT*, and *PD* the better the trading algorithm.

To contextualise the performances of our intelligent agents, two rule-based benchmarks are evaluated as well. The first, intended to gauge the minimal attainable profit from arbitrage trading on short-term markets, is the HOLD benchmark. The HOLD closes out all open DAM positions on the BAL; no trade is executed on the CID. The second benchmark (PRE-BA) follows the rules specified in (14):

$$a_t = \begin{cases} B, \text{if } p_t^a < 1/30 \times \sum_{i=t-30}^{t} p_i^a \;\&\; v_t < v^{max} \;\&\; \sum q^a < q^{high} \\ S, \text{if } p_t^b > 1/30 \times \sum_{i=t-30}^{t} p_i^b \;\&\; v_t > v^{min} \;\&\; \sum q^b < q^{high} \\ H, \text{otherwise} \end{cases} \tag{14}$$

PRE-BA uses the previous best bid and best ask prices to place trades. It is developed to highlight the risk associated with CID trading.

5.4.5. Results and discussion

Table 7 and Fig. 9 present the test results, spanning 1760 contracts, for all evaluated trading agents. Elaborating Table 7 summarises the test results; showing the total traded arbitraged quantity (quantity sum), the average traded arbitraged quantity per contract (quantity mean), the total/cumulative profit-and-loss (*PnL* sum), the average profit-and-loss per contract (*PnL* mean), the percentage of contracts with positive *PnL* over the test period (*PnL*>0), the profit-to-deviation (*PD*), and the profit-to-trade (*PT*). Fig. 9, meanwhile, displays the cumulative *PnL* of the evaluated agents across the test set.

Table 7
Traded quantity, profit and risk results on the test set.

	Quantity (in MWh)		*PnL* (in €)				
	Sum	Mean	Sum	Mean	*PnL*>0	*PD*	*PT*
A2C	33805.10	19.21	97853.69	55.60	61%	190.66	2.90
A3C	29571.70	16.80	89248.52	50.71	62%	174.06	3.02
PRE-BA	52070.50	29.59	1586.07	0.90	51%	3.21	0.03
HOLD	17600.00	10.00	1395.20	0.79	52%	2.83	0.08

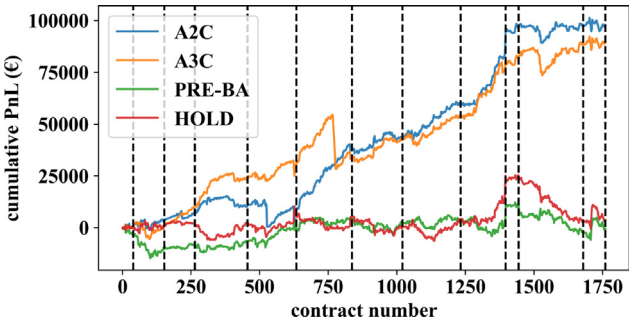


Fig. 9. Cumulative *PnL* across test contracts. Monthly rolling windows are shown with dashed black lines.

Analysing the summary results presented in Table 7, HOLD is observed to yield the lowest *PnL* and *PD*, and the second-lowest *PT*. Only 52% of traded test contracts return positive profits following HOLD. Evaluated across test set contracts, Fig. 9 shows the fluctuations in HOLD revenues. The cumulative *PnL* is almost flat at around 0. By not placing trades on the CID, HOLD exposes the importance of the CID in arbitrage trading.

Analysing the overall performance of PRE-BA, the second benchmark spurs the highest execution of trades. 52070.5 MWh of electricity is traded by PRE-BA. Despite this, PRE-BA yields the second-lowest *PnL* and *PD*, and the lowest *PT*. Evaluated across test set contracts, Fig. 9 exposes persistently low revenues. Similarly to HOLD, the cumulative *PnL* fluctuates around 0. PRE-BA highlights the risks of trading too frequently on the CID.

Turning to AC strategies, from the results in Table 7 we observe that 62% of traded contracts yield positive profits following A3C: 10% more contracts than HOLD and 11% more than PRE-BA. Fig. 9 further shows that A3C manages to steadily increase the cumulative *PnL*. A3C generates 6319% more profit per contract than HOLD and 5535% more than PRE-BA. Note, however, that extra profit is not generated by accepting disproportionately more risk. This is highlighted by A3C's higher *PD*.

