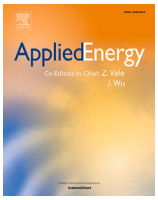
Applied Energy 397 (2025) 126262

Contents lists available at ScienceDirect

Applied Energy

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



Demand and supply curve forecasting using a monotonic autoencoder for short-term day-ahead electricity market bid curves

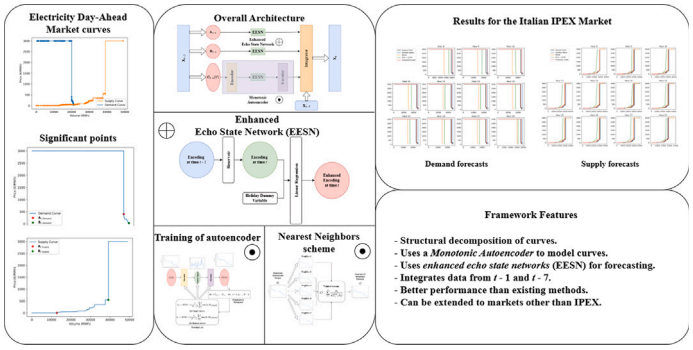
Nabangshu Sinha [a,*], Carlo Lucheroni [b]

[a] International School of Advanced Studies, University of Camerino, via Gentile III Da Varano, Camerino (MC), 62032, Italy
[b] School of Sciences and Technology, University of Camerino, via Madonna delle Carceri 9, Camerino (MC), 62032, Italy

HIGHLIGHTS

- Proposal of a short-term forecasting framework for day-ahead electricity market demand and supply curves.
- Curves are decomposed into two key points and a segment, enabling separate handling of volume and price dynamics.
- Implemented an enhanced Echo State Network (EESN) with weekday info to forecast the curve's key volume points.
- Implemented a monotonic autoencoder to encode the segment while preserving curve monotonicity and then forecast using EESN.
- Components are combined via optimization using special loss, and the model outperforms PCA and other benchmarks on IPEX data.

GRAPHICAL ABSTRACT



ARTICLE INFO

Keywords:
Day-ahead electricity market forecasting
Electricity demand and supply curves
Echo state network
Autoencoder
Heterogeneous curves mean absolute error

ABSTRACT

This paper proposes a novel short-term modeling and forecasting framework for day-ahead electricity market demand and supply price/volume curves. These economically and financially important curves are obtained daily from data derived from the full set of the price/volume bids submitted to the market, and are computed preliminarily to the market price setting phase. They contain a wealth of market information, but are difficult to forecast due to the peculiarity of their data structure. They are intrinsically monotonic, and take values on an irregularly distributed set of volume values which change in location and number each day. Unlike in the case of electricity price forecasting, only a few research groups have addressed the curve forecasting problem so far. In addition, because it is difficult to preserve monotonicity when forecasting these curves, and although its violation can result in incoherent forecasts, the existing curve forecasting models usually don't explicitly enforce this constraint. In this paper, a modeling and forecasting framework is proposed which decomposes each curve into three structurally meaningful and interpretable geometrical entities, corresponding to macroscopic features of the curves. At a given time t, these geometrical entities are two special (price,volume) curve points $\mathbf{A}_t$ and $\mathbf{B}_t$, and the price/volume vector $\mathbf{C}_t$ between them. On the one hand, the $\mathbf{A}_t$ and $\mathbf{B}_t$ points are directly and individually forecast using a variant of the Echo State Network machine learning architecture. On the other hand, the dependency on time of the $\mathbf{C}_t$ segment is simplified, and this simplified $\mathbf{C}_t$ is forecast by forecasting its reduced representation as internal to a suitably developed monotonic autoencoder network. Curve comparison, necessary for curve fitting, for the quality assessment of the forecasts, and for benchmarking the proposed framework against other available

* Corresponding author.
Email address: nabangshu.sinha@unicam.it (N. Sinha).

https://doi.org/10.1016/j.apenergy.2025.126262
Received 1 February 2025; Received in revised form 4 May 2025; Accepted 31 May 2025

models, is made by means of a suitably developed metric algorithm which we call 'Heterogeneous Curves Mean Absolute Error'. The three components of the curves, $\mathbf{A}_t$, $\mathbf{B}_t$ and $\mathbf{C}_t$, are hence optimally combined and glued together by means of optimization of this error. The framework is tested on data from the NORD zone of the Italian day-ahead IPEX zonal market. It is numerically shown that forecasting with the proposed framework outperforms forecasting with the few benchmarks available, including stochastic-functional and PCA-based models.

1. Introduction

A day-ahead electricity market (DAM) is a special type of market in which a market operator (MO), in collaboration with a system operator (SO), manages a daily auction in which electric energy is traded for the individual 24 h of the day, in a transparent and market-based way, and electricity prices are set by marginal pricing. On a given day, after the auction is closed, 24 prices for energy unitary volume (in currency/Megawatt-hour, like EUR/MWh) are set for the 24 h of the next day. Only a subset of bids in the auction is then marked as accepted, and are dispatched (allocated) for coming delivery. The pool of the participants in this market usually includes electricity generators, distributors, and large consumers. DAMs are organized in this way because the collection of trading information (at least one day in advance of electricity delivery) by part of the MO allows the SO to smoothly handle potentially disruptive electricity production/consumption unbalances. At the moment of clearing the auction, received demand and supply bids are aggregated into 'aggregated demand and supply curves' or, more in short, 'demand and supply curves', in the number of 24 pairs of curves, one pair per hour. In some markets, these curves are disclosed to the public ex-post for free, including information about which bids are accepted. This is a lot of information. The autoregressive modeling and forecasting of these data structures, the daily 24 pairs of curves, is the subject of this paper.

The reasons why research on curve modeling and forecasting is important come mainly from the following two considerations.

The advantage of achieving better modeling and forecasting DAM price dynamics and its volatility comes first. After aggregation, the MO uses the obtained curves for marginal pricing, which is the DAM's main financial aim. Marginal pricing is based on computing the intersection of the supply curve with the corresponding demand curve to give a market clearing price/volume pair. The MO takes care of this, after taking into account technical constraints indicated by the SO, if applicable. The 24 pairs' intersections become the so-called hourly 'realized price/volume pairs'. The realized prices get published, and become the official, reference prices for all DAM energy trades of the next day. In passing, notice here the double use of the word 'prices', as in 'bid prices' of curves (which are still not realized, i.e., virtual) and in 'realized prices' from auction clearing, two different entities, both usually referred to as 'prices'. Noticeably, in Europe most of the national DAMs are integrated into one continent-wide market, hence foreign entities can participate in each national market, making the fixing procedure even more complicated. Moreover, some of the national markets are zonal, meaning that they can be thought of as segmented into sub-markets (yet still interacting with the rest of the European system). Hence, gathering information on these curves, better understanding them, and possibly guessing their behavior for the next days, for the local and for the foreign markets, should constitute an all-important activity for market participants interested in electricity (realized) prices, regulators included. Indeed, this activity should not be less important than the activity of forecasting the 24 realized price/volume pairs themselves as autoregressions, a kind of forecast that is for itself nowadays routinely carried forward each day by DAM traders and energy finance researchers [1,2]. In the end, if the curves themselves can be accurately forecast, even the volatility of price dynamics can be better understood and managed. From this point of view, good curve forecasting is important mainly as a means of improving (realized) price forecasting, and not for itself.

However, curve forecasting is very interesting for understanding bid prices and for itself too. Especially for the short term of one day, which is the horizon which will be studied in this paper, there is a second key consideration to make. Good curve forecasting ability allows the SO to better understand and predict the betting behavior of specific generation agents on one side, and large demand and distribution (wholesale sellers) agents on the other side, and the reciprocal interaction between these market movers. This is analogous to studying limit order book data in stock finance. In addition, the SO can exploit good demand and supply curve modeling and forecasting for better understanding the consequences of its rules on the relationship between financial and technical aspects of its activity. Regarding this, there is in Europe an intense debate about pricing systems alternative to the current, marginal one, see for example Ref. [3]. This includes handling of transmission congestions by means of anticipating effects of financial supply–demand imbalances from market operations on power flows and grid stability. It also includes better scheduling and dispatching generation resources, especially renewable resources, based on both financial and environmental data [4]. As to the MO itself, such knowledge can also help it in its regulatory capacity of spotting odd bids intended to manipulate the market. Generators and distributors themselves, as market agents, can use this ability in the short term to predict other users' behavior and improve their optimal bidding strategies. In fact, one of the main prerequisites for finding optimal bidding strategies is knowing the competitors' bids [5,6]. For example, the aggregated supply curve can indeed be considered by generators as surrogate data for competitor generators' behavior [7]. As to longer term horizon forecasting, which can be considered a possible development of short term forecasting, it can also help SOs to manage the integration of renewable energy sources by taking into account the impact of their variable nature on agents, and in planning for an effective and economical deployment of energy storage or demand response systems [8–10].

Despite all these possible indirect and direct advantages, just a small number of papers in the literature are devoted to modeling and forecasting this kind of curves and bid prices. A possible reason for this is the technical difficulty of modeling and forecasting such curves because of their peculiar data structure. For example, these curves, seen as price values as function of volumes, are intrinsically monotonic, as demanded by microeconomic theory and by market regulations (this aspect will be discussed in detail in next sections). In addition, they have an irregularly distributed support in volume that changes in distribution each day due to the bidding process. Forecasting such structures is difficult, and in the specific monotonicity itself is difficult to be preserved when forecasting them, whereas lack of it can result in incoherent forecasts. Existing literature about curve modeling and forecasting is very limited. We have found eight basic models in all, collected in Table 1, plus some other models derived from these. These basic models use techniques that are very different from each other. They go from the stochastic functional approach of Ref. [11] to the non-stochastic function method of Ref. [12], to the machine learning methods used in Refs. [7,10,13,14], and to the purely econometric method of Ref. [15] and its derivations [16–18] (off-table). Noticeably, all models forecast individual hours as if the bids for different hours were uncoupled from each other. Since the reasons for the related modeling choices depend strongly on the peculiarity of the curves and on the specific features of the curves which researchers are most interested in, they will be discussed in detail only after the curves' data structures are discussed in the next Sections. However, it can be

Table 1
List of the foundational works discussed in the literature review section. Notice the column 'Support', which indicates the support (P or V) chosen by the modeler.

Authors	Title	Model Name	Year	Support	Reference no.
Pelagatti	Supply function prediction in electricity auctions	PCA + Linear Autoregression	2012	P	[13]
Ziel and Steinert	Electricity price forecasting using sale and purchase curves: The X-Model	X-model	2016	P	[15]
Shah and Lisi	Forecasting of electricity price through a functional prediction of sale and purchase curves	FAR	2020	V	[11]
Mestre et al.	Forecasting hourly supply curves in the Italian Day-Ahead electricity market with a double-seasonal SARMAHX model	Double seasonal SARHMAX model	2020	P	[19]
Guo et al.	Forecast aggregated supply curves in power markets based on LSTM model	PCA + LSTM	2021	V	[7]
Yildirim et al.	Supply curves in electricity markets: a framework for dynamic modeling and Monte Carlo forecasting	MCMC modelling and forecasting	2022	P	[10]
Ciarreta et al.	Forecasting electricity prices using bid data	Piecewise functional modelling + forecasting	2023	V	[12]
Tang et al.	Forecasting individual bids in real electricity markets through machine learning framework	Dimensionality reduction + Transformer	2024	V	[14]

anticipated that most of these models are designed for specific application to either supply and demand curve, which have very different nature, and most of the time with a specific national market in mind. Hence, they don't perform the same on the two curve types and in different markets. In addition, none of them explicitly takes into account monotonicity of the curves, which creates problems when interpreting their results in terms of real possible market outcomes. And, as much as importantly, when looking at the models in the aggregate as a pool of possible approaches, it becomes evident that this pool of models can be divided into two subgroups as detailed in the 'Support' column of Table 1. The two subgroups will be called $P(V)$ approach and $V(P)$ approach in the next Sections. This suggests a new modeling scheme that makes use of both approaches at once, and which can improve on the results of the two individual subgroups.

In this paper, a modeling and forecasting framework is hence proposed which aims to overcome some of the modeling and forecasting difficulties mentioned above, in a way not already present in past and current literature. This is achieved by decomposing the curves into three structurally meaningful and interpretable components corresponding to macroscopic features of the curves, connected to market behavior. These components are then forecasted individually using nonlinear autoregression, and coupled among themselves by means of an optimization technique called differential evolution. At a given time t, these geometrical entities are two special (price,volume) curve points $\mathbf{A}_t$ and $\mathbf{B}_t$, and the price/volume vector $\mathbf{C}_t$ between them. On the one hand, the $\mathbf{A}_t$ and $\mathbf{B}_t$ points are directly and individually forecast using a variant of the Echo State Network machine learning architecture, specifically designed for the task and to be discussed in detail in the next sections. On the other hand, the dependency on time of the $\mathbf{C}_t$ segment is simplified, and this simplified $\mathbf{C}_t$ is forecast by forecasting its reduced representation as internal to a suitably developed 'monotonic autoencoder' neural network. In the making, we will mix the $P(V)$ and $V(P)$ approaches. All of this will be discussed later in detail. The proposed architecture is modular, so that for example it has the advantage that, in case, it can easily include an arbitrary number of lagged (i.e., past) data pieces in its autoregressions. Like in all literature, different hours are assumed uncoupled, but the framework's modularity together with the very small number of components could be rather easily exploited to try to forecast all hours at once. Curve comparison, necessary for curve fitting (between ground truth and forecast), for the quality assessment of the forecasts, and for benchmarking the proposed framework against other available models, is made by means of a suitably developed metric algorithm which we call 'Heterogeneous Curves Mean Absolute Error'. The three components of the curves, $\mathbf{A}_t$, $\mathbf{B}_t$ and $\mathbf{C}_t$, after being forecasted, are then optimally combined and glued together by means of the optimization of this error.

Hopefully our approach can bring at once some apparently disjoint advantages to curve modeling and forecasting. It is backed up by an exhaustive and up to date catalog of available curve modeling and forecasting papers. It allows for segmenting the problem into easier parts clearly connected to market data interpretation, and for taking into account curve information at their important past lags in an easy way. It will also allow to take on supply and demand forecasting in the same way, while preserving monotonicity by means of the suitably designed 'monotonic autoencoder' architecture. In any case, once compared with a selection of existing models which tries to take into account most of the discussed modeling and forecasting strategies, the proposed method displays very good numerical results and less error than the tested models.

In order to ground all coming discussion on what is peculiar in this problem and on how past literature has dealt with that, it is hence better to recall right after this Introduction, namely in Section 2, the following three questions: I. how these curves are obtained from data, II. what is the type of information which they convey to the market, and III. why the curves' data structure is considered so much peculiar and difficult to model and forecast. Hopefully this will put in clearer perspective both the further discussion on available literature on electricity curves forecasting, and the presented forecasting framework solution, especially if put in its relation to available literature. All the rest of the paper will make implicit reference to information discussed in the next Section.

The paper is hence organized as follows. After this Introduction and after Section 2 devoted to a detailed discussion of the curves and related data and meaning, Section 3 will provide a review of electricity market curve forecasting literature, analyzing strengths and weaknesses of the currently available models, and will highlight what we think are the main innovations and contributions of our paper. Section 4 describes our framework. It will first present an overview of the proposed framework in terms of its main components, and a brief review of some background information about the standard machine learning architectures which are to be modified in order to be used in the framework will be included. Since modeling is not necessarily forecasting, it will then describe in detail and step by step the proposed framework, as a model and as a forecasting device. Section 5 will discuss the Italian Power Exchange (IPEX) dataset in the specific, the chosen benchmarks, and the forecasting error metric, as introduced and used. Section 6 will provide a detailed numerical (and to some extent graphical) analysis of the performance of the suggested framework on the IPEX data in comparison with the results from the selected benchmark models. Section 7, based on the numerical results and on the previous discussion on existing literature, will provide an overall discussion of the proposed framework. Finally, Section 8 will conclude by summarizing major findings, highlighting practical consequences, and suggesting future research directions. Appendix A will in addition detail how to practically collect the IPEX bid data used in the paper from the related web site.

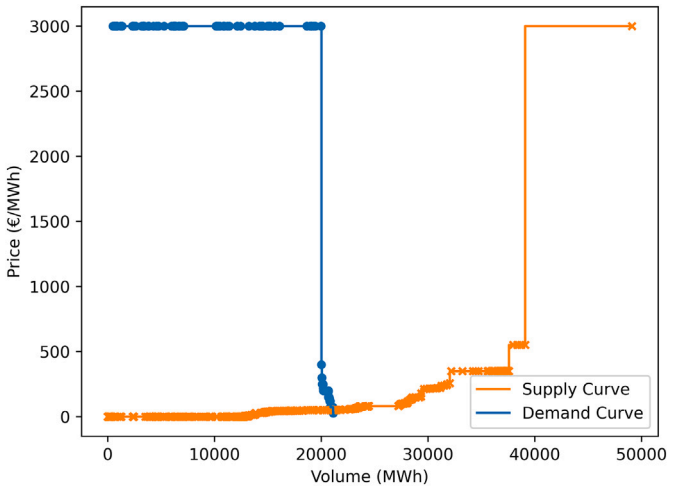


Fig. 1. Aggregated demand and supply curves for hour 9 of January 5, 2018 from the NORD zone of the Italian IPEX market, seen as prices (on the vertical axis) in function of volumes (on the horizontal axis). The blue dots show the demand bid data (contractual conditions for buying) and the orange crosses show the supply bids data (conditions for selling). The continuous lines are not in the data, but cannot be considered just as interpolations (see text). Noticeably, there are almost no supply bids for aggregated volumes larger than 39,000 MWh, and the line in that segment doesn't represent existing data. Notice also that there are no demand bids at all for volumes larger than about 23,000 MWh.

2. The curves: interpolated sequences

An exemplary plot of a supply and demand curves pair from the NORD zone of the Italian IPEX DAM for hour 9 of January 5, 2018 is shown in Fig. 1. The IPEX DAM is managed by the Gestore dei Mercati Energetici (GME) [20], an Italian state-owned company. This pair of curves is taken from the dataset used in this study, which consists of 730 IPEX market days from January 1, 2018, to December 31, 2019, collected from Ref. [20] (more on that in Section 5). For clarity of exposition, it is better to divide the discussion of these curves into the following three parts: what supply curves represent (Section 2.1), what demand curves represent (Section 2.2), and what is the role of price/volume curves in DAM pricing (Section 2.3). This discussion is also intended to make even more explicit the reasons why curves' modeling and forecasting are so much important for DAM business intelligence.

2.1. The supply curve as a sequence of points

This first subsection is related to the supply curve $P^S(V^S)$, represented in orange and with crosses in Fig. 1. The MO builds up this plot in the following way, based on collected supply bid price/volume data I^S pairs of form (p_j^S, v_j^S), $j = 1, \dots I^S$, where for example a bid pair could be ($p_j^S = 100$ eur/MWh, $v_j^S = 30$ MWh). Starting from the lowest value of p_j^S, usually zero (but not necessarily so), the MO orders the pairs

$$(p_j^S, v_j^S)$$

in an increasing way with respect to p_j^S. If there is more than one pair corresponding to the same value of p_j^S, the MO sub-orders them in an arbitrary order. In this way the MO obtains a new sequence of pairs, again of length I^S, but now strictly ordered in a new index i. Then it computes the strictly increasing sequence of 'aggregated volumes' $V_i^S = \sum_{k \le i} v_k^S$ (notice the $k \le i$ symbol) to which a non-decreasing monotonic sequence $P_i^S = p_i^S$ corresponds (hence not necessarily strictly monotonic). In this way a sequence

$$(P_i^S, V_i^S)$$

of price/'aggregated volume' pairs is obtained, which can be plotted as $P_i^S(V_i^S)$ and can have flat parts, since more than one value p_j^S can correspond to P_i^S. This is the orange sequence of crossed points shown in Fig. 1. Overall, this procedure results in a sequence of points monotonously sloping upward in the direction of increasing volumes by construction, with 'aggregated volume' coordinates on the abscissa, irregularly distributed along the horizontal axis, changing daily in their number I^S and in their positions, corresponding to prices P_i^S on the ordinate which change as much.

A lot of information about suppliers' behavior is contained in this sequence. At first, a supplier signals with its bid that it won't sell a specific volume v_j^S at a price less than $p_j^S = P_j^S$. However, there could be in the market other suppliers more than happy to sell a smaller volume for that same price P_j^S, and other suppliers happy to sell a suitably smaller volume for smaller prices. When prices are formed, this way of constructing the curve will take care of that interest, as it will be seen shortly. In addition, the position of a data point along the curve is revealing a lot of information about market consistency and structure. For example, the zero-price left flat part of the supply curve usually corresponds to the bids of solar or other renewables-based generators which in this way declare themselves as available to take the risk of delivering for free the energy which they cannot store (but in the marginal pricing scheme they will actually not produce for free, as it will be seen shortly). This means that, daily, the larger the available quantity of renewables, the more rightward is the placement of the first nonzero price. This threshold is then connected to a special point on the curve, which one could call $\mathbf{A}_{supply}$. For serving larger aggregated volumes, the cost of production increases from zero in function of the type of production technology required, and suppliers start on their part requiring minimum remuneration prices that then increase in steps: this stepped up structure is called 'power stack'. The most expensive technology available in the market corresponds to the highest power stack price. This creates a second special point, which one could call $\mathbf{B}_{supply}$. These two special points and the overall power stack pattern (the steps) can be easily detected by eye in the supply curve of Fig. 1. In this curve, it is also worth considering the last data point on the r.h.s., the point at the maximum price allowed by the market rules, in this case 3000 euros per MWh. That point is due to typical selling agent strategic behavior. An agent places a bid there at the risk of almost never getting accepted for this bid, but also in order to capture and exploit tight market conditions, in the case when buyers are forced to accept any price.