Comparing the results relative to [20], A3C also yields greater returns and a higher *PD* than the A3C implemented in [20]. Across the same test set, the A3C in [20] yields: a total *PnL* of €19927.22 with an average *PnL* of €11.32 per contract, a *PD* of 142.80, and a *PT* of 2.84. Relative to these results, our A3C generates 348% more profit, a 22% higher *PD*, and a 6% higher *PT*. Relaxing trading quantity constraints and using the DAM in arbitrage trading thus appears to increase the profit and reward-risk ratios.

Finally analysing A2C results, Table 7 shows that A2C trades: 92% more MWh per contract than HOLD, 35% less than PRE-BA, and 14% more than A3C. A2C yields the highest profit per contract: 6938% greater than HOLD, 6078% greater than PRE-BA, and 10% greater than A3C. Evaluated across test set contracts,

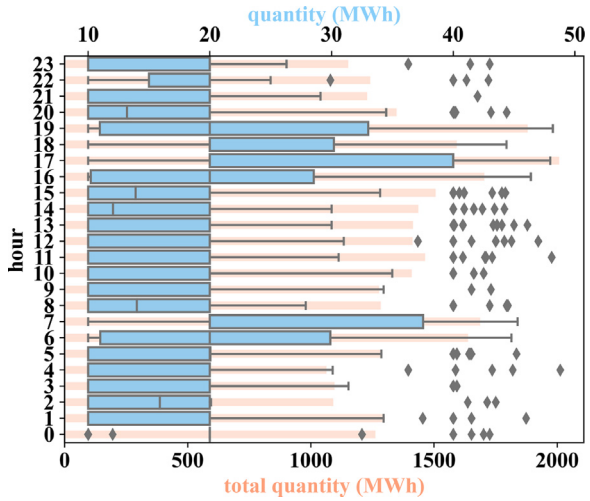


Fig. 10. Traded quantity distribution (blue) and total quantity (salmon) of A2C across the test set for each delivery hour.

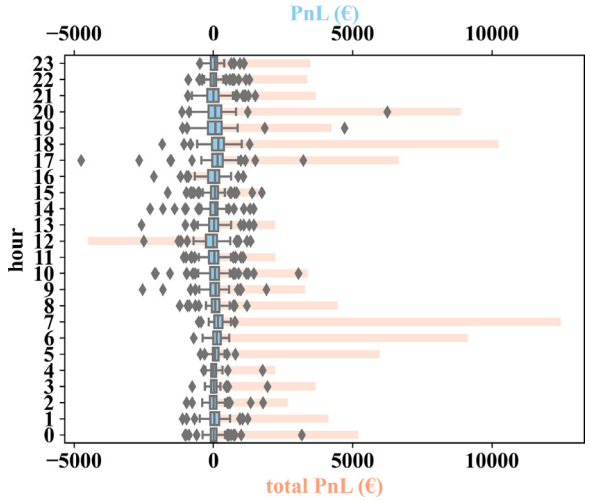


Fig. 11. *PnL* distribution (blue) and total *PnL* (salmon) of A2C across the test set for each delivery hour.

Fig. 9 shows that A2C gradually and broadly consistently increases revenues; surpassing A3C total revenues by the final test contract. Despite A3C having a marginally higher *PT*, evaluated using the *PnL* and *PD*, A2C is found to be the best arbitrage trading agent across short-term markets.

Note that despite their significant positive cumulative *PnL*s, it must be highlighted that neither A3C nor A2C offers a free lunch. Analysing changes in the cumulative *PnL* across the test set, Fig. 9 exposes periods of A3C and A2C underperformance. Elaborating, around contract 750, for example, A3C consecutively yields losses (9), precipitating a significant decline in the cumulative *PnL*. Similarly, A2C yields consecutive losses (2) around contract 550. Losses result from a build-up of unprofitable positions that the algorithms are forced to close out either on the CID or BAL. To highlight both the potential risks and rewards associated with the most profitable strategy, below A2C trades are further analysed.

A2C performance across delivery hours. Analysing A2C results further, in Fig. 10 the box plot shows the distribution of traded quantities, while the bar graph shows the total quantity of executed trades for each delivery hour. Across a majority of hours, A2C trades between 10 and 20 MWh per contract. The total traded quantity across these hours is between 1000 and 1500 MWh. Across a minority of hours, A2C trades between 20 and 40 MWh. This occurs for h_7, h_{17} and h_{18}. The total traded quantity is also high for these hours, more than 1500 MWh, in line with expectations. A2C trades more during day delivery hours than night hours when liquidity is typically lower.