2.2. The demand curve as a sequence of points

This second subsection is related to the demand curve $P_i^D(V_i^D)$. A demand curve plot is obtained like the supply curve plot, but in reverse, by first organizing the prices of I^D demand bids (p_j^D, v_j^D) in descending order on the vertical axis, the first price being the highest. In most markets, this first, highest price is set by regulations for the protection of the consumers (3000 euros per MWh in the IPEX market). As price values decrease, corresponding aggregated volumes on the horizontal axis are associated with prices $P_i^D = p_i^D$, and are obtained as $V_i^D = \sum_{k \ge i} v_k^D$ (notice the $\ge$ symbol), in reverse analogy to supply volumes. Overall, this procedure results in a sequence (P_i^D, V_i^D) monotonically decreasing rightward down to the minimum price allowed by the market rules, with aggregated volume coordinates irregularly distributed along the horizontal axis, and with a daily varying number I^D of points. This sequence can be plotted as $P_i^S(V_i^S)$ and can have flat parts. This is the blue sequence of dots shown in Fig. 1.

A lot of information about demand behavior is contained in this structure. With a bid (p_j^D, v_j^D), a demand-side agent signals the maximum price at which it would buy a specific quantity. However, there could be other demand in the market more than happy to buy a larger volume for that same price $p_j^D = P_j^D$, and there could be other demand happy to buy a suitably larger volume for larger prices. Position along the curve is revealing. Corresponding to the highest price, there could be in the market

a demand minimum threshold due to large consumers like production plants, or uninterruptible businesses or public services, that cannot do without electricity, and accept any (then maximum) prices. As working activity decreases, like it happens on weekends (hence this piece of information is all-important), the position of this threshold on the curve moves leftward, and a price appears at a lower value. This is thus another special point, which one could call $\mathbf{A}_{demand}$. A further special point is the threshold point corresponding to all possible demand generated in the market, beyond which demand falls to zero. One can call it $\mathbf{B}_{demand}$. These two special points can be easily detected by eye in the demand curve of Fig. 1, where represented uninterruptibles are on the left. In the Figure, the demand curve ends at a volume of about 23,000 MWh (23 GWh). On the r.h.s. of this value there is no further requested aggregated volume in the dataset—which means that the whole electricity system of the NORD zone of IPEX, on this specific day, didn't ask for more electricity than that. Noticeably, the curve segment between the two special points is very steep and the corresponding aggregate volume range is very thin, about one GWh in the Figure, and quickly and weekly moving backward and forward as time goes by. This is a very difficult target to track or forecast. However, this segment represents core information for the DAM price fixing procedure, and for explaining price dynamics and price volatility.

2.3. DAM pricing and the interpolation of the points

This third subsection is related to the fact that data come in points, but are represented as curves. Such curves could be thought of as some form of interpolation, not being actually present in the original data. Hence the word 'aggregated curve' could be a misnomer, which is not. The need for formally adding such interpolation structure to the sequence structures comes straight from marginal pricing. Based on marginal pricing, the MO finds the clearing volume V_* as the solution of the equation $P^S_*(V_*) = P^D_*(V_*) = P_*(V_*)$. At this condition, any supply below V_* will be paid not less than $P^S_*(V_*)$ (renewables included), and demand will be satisfied at not more than $P^D_*(V_*)$ (uninterruptibles included). Supply and demand bids outside this range are rejected, which just means discarding participants who signaled through their bids of not being interested in trading at the set price (for example, generators which are too much expensive). All other participants will instead be paid, or they will pay, P_* per unitary volume (1 MWh, in the case of the figure), the official price. Many sellers (those whose volumes v^S_j end up on the left of V_*, due to the condition $V^S_i = \sum_{k\le i} v^S_k$) will be paid a price equal to or larger than requested, many buyers (those whose volumes v^D_j end up on the left of V_*, due to the condition $V^D_i = \sum_{k\ge i} v^D_k$) will pay a price equal to or less than requested. Yet, finding the crossing point exactly at one of the bid volume points $V^{S/D}_i$ is highly unlikely, and volume values outside those already present in the data must be used. Hence, interpolation is contractually necessary, and each market has its own contractual rules for that. For example, the IPEX interpolates with steps [13,19]. The PJM (a North American zonal-like market), in contrast, considers some supply curves to be linearly interpolated and others to be step interpolated [7]. Hence, the lines in the Figure are there to make clear and official the procedure by which the MO places the official aggregated volume in that specific position. In addition, these lines suggest at what price on that specific day the market would have priced any other possible volume. For example, price risk management teams are very interested in the positioning of the flat rightmost segment of the supply curve, because if a crossing ends up there a price spike is formed, which is perceived as an extremely favorable (for supply) or adverse (for demand) market condition. Modelers and forecasters take notice, tend to work with curves instead of sequences, and usually simply call the two functions $P^S(V^S_i)$, $i = 1, \dots, I^S$ and $P^D(V^D_{i'})$, $i' = 1, \dots, I^D$, duly endowed with their interpolations, supply and demand curves $P^S(V)$ and $P^D(V)$.

The sequences in the curves change each day in support (positions of aggregated volumes and number of data), shape, positions of special points (this is especially critical in the case of the demand curve segment between its two special points, which is very steep and quickly moving), and consequently crossing points. As to crossing points, they mostly appear in the non-flat segments between the special points of both curves, hence the dynamics of these special segments play a major role in price formation, day after day. We will hence use the adjective 'heterogeneous' to characterize the sets of these curves, which are not defined on equi-spaced never-changing grids. The behavior of curves within their heterogeneous sets depends on weather forecast information, cost of fuel forecasts, scheduled and unscheduled maintenance activity, power grid conditions, conditions of external markets, social events, and weekdays [1]. Not even mutual interaction between hours is to forget, since DAM traders can see different hours as statistically coupled, and strategically bid in consequence. Modeling and forecasting realized price/volume pairs (P_*, V_*) for each hour, instead of full curves, is obviously much easier, and there is a very large body of literature on that. However, the quantity of potentially interesting information contained in the full curves, even for the sole aim of improving realized price forecasting, is so large, that it can be just worthwhile to attempt full curve forecasting, and exploring new models for that.

3. Literature review and proposed improvements

All the few available papers on curve forecasting invariably treat the sequences directly as curves, and can largely be categorized into two categories, depending on which support is used for modeling the curves:

1. papers that consider P as the support, hence focusing on $V(P)$,
2. papers that consider V as the support, hence focusing on $P(V)$.

As seen, Table 1 lists all papers to be discussed in this section, which in practice amount to most of the available literature devoted to DAM curve forecasting. Its 'Support' column indicates which support uses which model.

3.1. P as the support

The following papers all consider curves such that prices are on the abscissa, and the curves are consequently formulated as $V(P)$. Since the P axis has fixed minimum and maximum values (those defined by related market regulations), all such curves can be seen as varying over this support within fixed limits, which is a useful feature.

The first work in market curve forecasting was a paper by Pelagatti [13], published in 2012, which considered P as the support. The author reduced the number of P points of the curves by collapsing their set into those values that define their quantile boundaries. This allowed to define the curves over a uniform grid of (selected quantile) prices. Principal Component Analysis (PCA) was used to break down the redefined curves into independent components, of which the weights could be then forecast. By forecasting just these components the author was able to reconstruct the full forecasted curve. This work forecasted only supply curves. In 2016 Ziel et al. [15] tackled a different problem, that of forecasting hourly prices per MWh by means of predicting the curve crossing points. In their procedure, among others, they took on the problem of modeling irregularly spaced prices and related curves. They defined a grid of price bins common to all curves. The curves were reorganized using this price grid, and the V values corresponding to the bins were forecasted using independent LASSO (Least Absolute Shrinkage and Selection Operator) autoregressions, that is, linear autoregressions where a large number of uninteresting time lags are automatically discarded. This model was called the X-Model. Three further papers applied this model to hourly price forecasting. Refs. [16] and [17] were a probabilistic extension of the original model. Ref. [18] used some of the mechanisms of the X-model, yet limited to only a segment of entire curve. All this makes it apparent that the X-model approach is hence important for curve forecasting too, although not born to be aimed at it. Based on it, curve forecasting can be made by attaching the forecasted

V value to each price of the price binning set. Based on the probability empirical distribution of the bid prices, Li et al. [21] developed a new distance metric, which they utilized to cluster curves, and then, for a given day d and hour h, picked up cluster representatives, and forecasted them. This work considered only supply curves. Mestre et al. [19] modeled the supply curve using a deterministic-functional model that took seasonality into account. In this case modeling was based on representing the steplike structure of the curve as a series of sigmoid functions. Notice that this is a limiting assumption about the supply curves' shape, and this approach might not apply to other electricity markets such as the European Power Exchange (EPEX) or the North-American Midcontinent Independent System Operator (MISO) energy market, because these markets consider the curves to be linearly interpolated, either partly (MISO) or fully (EPEX). Linearly interpolated curves may not have the step like structure that is assumed in this work.

As said, all the techniques reported in Refs. [7,10,13,19] were specifically designed for supply curves, with their methods being developed using certain transformations that might not apply to demand curves. Except for the model by Mestre (which modeled the supply curve only), no guarantee of monotonicity was taken into consideration.

3.2. V as the support

The following papers consider curves such that volumes are on the abscissa, as P(V).