Finally, Fig. 11 shows that the *PnL* distribution of A2C is centred around 0 for each delivery hour. The total *PnL*, however, is positive for all delivery hours except h_{12} and h_{16}. Night hours are less volatile than day hours and bring around €4000. Day hours bring roughly between €2000 and €12500. More specifically, hours $h_{11} - h_{16}$ are less profitable, whereas hours $h_5 - h_7$ and $h_{17} - h_{20}$ are significantly more profitable. Considering day hours are more liquid than night hours, A2C thus manages to exploit the liquidity across day hours – by trading more frequently and bringing more profit – without receiving any trading rule regarding day or night trading.

6. Conclusion and future work

In this paper, arbitrage trading agents capable of trading across the day-ahead (DAM), continuous intraday (CID), and balancing (BAL) markets were developed and evaluated. A rule-based trading method, using forecasts of DAM prices and CID volume-weighted average price of trades (vwap), was developed to open positions on the DAM. DAM prices were predicted utilising technical indicators (TI) and data augmentation methods, such as autoencoders, variational autoencoders, and Wasserstein generative adversarial network with a gradient penalty. Vwap, meanwhile, was predicted using an ensemble model; taking engineered features from the limit order book and trade book as inputs. Using the above forecasts, 74% of positions were accurately opened across the DAM following our rule-based trading method.

Focusing on the CID and BAL, a deep reinforcement learning (DRL) agent, employing the synchronous advantage actor–critic algorithm (A2C), was trained. Behaviour cloning, i.e. goal-based exploration, was employed to increase the performance of the agent. A two-headed shared deep neural networks was used to determine the agent's policy. The performance of the agent was compared against three benchmark policies: HOLD, PRE-BA and A3C. A2C surpassed A3C and significantly outperformed HOLD and PRE-BA.

Overall, using TIs, data augmentation, ensemble model and A2C, our best agent was found to trade 33805.10 MWh of electricity across 1760 hourly test contracts; yielding significantly positive profits of €97853.69. We hope our findings inspire others to utilise novel DAM price forecasting methods and DRL algorithms in statistical arbitrage trading. Researchers interested in building upon our work are advised to, for instance, assess other actor–critic versions, such as soft actor–critic (SAC).

CRediT authorship contribution statement

Sumeyra Demir: Conception and design of study, Acquisition of data, Analysis and interpretation of data, Software, Formal Analysis, Drafting the manuscript. **Koen Kok:** Revising the manuscript critically for important intellectual content. **Nikolaos G. Paterakis:** Revising the manuscript critically for important intellectual content.

Acknowledgements

The authors would like to thank Walter van Alst from Scholt Energy for his support and guidance. All authors approved the version of the manuscript to be published.

References

[1] C. Saravia, Speculative trading and market performance: the effect of arbitrageurs on efficiency and market power in the New York electricity market, Center Study Energy Mark. (2003).

[2] S. Baltaoglu, L. Tong, Q. Zhao, Algorithmic bidding for virtual trading in electricity markets, IEEE Trans. Power Syst. 34 (1) (2019) 535–543, http://dx.doi.org/10.1109/TPWRS.2018.2862246.

[3] J. Lago, G. Marcjasz, B. De Schutter, R. Weron, Forecasting day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark, Appl. Energy 293 (2021) 116983, http://dx.doi.org/10.1016/j.apenergy.2021.116983.

[4] J. Lago, F.D. Ridder, B.D. Schutter, Forecasting spot electricity prices: Deep learning approaches and empirical comparison of traditional algorithms, Appl. Energy 221 (2018) 386–405, http://dx.doi.org/10.1016/j.apenergy.2018.02.069.

[5] S. Demir, K. Mincev, K. Kok, N.G. Paterakis, Introducing technical indicators to electricity price forecasting: a feature engineering study for linear, ensemble, and deep machine learning models, Appl. Sci. 10 (1) (2019) http://dx.doi.org/10.3390/app10010255.

[6] S. Demir, K. Mincev, K. Kok, N.G. Paterakis, Data augmentation for time series regression: Applying transformations, autoencoders and adversarial networks to electricity price forecasting, Appl. Energy 304 (2021) 117695, http://dx.doi.org/10.1016/j.apenergy.2021.117695.

[7] T.A. Nakabi, P. Toivanen, Deep reinforcement learning for energy management in a microgrid with flexible demand, Sustain. Energy Grids Netw. 25 (2021) 100413, http://dx.doi.org/10.1016/j.segan.2020.100413.