Shah and Lisi [11] used a stochastic-functional single-lag linear autoregression (FAR) to model and forecast the prices of both demand and supply curves, using V as the support defined on a grid. In this case, curves were represented as abstract stochastic functionals, of which the market curves were sampled representatives, and the expected value $\pi(V)$ corresponded to a forecast. A functional kernel which could be extracted from data would allow for statistically connecting one day to the next (because of the only available lag), as a forecast. The authors then extended this approach to arrange the use of an extra lag besides the first one (within the same stochastic one-lag theory) to improve their forecasts. The approach was applied to both supply and demand curves. Ciarreta et al. [12] proposed a gridded deterministic-functional approach called piecewise model which used a piecewise function for the curves. In this case monotonicity was implemented naturally, and not as a constraint. This model was developed with the Spanish market OMIE in mind. This work considered both demand and supply curves. Actually, in the OMIE, prices gradually change from P_{max} to other prices, and there is no abrupt drop or rise in the curves. Hence, the authors' piecewise model can work without the necessity of any forecast of transition volumes. This, however, makes this approach not applicable to the Italian IPEX, where an abrupt transition from P_{max} to other values (or the other way around) does occur. Guo et al. [7] reduced the dimensionality of the curves using a non-linear mapping and PCA. Then, they employed an LSTM neural network to forecast the principal components. This technique is quite akin to the one used in Pelagatti's approach in Ref. [13], although there are some notable differences between the two. This work considered only the supply curves. Tang et al. [14] used a combination of dimensionality reduction methods, namely PCA, non-negative matrix factorization (NMF), dictionary learning, and just downsampling of the curve, to reduce the curve representation, and then used a Transformer-based forecasting model. This work considered only the supply curves. No guarantee of monotonicity was taken into consideration except for the model by Ciarreta et al.

3.3. Common features of the models in the literature

The models discussed so far have some elements in common, namely:

1. All of them use some kind of approximation to deal with the curve heterogeneity problem, looking at the data sets as curves and not as sequences. These approximations introduce some amount of error in curve reconstruction.
2. All of these models more or less follow the following methodology: transform the whole curves (large flat parts included) by remapping them onto a common support, use a suitable dimensionality reduction method that may be direct (PCA, NMF) or indirect (stochastic functional parameterization like FAR, deterministic functional piecewise linear modeling) which takes into account the whole curve, use a suitable forecasting model to forecast the individual elements of the vectors with reduced dimensionality—linear, or taken from machine learning (ML) like long short-term memory (LSTM) or transformer networks—and finally reconstruct the curves using an inverse of the dimensionality reduction method. It should be noted that whereas both linear and non-linear forecasting models are used, the dimensionality reduction techniques are always linear.
3. Besides a few models, namely those in Refs. [12,19], none of the models enforced the condition of monotonicity in their respective forecast curves.
4. Other than Refs. [11,15], none of the other models consider the full span of the support, whether it is the P axis or the V axis. Instead, they focus on a limited segment of the curve.
5. Only Refs. [11,15] work with both demand and supply curves. In addition to these, other works focus mainly on forecasting supply curves, which don't include difficult, back and forth swinging, steep segments unlike demand curves.
6. All of these works except [11,12,15] consider an error function, the loss, that is heavily based on the probability distribution of historical prices. This approach is maybe too much specific to the studied dataset. In fact, it cannot lend itself to situations where the bids happen to deviate from the probability distribution of historical prices, at detriment of generality. For example, after a major energy market shock in 2022, the bidding price patterns of generators changed, and the price definition intervals expanded. If only pre-2022 historical bid prices are considered, then these prices will not be considered as the important parts of the price P axis in the related calibrated forecasting procedure, in this way missing what it is the true bidding nature of the market agents after a market shock. Similar situations could be due to market regulations changes, or some bids placed for deliberately manipulating the market.
7. All models assume that bidding on a specific hour is independent of bidding at all other hours. This uncoupling is clearly made to make the approach numerically feasible (and stable).

3.4. Items of our framework in common with existing literature

The framework we are presenting could be seen as having points in common with the model presented in Ref. [15] by Ziel et al., with the model presented in Ref. [12] by Ciarreta et al., and with the model presented in Ref. [11] by Shah et al., Recalling the discussion of literature carried out in Section 3, the following items mark the differences. Ref. [15] mainly worked on forecasting curves to determine prices. This was done by creating uniform bins to put prices in and then forecasting the associated V values. The drawback of this approach is that the local curve information is often lost, especially in the segment that contains the bulk of bids. There is no binning in the presented model. Ref. [12] developed a model for curve forecasting that breaks the curves into constituent parts and tries to model them linearly or logistically, but the model lacks any kind of forecasting for the breakpoints of the curve. As mentioned in the literature Section 3, this technique is not general enough, and it can only work in the Spanish market. In contrast, the model proposed in our paper is applicable to any market, including the Spanish market. Finally, Ref. [11] does use interpolation, but in a stochastic-functional framework, an approach which is unrelated to ours. In addition, all papers discussed in Section 3 used known models for dimensionality reduction, namely principal component analysis [7,13], or non-negative matrix factorization [14], or

downsampling [14,15]. However, a powerful deep learning model for dimensionality reduction such as the autoencoder was never taken into consideration.

As to this last point, at the beginning of our project we tried with a much more simple framework, which was based mainly on just forecasting the special points without much care to nonlinearity and to the interesting segment [22]. Since those results were not satisfactory, this led us to the framework as presented in this paper.

3.5. Our new contributions to literature

Overall, and also in the light of both DAM structure and of the literature discussion, the new contributions proposed in this paper are the following:

1. A novel data driven DAM supply and demand curve forecasting framework is introduced that is based on looking at curves as both sequences and curves. This is mainly done from the $P(V)$ perspective, but it internally does use the $V(P)$ perspective.
2. Within this framework, the interesting segment of the sequence-curve is extracted by locating its ends, these ends are forecast, and then the structural details of that segment are forecast as well by means of a specially devised autoencoder aimed at preserving monotonicity.
3. A network called the Enhanced Echo State Network is used to forecast the special points and the autoencoders' encodings.
4. Finally, an 'integration model' is introduced that puts together the three parts of the curve, optimally reconstructing the final curve over the entire span of the support V by using differential evolution.
5. In order to evaluate the dissimilarity between ground truth and forecasted heterogeneous curves, a novel metric called Heterogeneous Curve Mean Absolute Error (HC-MAE) is proposed, which has non-negativity, identity, and symmetry properties. It compares curves for all V values, and not just a segment of the curve. This is to ensure that this metric is not dependent on any factors that are not already explicitly present in the data, such as abstract distributions.

4. The proposed framework

In this section, we introduce our framework in relation to the curve data structures and their context (Section 2), and to how they were previously modeled and forecasted (Section 3). In accordance with the discussion of Section 2, capital V will denote the 'aggregate volume' of the curves, in contrast to small v used to denote 'volume' for pre-aggregation data. However, henceforth, we will very often use 'volume' to mean 'aggregate volume', for brevity.

4.1. Modeling problems and the data

Besides directly looking at the datasets as curves, one can also decide to start looking at the datasets as mere sequences, and as curves only after some preprocessing. Having decided to take the $P(V)$ approach, forecasting any of the sequences (either supply or demand) can be viewed as forecasting two mutually related vectors $\vec{P} = (P_1, \dots, P_i, \dots, P_I)$ and $\vec{V} = (V_1, \dots, V_i, \dots, V_I)$, each with the same number of entries, starting from a set of other lagged vectors with possibly different values of I. There are specific problems to this. When studied separately in time, the two dynamics of prices and volumes differ, if nothing else because the price P and volume V axes' maximum and minimum limits are different. Whereas the market regulations limit P into a fixed interval

$$[P_{\min}, P_{\max}],$$

the V axis, necessarily nonnegative, is always unbounded from above (of course to a reasonable extent). Another problem is the non-uniform monotonic structure of the sequences. As seen, a large part of a market sequence can be flat and contain very diluted information (or, when seen as segments, much redundant information), whereas most of information needed for forecasting interesting details is usually concentrated in a small segment. In this paper, this segment will hence be termed as 'the interesting segment'. A further issue is handling of time. Especially in the case of the demand curve, this segment shifts back and forth in time along the V axis, longitudinally from day d to day $d+1$ during the week and across the seasons of the year, and cross-sectionally from hour h to hour $h+1$ during the day. However, since in usual DAM protocols the clearing for the 24 h is made daily at once, the true dynamics is just along days, even though it seems that a fictitious dynamics develops along hours too. For this reason, a curve timestamp t has to be represented by splitting day d from hour h, as $\{t = d, h\}$, keeping in mind that the modeling and forecasting autoregressions will be performed on d only.

4.2. Data: the special points $\mathbf{A_t}$ and $\mathbf{B_t}$, and the interesting segment $\mathbf{C_t}$

Figs. 2 and 3 show plots of a typical demand $P_t^D(V)$ and supply $P_t^S(V)$ sequence-plus-curve respectively, interpolated in steps, from the NORD zone of the IPEX for hour 9 on January 1, 2018. Basing ourselves on data, what we consider special points are marked explicitly as $\mathbf{A}_{t,demand}$, $\mathbf{A}_{t,supply}$, to be called more simply $\mathbf{A}_t$, and $\mathbf{B}_{t,demand}$, $\mathbf{B}_{t,supply}$, to be called more simply $\mathbf{B}_t$. The curve segments between the points $\mathbf{A}_t$ and $\mathbf{B}_t$ (included), to be called respectively the 'interesting segments' $\mathbf{C}_{t,demand}$ and $\mathbf{C}_{t,supply}$ (also to be called more simply $\mathbf{C}_t$), are the parts of the curves with most details, since most of the (aggregate) variations of the bids, both in terms of prices and volumes, are concentrated in these regions. Fig. 4 visualizes the daily dynamics ('longitudinally', in econometric parlance) of the interesting segment $\mathbf{C}_{t,demand}$ of the demand curve $P_t^D(V)$ over time for the entire dataset, from January 1, 2018 to December 31, 2019. It is very relevant here to notice that this segment of the sequence, while widely and constantly shifting back and forth on the V axis, maintains more or less the same profile (a lengthier discussion of Figs. 2–4 will be provided in the next sections). This regularity can perhaps be exploited.

One can hence think that the problem of forecasting these curves can be simplified by breaking down the forecasting procedure into three sub-problems, namely:

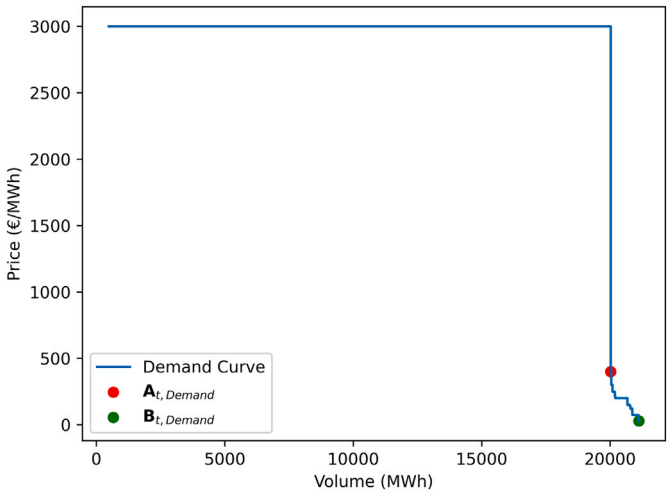


Fig. 2. $\mathbf{A}_{t,demand}$ and $\mathbf{B}_{t,demand}$ for demand curve, $P_t^D(V)$, for hour 9 of January 1, 2018.

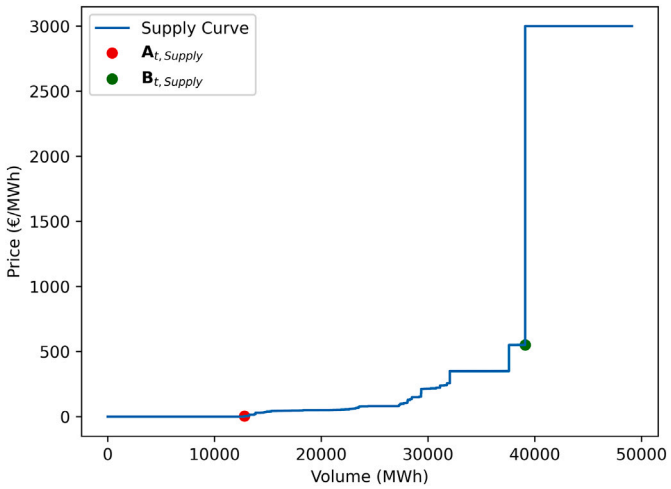


Fig. 3. $\mathbf{A}_{t,Supply}$ and $\mathbf{B}_{t,Supply}$ for supply curve, $P_t^S(V)$, for hour 9 of January 1, 2018.

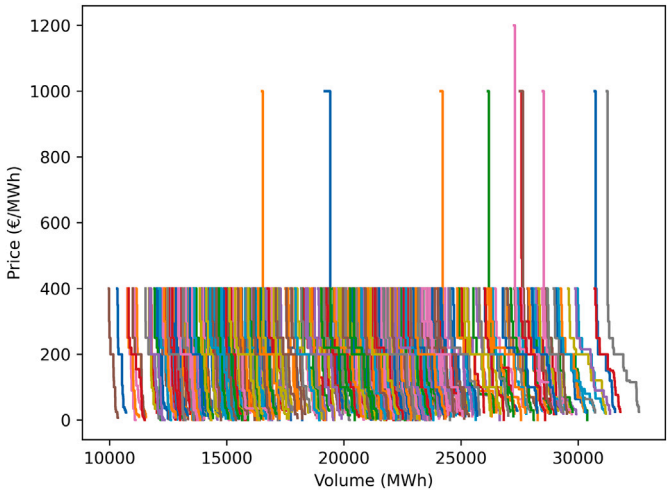


Fig. 4. The dynamics of the interesting segment of the demand curve $P_t^D(V)$ longitudinally along the entire dataset.

1. forecast the extrema within which the interesting segment occurs, that is, forecast the V coordinates of $\mathbf{A}_t$ and $\mathbf{B}_t$,
2. forecast just the price profile (say $\vec{P}$) of the sequence of the interesting segment $\mathbf{C}_t$, leaving the support sequence (say $\vec{V}$) aside—this operation is analogous to using a $V(P)$ approach—,
3. since the two extrema of $\mathbf{C}_t$ correspond to the P coordinates of $\mathbf{A}_t$ and $\mathbf{B}_t$, the two special points too are thus fully forecast.

For a given a specific hour, this would in practice consist of: first of defining in each daily curve plot a rectangular window, sliding in time, aimed at framing the interesting segment on the plane P/V, then of tracking this window and focusing further forecasting power only on the image inside the window. Notice that the width (linked to V) of this window will contract and dilate in time, whereas its height (linked to P) will not, which is one of the advantages of the $V(P)$ approach. Hence an assumption is made here that some basic stepping pattern of the segment is maintained across these variations. A further approximation which can help is that, since the correlation among the hours ('cross-sectional' in econometric parlance) is weak, hour-to-hour correlation can be neglected, so that hours can be forecast independently of each other.

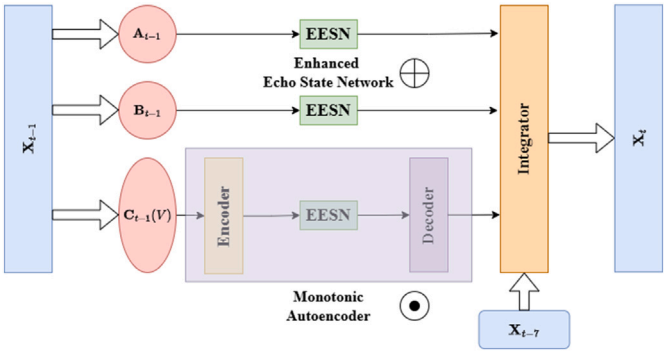


Fig. 5. Graphical representation of one possible implementation of the entire architecture of the proposed framework. X_t represents the curve sequence, and today's curve at t is forecast with data of yesterday at $t-1$ and one week before at $t-7$. More graphical information about the EESN boxes ($\oplus$) in Fig. 9, and about the monotonic autoencoder box ($\odot$) in Figs. 10 and 11.

4.3. Our framework: A, B, C, EESN, monotonic autoencoder, HC-MAE

We hence propose a curve forecasting framework based on looking at $P(V)$ curves as sequences-plus-curves. Its graphical representation can be seen in Fig. 5. Assuming that we want to forecast the curve realized today at time t with data coming from past days like $t-1$ (yesterday) and $t-7$ (one week ago), we address the subproblems above in the following way:

1. As a first step, the curves' points $\mathbf{A}_t$ and $\mathbf{B}_t$ are extracted from data. This allows us to capture the size and position of the interesting segment. Then, the V coordinates of $\mathbf{A}_t$ and $\mathbf{B}_t$ are forecast by a method based on the Echo State Network (ESN) architecture, because of computational speed reasons, as discussed later.
2. Noticeably, an ESN in its standard form uses the past $t-1$ value for forecasting the current one. A novel model called the Enhanced Echo State Network (EESN), based on the ESN architecture, is introduced. First, the ESN parses the volume time series and forecasts it. Then, the EESN is used to inject exogenous information (such as weekend and holiday timings) into the ESN outputs.
3. The profile (as a sequence) of the interesting segment $\mathbf{C}_t$ between $\mathbf{A}_t$ and $\mathbf{B}_t$ is forecasted in the following manner. First, an autoencoder-like neural network, consisting of an encoder and a decoder, is pre-trained over segments taken from the training dataset (a short recap of the standard autoencoder architecture will be given in the coming Section 4.4). As discussed in more detail in the next section, a specialized loss function to generate lower dimensional encodings within the autoencoder, namely between encoder and decoder, will be used. This autoencoder will be trained before training the forecasting model, and will implicitly contain a sort of basis on which to compute the encodings. Noticing that an autoencoder has a fixed number of inputs, all interesting segments (with variable I) must be associated with one grid, common to all curves, with a fixed number of nodes, by interpolating $P(V)$ on that support. This means looking at the sequence also as a curve, because of the interpolation. These encodings are then individually forecasted by the EESN method to predict future encodings. Then, using the decoder part of the autoencoder, the encodings are transformed back to their curve form. Very importantly, right because in this approach there is no theoretical guarantee that the reconstructed curves will be monotonic, a monotonic curve is created out of the autoencoder output by use of the Nearest Neighbors algorithm [23] and of five monotonic 'neighbours' of each forecast (more on that later). Noticeably, once $\mathbf{C}_t$ is forecast, the price components of $\mathbf{A}_t$ and $\mathbf{B}_t$ are forecast too, thus completing the forecast of the special points.

4. Once $\mathbf{A}_t$, $\mathbf{B}_t$ $\mathbf{C}_t$ are individually forecast, these three pieces must be glued together. This could be made by simply attaching them one to the other in an A, C, B sequence. However, better than that, one can try to optimize their fusion. This step will be called 'integration'. Optimization can be obtained by defining a measure for the distance between the obtained forecast curve and the ground truth data, and in some way minimize this distance. To this aim, a measure called HC-MAE is introduced, specific for the heterogeneity problem, similar to Mean Absolute Error (MAE). The HC-MAE turns out to be not differentiable, hence a special optimization technique is needed, called differential evolution [24], a metaheuristic algorithm. Hence, in a final step, intended to 'spread' the information (so far segregated in the individual three pieces of the curve) all over these three pieces, a non-differentiable optimization is carried out by minimizing the HC-MAE. This step is also exploited to inject lag 7 information in the $\mathbf{A}_t$, $\mathbf{B}_t$ and $\mathbf{C}_t$ regressions.

At the end of this procedure, the forecasts will be based on lagged information coming from $\mathbf{A}_{t-1}$ and $\mathbf{A}_{t-7}$, $\mathbf{B}_{t-1}$ and $\mathbf{B}_{t-7}$, and also from the 'interesting segment' $\mathbf{C}_t$ at times $t-1$ and $t-7$. These times are chosen based on the fact that the daily dynamics of a specific hour have a clear weekly seasonality, and of course depends heavily also on market and weather conditions of the day before.

4.4. Origin of the ML structures used in the framework: the autoencoder

As seen, modifications of two standard structures from machine learning, namely the autoencoder and the echo state network, are used in the proposed framework. A short discussion about these standard structures follows, in preparation to the discussion about their modifications as developed in the paper.

Autoencoders [25] are neural networks used for unsupervised learning of efficient encodings of data in a given dataset. These encodings can be seen as abstract representations of those data, but living in spaces either larger (for manifold embedding) or smaller (for dimensionality reduction) than the space of the original data. Given a static data set containing vectors of the same dimension s_d, an autoencoder acts as a standard, multilayer, vector feedforward network that presents at its vector output the transformation of its vector input (therefore its name). The network weights are trained by means of a loss function to produce an output as much close as possible to its input. At a specified layer inside the trained network, chosen deep, a hidden representation is hence formed for each data point of the training set. This representation hopefully takes into consideration all information available in whole training dataset, hence taking care of the full dataset context. If the specific internal layer has dimension $s_l < s_d$, that is, smaller than that of the input (and output) layers s_d, this representation can be seen as a compression of any data point presented to the network. This is the setup which will be used in the proposed framework. The stack of layers before the specific layer acts as an encoder (a compressor), whereas the stack after that layer acts as a decoder (a decompressor). After training, this setup will hence provide a sort of basis written in the weights, on which inputs are nonlinearly projected. In addition, any arbitrarily chosen vector of dimension s_l, not necessarily part of the training set, can be injected directly into the decoder as its input. The decoder would also generate an output of dimension s_d. This injected vector can be for example a forecast of the hidden representation of a data point recorded at a given time. This undecoded forecast would not be present in the training set, but its decoded vector can still be used as a forecast of that data point at next time. Autoencoders, as a dimensionality reduction devices [26], have found use in a range of varied application fields such as anomaly detection, imaging, and pharmaceutics, like those described in Refs. [27–29]. Noticeably, it is also possible to introduce additional terms in the loss function which are aimed to bias the autoencoder towards specialized purposes. In the next Section 4, it will be discussed how to modify an autoencoder to bias its outputs towards monotonicity by means of a suitable additional loss term. Since however this modification alone won't be able to force monotonicity, but just to bias towards it, a further final step will be needed and performed to achieve perfect monotonicity.