[8] C.-S. Tai, J.-H. Hong, D.-Y. Hong, L.-C. Fu, A real-time demand-side management system considering user preference with adaptive deep Q learning in home area network, Sustain. Energy Grids Netw. 29 (2022) 100572, http://dx.doi.org/10.1016/j.segan.2021.100572.

[9] J. Schulman, P. Moritz, S. Levine, M.I. Jordan, P. Abbeel, High-dimensional continuous control using generalized advantage estimation, 2015, pre-print, arXiv:1506.02438.

[10] V. Mnih, A.P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, K. Kavukcuoglu, Asynchronous methods for deep reinforcement learning, in: M.F. Balcan, K.Q. Weinberger (Eds.), Proceedings of the 33rd International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 48, PMLR, New York, New York, USA, 2016, pp. 1928–1937.

[11] H. Yang, X.-Y. Liu, S. Zhong, A. Walid, Deep reinforcement learning for automated stock trading: an ensemble strategy, in: Proceedings of the First ACM International Conference on AI in Finance, ICAIF '20, Association for Computing Machinery, New York, NY, USA, 2020, http://dx.doi.org/10.1145/3383455.3422540.

[12] Z. Zhang, S. Zohren, S. Roberts, Deep reinforcement learning for trading, J. Financ. Data Sci. 2 (2) (2020) 25–40, http://dx.doi.org/10.3905/jfds.2020.1.030.

[13] Y. Liu, D. Zhang, H.B. Gooi, Data-driven decision-making strategies for electricity retailers: A deep reinforcement learning approach, CSEE J. Power Energy Syst. 7 (2) (2021) 358–367, http://dx.doi.org/10.17775/CSEEJPES.2019.02510.

[14] D. Xiao, J.C. do Prado, W. Qiao, Optimal joint demand and virtual bidding for a strategic retailer in the short-term electricity market, Electr. Power Syst. Res. 190 (2021) 106855, http://dx.doi.org/10.1016/j.epsr.2020.106855.

[15] D. Xiao, W. Qiao, L. Qu, Risk-constrained stochastic virtual bidding in two-settlement electricity markets, in: 2018 IEEE Power Energy Society General Meeting, PESGM, 2018, pp. 1–5, http://dx.doi.org/10.1109/PESGM.2018.8586115.

[16] H. Mehdipourpicha, S. Wang, R. Bo, Developing robust bidding strategy for virtual bidders in day-ahead electricity markets, IEEE Open Access J. Power Energy 8 (2021) 329–340, http://dx.doi.org/10.1109/OAJPE.2021.3105097.

[17] W. Tang, R. Rajagopal, K. Poolla, P. Varaiya, Model and data analysis of two-settlement electricity market with virtual bidding, in: 2016 IEEE 55th Conference on Decision and Control, CDC, 2016, pp. 6645–6650, http://dx.doi.org/10.1109/CDC.2016.7799292.

[18] Y. Li, N. Yu, W. Wang, Machine learning-driven virtual bidding with electricity market efficiency analysis, IEEE Trans. Power Syst. 37 (1) (2022) 354–364, http://dx.doi.org/10.1109/TPWRS.2021.3096469.

[19] L. Pozzetti, J. Cartlidge, Trading electricity markets using neural networks, in: Proceedings of the 32nd European Modeling & Simulation Symposium, EMSS 2020, 2020, pp. 311–318, http://dx.doi.org/10.46354/i3~m.2020.emss.045.

[20] S. Demir, B. Stappers, K. Kok, N.G. Paterakis, Statistical arbitrage trading on the intraday market using the asynchronous advantage actor–critic method, Appl. Energy 314 (2022) 118912, http://dx.doi.org/10.1016/j.apenergy.2022.118912.

[21] J. Lago, F.D. Ridder, P. Vrancx, B.D. Schutter, Forecasting day-ahead electricity prices in Europe: The importance of considering market integration, Appl. Energy 211 (2018) 890–903, http://dx.doi.org/10.1016/j.apenergy.2017.11.098.

[22] S. Gunduz, U. Ugurlu, I. Oksuz, Transfer learning for electricity price forecasting, 2020, pre-print, arXiv:2007.03762.

[23] Z. Yang, L. Ce, L. Lian, Electricity price forecasting by a hybrid model, combining wavelet transform, ARMA and kernel-based extreme learning machine methods, Appl. Energy 190 (2017) 291–305, http://dx.doi.org/10.1016/j.apenergy.2016.12.130.

[24] J. Zhang, Z. Tan, Y. Wei, An adaptive hybrid model for short term electricity price forecasting, Appl. Energy 258 (2020) 114087, http://dx.doi.org/10.1016/j.apenergy.2019.114087.

[25] E.E. Elattar, S.K. Elsayed, T.A. Farrag, Hybrid local general regression neural network and harmony search algorithm for electricity price forecasting, IEEE Access 9 (2021) 2044–2054, http://dx.doi.org/10.1109/ACCESS.2020.3048519.

[26] W. Shi, Y. Wang, Y. Chen, J. Ma, An effective two-stage electricity price forecasting scheme, Electr. Power Syst. Res. 199 (2021) 107416, http://dx.doi.org/10.1016/j.epsr.2021.107416.

[27] I. Oksuz, U. Ugurlu, Neural network based model comparison for intraday electricity price forecasting, Energies 12 (23) (2019) http://dx.doi.org/10.3390/en12234557.

[28] K. Maciejowska, B. Uniejewski, T. Serafin, PCA forecast averaging—predicting day-ahead and intraday electricity prices, Energies 13 (14) (2020) http://dx.doi.org/10.3390/en13143530.

[29] M. Narajewski, F. Ziel, Ensemble forecasting for intraday electricity prices: simulating trajectories, 2020, Papers, arXiv.org.

[30] B. Uniejewski, G. Marcjasz, R. Weron, Understanding intraday electricity markets: Variable selection and very short-term price forecasting using LASSO, Int. J. Forecast. 35 (4) (2019) 1533–1547, http://dx.doi.org/10.1016/j.ijforecast.2019.02.001.

[31] S. Hagemann, Price determinants in the german intraday market for electricity: an empirical analysis, J. Energy Mark. 8 (2015) 21–45, http://dx.doi.org/10.21314/JEM.2015.128.

[32] T. Janke, F. Steinke, Forecasting the price distribution of continuous intraday electricity trading, Energies 12 (22) (2019) http://dx.doi.org/10.3390/en12224262.

[33] D. Shah, S. Chatterjee, A comprehensive review on day-ahead electricity market and important features of world's major electric power exchanges, Int. Trans. Electr. Energy Syst. 30 (7) (2020) e12360.

[34] R. Scharff, M. Amelin, Trading behaviour on the continuous intraday market Elbas, Energy Policy 88 (C) (2016) 544–557.

[35] S. Demir, K. Kok, N.G. Paterakis, Exploratory visual analytics for the european single intra-day coupled electricity market, in: 2020 International Conference on Smart Energy Systems and Technologies, SEST, 2020, pp. 1–6, http://dx.doi.org/10.1109/SEST48500.2020.9203043.

[36] P. Shinde, M. Amelin, A literature review of intraday electricity markets and prices, in: 2019 IEEE Milan PowerTech, 2019, http://dx.doi.org/10.1109/PTC.2019.8810752.

[37] T. Brijs, F. Geth, C. De Jonghe, R. Belmans, Quantifying electricity storage arbitrage opportunities in short-term electricity markets in the CWE region, J. Energy Storage 25 (2019) 100899, http://dx.doi.org/10.1016/j.est.2019.100899.

[38] B. Stappers, N.G. Paterakis, K. Kok, M. Gibescu, A class-driven approach based on long short-term memory networks for electricity price scenario generation and reduction, IEEE Trans. Power Syst. 35 (4) (2020) 3040–3050, http://dx.doi.org/10.1109/TPWRS.2020.2965922.

[39] R. Tibshirani, Regression shrinkage and selection via the lasso, J. R. Stat. Soc. Ser. B Stat. Methodol. 58 (1) (1996) 267–288.

[40] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32, http://dx.doi.org/10.1023/A:1010933404324.

[41] J.H. Friedman, Greedy function approximation: a gradient boosting machine, Ann. Statist. 29 (2000) 1189–1232.

[42] R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, second ed., The MIT Press, 2018.

[43] I. Boukas, D. Ernst, T. Théate, A. Bolland, A. Huynen, M. Buchwald, C. Wynants, B. Cornélusse, A deep reinforcement learning framework for continuous intraday market bidding, Mach. Learn. (2021).

[44] C. Kath, F. Ziel, Optimal order execution in intraday markets: minimizing costs in trade trajectories, 2020, pre-print, arXiv:2009.07892.

[45] ENTSO-E Transparency Platform, 2020, https://transparency.entsoe.eu/. (Accessed 12 December 2020).

[46] Scholt Energy Control, 2020, https://www.scholt.com. (Accessed 12 December 2020).