4.5. Origin of the ML structures used in the framework: the ESN

Echo State Networks (ESN) [30] are one of the variants of the recurrent neural network (RNN) architecture, hence especially devoted to model and forecast dynamics. RNNs have an input layer and an output layer, and one or more looped layers in between for autoregression. They can be trained on a data set of sequences, or on one single long sequence, to produce an output as close as possible to a target, which can be the next value of the sequence. For this reason they are very suitable for forecasting. However, RNNs usually require a computationally expensive training. An ESN is an RNN-like architecture designed to drastically reduce RNNs' training time. The ESN mechanism is based on the idea of reservoir computing [31]. The ESN input layer injects the signal into a random neural network of large dimension, the reservoir, that embeds the initial signal into a large vector which becomes subject to internal, complicated reservoir-dependent high-dimensional dynamics. In this sense, an ESN is the opposite of a dimension-reducing autoencoder, since its is an embedder. The reservoir remains always untrained. The output layer recollects the components of this embedding and prepares an output that can be compared with the target. The trick is that only the output layer is trained. If this layer is of simple design, training can be carried out in an easy and fast way. Noticeably, whereas in standard feedforward networks the freedom of training many layers is the source of adaptability to data (that is, the parametric space given by the weights), adaptability in an ESN is obtained differently. Here it is the untrainable but complex reservoir which acts as a creator of dynamic variability, whereas the output layer just harnesses this variability into one of the many possibilities of following the target. In order to working properly, ESNs require the definition of two meta-parameters, called spectral radius (related to stability) and leaking rate [30], which can be found empirically.

Standard or modified ESNs have already been used in electricity demand forecasting [32–34]. In these networks the internal structure of the neurons of the reservoir is always pretty standard, and always left untouched, so that standard ESNs don't usually incorporate dummy variables for exogenous information. In this paper, a method is proposed to enhance ESN inference with exogenous information. This can be obtained by adding a sort of linear regression after the ESN prediction. This linear regression maps the ESN output to the ground truth data, and adds exogenous information using dummy variables.

4.6. Dataset usage and partitioning

As said, we want to forecast the $P(V)$ curve at time t with data coming from previous days like at $t-1$ and $t-7$. In order to more clearly explain the proposed procedure, it is better to introduce at first the segmentation of the dataset used to test the procedure. After that, we will discuss how to forecast the special points, the components needed to forecast the special segment, and the procedure with which these parts are assembled into the forecasted curves.

The chosen IPEX dataset consists of 24 multi-column XML datasheets for each of 730 days, which go from January 1, 2018, to December 31, 2019 (two years). From each sheet, the two supply and demand curves of varying lengths are built from the contained individual bids, limited to the NORD zone.

The resulting curves' dataset is divided into the training, validation, and testing segments, as shown in Fig. 6. Specifically, the first 385 days (January 1, 2018–January 19, 2019) serve as the training set. The next 165 days (January 20, 2019–July 4, 2019), up to day 550, serve as a validation set. The final 180 days (July 5, 2019–December 31, 2019), up to day 730, serve as the test set.

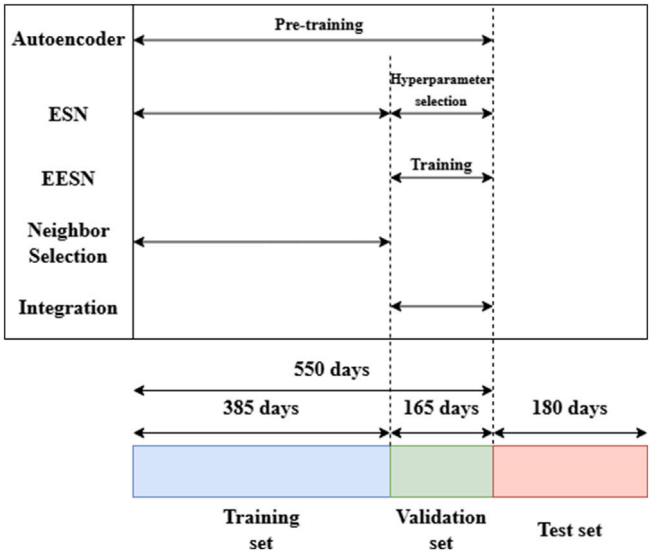


Fig. 6. Dataset segmentation. Data consist of 24 XML sheets for each of 730 market days, going from January 1, 2018, to December 31, 2019.

The framework components to be discussed in the following are trained as follows:

- Autoencoder training: The autoencoder is at first trained on the training plus validation segment, 550 day long, while keeping the rest of the framework still not trained. This is what we call pre-training.
- Guaranteeing monotonicity from the autoencoder: The five neighbors are selected from the training segment.
- Training of ESN and hyperparameter tuning: The Echo State Network model is trained on the training plus validation segments. A hyperparameter tuning (searching for the number of neurons to be used and the spectral radius) is in the specific performed on the 165 day validation segment.
- Training of the Enhanced ESN with exogenous information: The EESN model is trained on the ESN outputs from the validation set, using the same 165 days to integrate exogenous information.
- Training of the integration model: The integration model is trained on the validation set.

4.7. Details of the proposed framework: modeling $\mathbf{A_t}$, $\mathbf{B_t}$ and $\mathbf{C_t}$

The special points $\mathbf{A_t} = (V_A, P_A)$ and $\mathbf{B_t} = (V_B, P_B)$ of the curves $P_t(V)$ that delimit $\mathbf{C_t}$ can be defined formally (and operationally) by means of their V coordinates in the following way, together with $\mathbf{C_t}$ itself.

For the demand curve (see Fig. 2) the coordinates V of the points and their corresponding P values are

$$\mathbf{A}_{\mathbf{t},\text{demand}} \to V_A = \min\{V \mid 0 < P_t^D(V) < P_{max}\}, P_A = P_t^D(V_A), \tag{1}$$

$$\mathbf{B}_{\mathbf{t},\text{demand}} \to V_B = \max\{V \mid 0 < P_t^D(V) < P_{max}\}, P_B = P_t^D(V_B) \tag{2}$$

(notice the strict inequalities). The special segment $\mathbf{C}_{\mathbf{t},\text{demand}}$ is defined as the set of all points in between, including $\mathbf{A}_{\mathbf{t},\text{demand}}$ and $\mathbf{B}_{\mathbf{t},\text{demand}}$.

For the supply curve (see Fig. 3) the coordinates V of the points and their corresponding P values are

$$\mathbf{A}_{\mathbf{t},\text{supply}} \to V_A = \min\{V \mid 0 < P_t^S(V) < P_{max}\}, P_A = P_t^S(V_A) \tag{3}$$

$$\mathbf{B}_{\mathbf{t},\text{supply}} \to V_B = \max\{V \mid 0 < P_t^S(V) < P_{max}\}, P_B = P_t^S(V_B) \tag{4}$$

(notice again the strict inequalities). The special segment $\mathbf{C}_{\mathbf{t},\text{supply}}$ is defined as the set of all points in between, including $\mathbf{A}_{\mathbf{t},\text{supply}}$ and $\mathbf{B}_{\mathbf{t},\text{supply}}$.

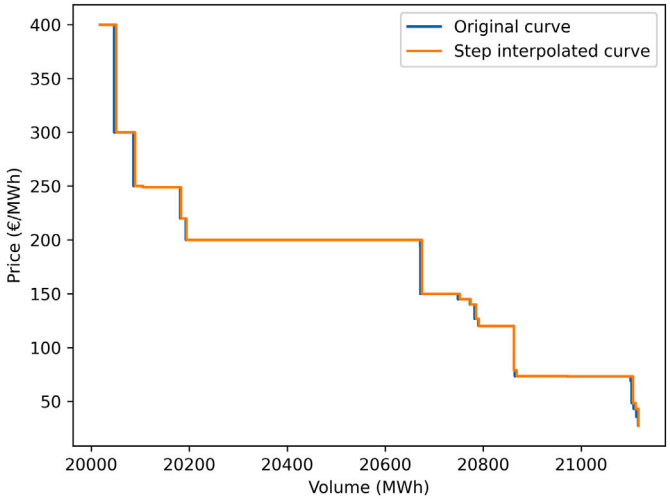


Fig. 7. Comparison of the original $\mathbf{C}_{t,demand}(V)$ with the equispaced $\hat{\mathbf{C}}_{\mathbf{t},\mathbf{demand}}(V)$ for hour 9 of January 1 st, 2018, obtained by merging the definition volume grids into one grid.

4.8. Decoupling the prices from the volumes: $\hat{C}_t$

One of the major difficulties in modeling and forecasting the curves is that their interesting segment swings back and forth in time (with weekly seasonality, at least), while displaying a changing complicated stepped structure within itself. This effect is particularly strong in the case of the demand curve, as it was seen in Fig. 4. In that case, the volume interval in which the interesting segment is indeed very small, whereas the overall motion spans a very large volume segment—that is, the small interesting part of the support itself varies a lot.

That problem can be alleviated by making the following modeling assumption, which is especially useful for demand. Within the $\mathbf{A_t}$ to $\mathbf{B_t}$ window, the very local variation which develops over time is assumed to be independent of where the window lies. This variation can hence be captured in the following way:

1. Define a uniformly spaced grid on V of 200 points (a value obtained by trial and error) between $\mathbf{A_t}$ and $\mathbf{B_t}$, the same for all days and hours.
2. Use step interpolation (because of IPEX regulations) to define P values on this uniformly spaced grid. The resulting segment $\hat{\mathbf{C}}_\mathbf{t}(V)$ will be called the 'equispaced segment'. Notice also that the original data points are not equispaced in V.
3. Discard the grid, and keep only the price part of $\hat{\mathbf{C}}_\mathbf{t}(V)$, that is, a vector to be called $\hat{C}_t$, thus disentangled from the volume. Notice that in principle the equispaced segments can vary in length (but not in number of points) as time t runs on. This is because the distance between $\mathbf{A_t}$ and $\mathbf{B_t}$ varies, but gets always divided in the same number of parts.

An example of what one can get with step 2 of this operation is shown, for one of the days in the dataset (hour 9 of January 1, 2018), in Fig. 7. It can be seen that the price vector $\hat{\mathbf{C}}_{\mathbf{t},\mathbf{demand}}(V)$ is only slightly different from $\mathbf{C}_{\mathbf{t},\mathbf{demand}}(V)$, when both are plotted on the merge of equispaced and non-equispaced grids. After step 3 of this procedure, a dataset of equisampled vectors $\hat{C}_{t,demand}$ and $\hat{C}_{t,supply}$ is obtained for the entire dataset.

As shown in Fig. 8 for the case of demand, these vectors do not shift so much widely on the fixed and equispaced support, especially in comparison with what happens in Fig. 4. This can make forecasting much easier. For all t, $\hat{C}_{t,demand}$ and $\hat{C}_{t,supply}$ are defined on the same sequence of indices of the grid, and these indices are not related to volumes anymore.

As anticipated in Section 4, the mild assumption made here, statistically supported, can be made even more explicit. The procedure consists of three steps. The first step is to extract from each curve the window where most action is concentrated. The second step is to mark its bounds (the $\mathbf{A_t}$ and $\mathbf{B_t}$ points at the ends of the interesting segment $\mathbf{C_t}(V)$). The third step is to assume that the prices' pattern of each window is somewhat invariant with respect to dilatation and compression of the related volume segment boundaries. This is what Fig. 8 hints to. Using the same grid for all interesting segments in the dataset means that real data in each segment is dilated or compressed according to the distance between the V components of $\mathbf{A_t}$ and $\mathbf{B_t}$. The full curve forecasting problem, which includes the dynamics of the absolute positions of $\mathbf{A_t}$ and $\mathbf{B_t}$ and the dynamically relative positions of the points inside the box (actually, only the dynamics of P with respect to the fixed grid), will hence be reduced to that of forecasting the box position and that of forecasting the (thus reduced) internal dynamics of the box. After the next Section 4.9, namely in the coming Section 4.10, it will be shown how to further reduce the residual complexity of this forecasting problem by the use of the monotonic autoencoder.

4.9. Forecasting: the enhanced ESN

As discussed in Section 4.4, one of the machine learning structures used in this work is the proposed EESN. It is built in the following way.

As said, the EESN is an enhancement of the vanilla ESN. It not only looks at the history of the time series data as standard ESNs do, but also incorporates external factors (like past, current, and future deterministic exogenous information) that could affect the outcome. In the presented case, the EESN uses holiday timing information to enhance the forecast of $\mathbf{A}_t$, of $\mathbf{B}_t$, and of each individual element of the autoencoder encodings. For example, when the EESN forecasts a future $\mathbf{A}_t$, it doesn't just rely on patterns from the past, but also it adds that, conditionally on whether today is a holiday and tomorrow is the same, the volume value of $\mathbf{A}_t$ should be low. In a sense, this helps peep 'legally' into the future. By combining the memory of past patterns with these external influences, the EESN provides a more accurate forecast than an ESN.

The output $\hat{y}_{t,ESN}$ of the ESN is enriched by means of the following regression (notice, not autoregression) approach. The standard regression of a variable both w_t on the regressor x_{t-1} for which future values cannot be known, and on the deterministic sequence z_t for which future values can be known, can be written as

$$w_t = \beta_0 + \beta_1 x_{t-1} + \beta_2 z_{\text{any } t} + \epsilon_t, \tag{5}$$

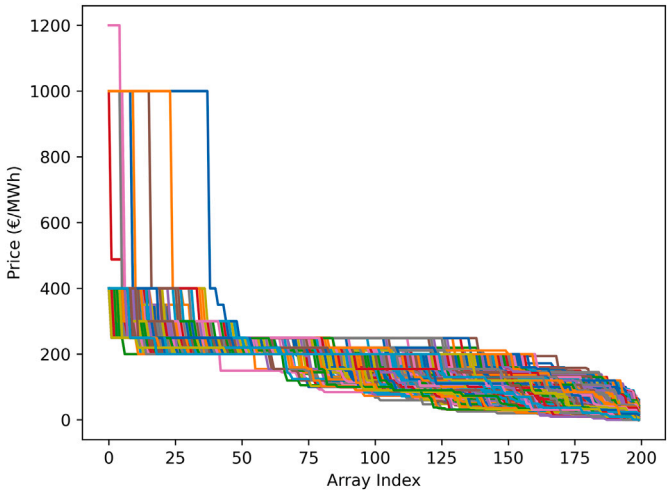


Fig. 8. Disentangled prices $\hat{C}_{t,Demand}$ plotted as vectors against the equispaced array's 200 indices, for the entire dataset. See text.

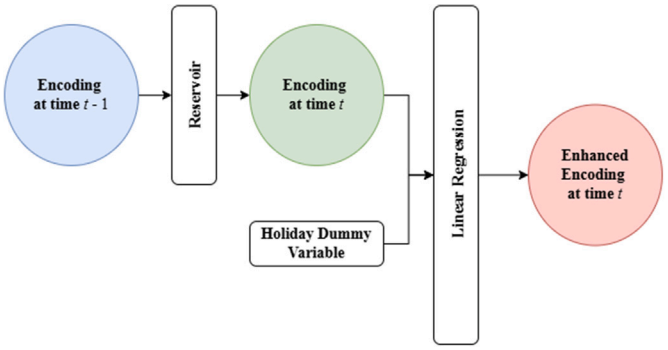


Fig. 9. Architecture of the enhanced ESN (EESN).

where ϵ_t is a sequence of i.i.d. Gaussian variables. Applying Ordinary Least Squares, the aim of the regression is to find estimates $\hat{\beta}_0$, $\hat{\beta}_1$ and $\hat{\beta}_2$ of β_0, β_1 and β_2 by finding the argmin of the sum of terms like

$$\left(w_t - \beta_0 - \beta_1 x_{t-1} - \beta_2 z_{\text{any } t}\right)^2. \tag{6}$$

Be H the set of all public holidays and weekends in Italy, and $\mathbb{1}_{t\in H}$ an indicator function for whether t belongs to H. At time t (that is, day d for a fixed hour h), for a scalar time series of data values $\bar{y}_t$ and for a forecasting horizon $f = 1$:

1. obtain from the series the ESN forecast $\hat{y}^f_{\text{t-1, ESN}}$ issued at $t-1$ for today t using a vanilla ESN. The hyperparameters of the ESN are the same for all 24 h. They are found during validation to be: reservoir size, 80 (fixed); leaking rate, 1.0 (no past activation required); spectral radius, 0.9995;
2. use Eq. (6) to regress the ground truth of today $\bar{y}_t$ (as w_t) on the forecast for today $\hat{y}^f_{\text{t-1, ESN}}$ (as x_{t-1}), and on $\mathbb{1}_{t\in H}$ (as $z_{\text{any } t}$), then obtain $\hat{\beta}_0$, $\hat{\beta}_1$ and $\hat{\beta}_2$;
3. with the estimated coefficients define the EESN forecast $\hat{y}^f_{\text{t-1, EESN}}$ made yesterday at $t-1$ for today at t as

$$\hat{y}^f_{\text{t-1, EESN}} \equiv \hat{\beta}_0 + \hat{\beta}_1 \hat{y}^f_{\text{t-1,ESN}} + \hat{\beta}_2 \mathbb{1}_{t\in H}. \tag{7}$$

The architecture of the EESN is shown in Fig. 9. Notice that a $\beta_2 \mathbb{1}_{t\in H}$ term could also be added inside some of the very many reservoir neuron components. However, because of the highly nonlinear environment of the reservoir, the effect would not be controllable and almost meaningless.

4.10. Forecasting: the monotonic autoencoder, plus refinement

As discussed in the literature review presented in Section 3, many of the works on curve forecasting use dimensionality reduction [13,14,35]. This is because, already at a first sight, part of the curve data looks redundant. In the demand curves, for example, price variation is concentrated over just a narrow range of V and P values, whereas only one price value like $P_{\max}$ or $P_{\min}$ corresponds to volumes outside the interesting segment. Discarding these 'outside' points would hence have no impact on the shape of the curve. The supply curves also contain several steps where the price P remains constant over a range of bid volumes. Discarding points 'internal to these internal steps' would have no impact as well on the shape of the curve. That is to say that the shape of aggregated curves $P(V)$ is actually related to only a small fraction of all available data points $p(v)$. Said another way, volume aggregation washes out a lot of information contained in the pre-aggregation dataset on which forecast is based. This is in contrast with the expectation that accurate forecasts should make use of all information available. Hence, dimensionality reduction techniques are used in the hope of automatically capturing just relevant pieces of information, while still using all

pre-training information. However, the issue of what is the best balance in information selection has not been resolved in the literature so far, and the presented framework proposes a way in between.

Noticeably, by means of the use of the points $\mathbf{A_t}$ and $\mathbf{B_t}$ and the vector $\hat{C}_t$ discussed above, information present in the curve is already compressed to some extent. However, it is not yet compressed to a satisfactory level. Actually, within $\hat{C}_t$ there still seems to be structure, the internal steps, which can surely be further reduced to some extent. Yet, this reduction must be made with care, because the profile of $\hat{C}_t$ is monotonic for both demand and supply, and such monotonicity must be preserved. It is then hoped to compress a further amount of pre-training information by means of an autoencoder, but this autoencoder must be designed in such a way as to preserve monotonicity.

However, in the literature, truly monotonic autoencoders are not directly available. There are some papers that discuss how to achieve a bias towards monotonicity within the vector output of generic vector neural networks (that is, cross-sectionally), as it is discussed in Ref. [36], Ref. [37], and more in general in Ref. [38]. Some other papers discuss how to obtain dynamic scalar sequences of points that monotonically increase or decrease in time by means of autoencoder techniques (that is, longitudinally) [39–41]. Ref. [42] actually implements autoencoder monotonicity, but only in the decoder part. Hence, our setup will start from the general ideas discussed in Ref. [38] for biased generic neural networks, and develop them to achieve guaranteed (hence, not just biased towards) cross-sectional monotonicity, being specifically inspired by Fig. 3 therein.

As a first step, the proposed monotonic autoencoder will encode the sequence $\hat{C}_t$, while trying to maintain constant the sign of the sequence changes in the output of the network. This will generate an internal, reduced-dimension representation of monotonic $\hat{C}_t$, while biasing towards monotonicity. The compression chosen will reduce $\hat{C}_t$ from its 200 entries to the 50 entries of its internal encoding vector (this compression ratio was obtained as computationally optimal by trial and error). The autoencoder will be trained on the training plus validation dataset segment before the rest of training is performed. This is what we call pre-training, aimed to capture the basis. The reduced-dimension representation will be then forecast and back-transformed to the forecast of $\hat{C}_t$. As a second step, a refinement will be operated on the autoencoder output in order to guarantee monotonicity in full.

The autoencoder used has five layers (two in the encoder and two in the decoder), and the number of neuron units used for our architecture is

- Encoder: 200 (input size) → 128 → 64
- Decoder: 64 → 128 → 200.

More in detail, a special loss function is applied to the autoencoder network, in which a regularizing term is added to a standard autoencoder loss. Given N as the number of entries of the equispaced interesting segment vector, each entry can be labeled as $\hat{C}_{t,i}$ where $i = 1, \dots, N$. $\hat{C}_{t,i,\text{demand}}$ and $\hat{C}_{t,i,\text{supply}}$ can be defined consequently. A monotonicity vector M_t can be defined as

$$M_t = (\hat{C}_{t,i+1} - \hat{C}_{t,i} | \quad \text{for} \quad i = 1, 2, \dots, N-1). \tag{8}$$

$M_{t,demand}$ and $M_{t,supply}$ can be defined consequently for the specific cases of demand and supply. M_t contains the first-differences along the vector $\hat{C}_t$, and, when computed on a monotonic vector, M_t should have values less than or equal to zero for decreasing functions, and greater than or equal to zero for decreasing functions. Using M_t, two different loss functions, L_{demand} and L_{supply} can hence be defined, for demand and supply curves respectively. For demand curves the value of L_{demand} at time t is defined as

$$L_{t,demand} = \frac{1}{N}\sum_{i=1}^{N}(\hat{C}_{t,i,\text{demand}} - C_{t,i,\text{demand}})^2 + \frac{\lambda}{N-1}\sum_{i=1}^{N-1}\max(0, M_{t,demand}), \tag{9}$$

where the first term has the usual MSE form, and λ in front of the second (regularizing) term is a hyperparameter that controls the amount of 'pressure' on the loss towards of monotonicity. For supply curves, $L_{t,Supply}$ is defined in analogy as

$$L_{t,supply} = \frac{1}{N}\sum_{i=1}^{N}(\hat{C}_{t,i,\text{supply}} - C_{t,i,\text{supply}})^2 + \lambda\frac{1}{N-1}\sum_{i=1}^{N-1}\min(0, M_{t,\text{supply}}). \tag{10}$$

When minimized, $L_{t,demand}$ encourages the 'monotonized' $\hat{C}_{t,\text{demand}}$ to be monotonically decreasing, since $M_{t,\text{demand}}$ is pushed downward towards 0 by the last term of Eq. (9). When minimized, $L_{t,supply}$ encourages the 'monotonized' $\hat{C}_{t,\text{supply}}$ to be monotonically increasing. The value of λ is chosen to be 1, after a grid search in $[0.01, 2]$. The autoencoder has an encoder part whose input is $\hat{C}_t$, and whose output is an encoding e_t (that is, the embeddings $e_{t,demand}$ or $e_{t,supply}$, respectively). It also has a decoder part whose input is e_t and whose output could also be called $\hat{C}_t^A$, where the superscript A denotes the effect of the monotonic autoencoder. Using this autoencoder on a training set of disentangled interesting segments $\hat{C}_t$, one can hence create a time series of corresponding training encodings e_t, which are compressed vectors corresponding to the 'almost monotonized' vectors. Subsequently, on this training set we can train and use the EESN defined in Eq. (7). Then, on a test set, Eq. (7) will allow to forecast independently each entry of e_t obtained inside the autoencoder (fed with the test $\hat{C}_t$). This will deliver the compressed vector e_t^f (that is, $e_{t,demand}^f$ and $e_{t,supply}^f$), as shown in Fig. 5. Thus, the pre-trained decoder is used to construct a 'monotonized' forecast $\hat{C}_t^{Af}$ from the forecast e_t^f.

At this first step monotonicity is yet just encouraged, but not guaranteed. This can be seen numerically and graphically in Fig. 10, where the entire architecture of the monotonic autoencoder is sketched, and where the outputs of the model at each stage are shown above each related component. Specifically, by looking at the third graph on the right, just above the output, it becomes clear that this output is not necessarily monotonic in full.

At the end of the following second step, monotonicity of the final forecast is on the contrary fully ensured by the following refining procedure, which uses the computed $\hat{C}_t^{Af}$ just as a guiding reference:

1. The $K = 5$ closest neighbors N_k of $\hat{C}_t^{Af}$ are picked up from the original (then, step-like) pre-training set data $\hat{C}_t$ (actually from the training segment). How close these neighbors are is evaluated by using the L_1 distance [43] (also called Manhattan distance) $d(\hat{C}_t^{Af}, N_k)$ between $\hat{C}_t^{Af}$ and N_k.
2. The final, guaranteed monotonic, refined forecast is computed as the distance-weighted average of the neighbors as

$$\hat{C}_t^f = \sum_{k=1}^{K}\frac{N_k}{d(\hat{C}_t^{Af}, N_k)}. \tag{11}$$

Now the forecast $\hat{C}_t^f$, built as guided by $\hat{C}_t^{Af}$, is truly monotonic. This is because all the constituent terms of the sum in Eq. (11) are monotonic, being a ratio between N_k, which is monotonic, and a positive factor lower than 1, so that their combination is necessarily monotonic. In passing, note that the five neighbors are chosen from the pre-training data, and hence, no future information is used in the forecasting process when run on the test set. This procedure is graphically sketched in Fig. 11.

Recapitulating, our monotonic autoencoder approach to the forecast of the interesting segment is based on: 1. identifying an internal basis by means of the modified autoencoder; 2. using this basis to generate the compressed embeddings corresponding to segments; 3. using the EESN to forecast the embeddings; 4. using the decoding part of the autoencoder to generate quasi-monotonic forecasts of the segments

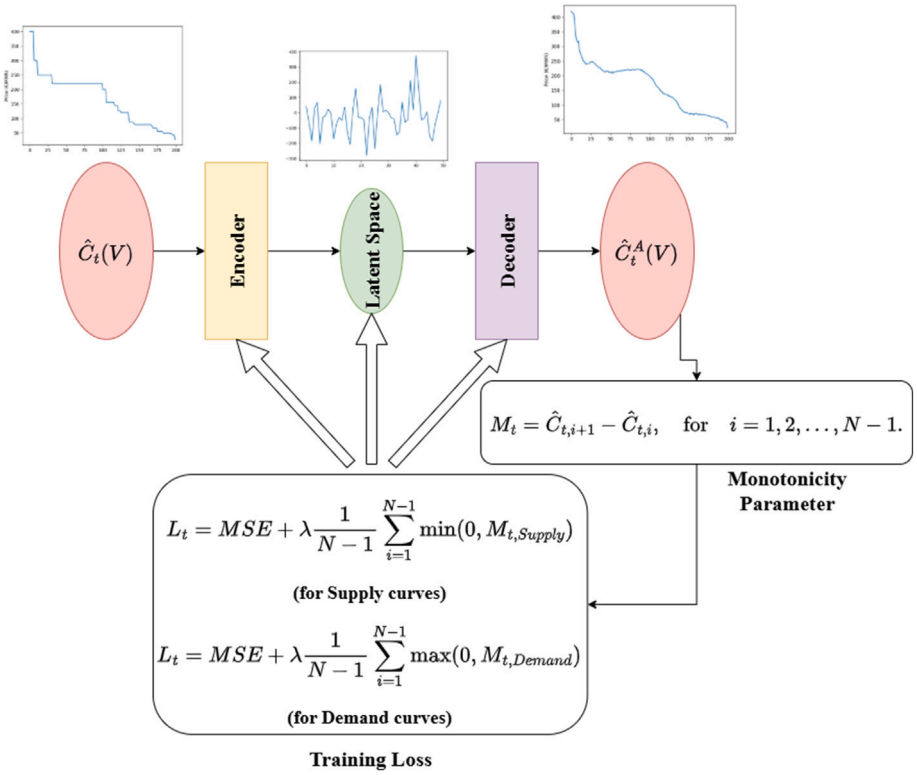


Fig. 10. Architecture of the monotonic autoencoder as related to its use in the first step. Notice the input and output shapes of $\hat{C}_t$ and $\hat{C}_t^A$.

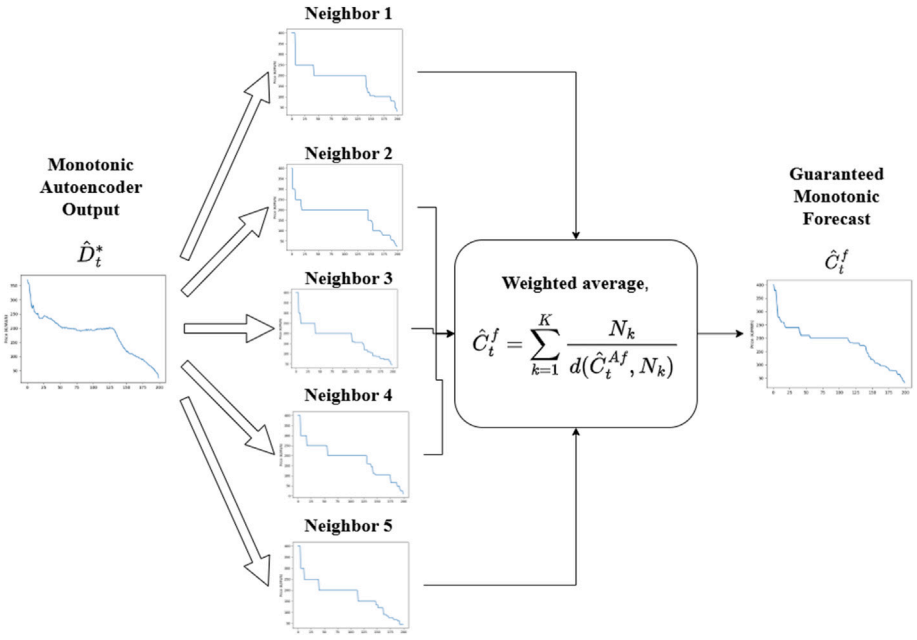


Fig. 11. Monotonic autoencoder second step. Creating a monotonic approximation from the autoencoder output.

which will act as a 'driver'; 5. refining these forecasts by means of the first-neighbor procedure (guided by the 'driver' $\hat{C}_t^{Af}$), taking them to guaranteed monotonicity. All this procedure doesn't use functional methods, unlike most of the related literature, and mainly relies on sequence representation of the curves.

4.11. The assembling of the curves given the forecasts $\hat{C}_t^f$, $\mathbf{A}_t^f$ and $\mathbf{B}_t^f$

As a final step, given the individual forecasts $\hat{C}_t^f$, $\mathbf{A}_t^f$ and $\mathbf{B}_t^f$, the curves need to be assembled. Demand and supply need a slightly different procedure.

4.11.1. Demand curves

Given the forecasts $\hat{C}_t^f$, and consequently $\mathbf{A}_{t,demand}^f$ and $\mathbf{B}_{t,demand}^f$, the full demand curve is assembled in the following manner:

1. Define a volume vector, $V_{demand,assem}$, that is the concatenation of 0 and an equispaced grid of 200 points between $\mathbf{A}_{t,demand}^f$ and $\mathbf{B}_{t,demand}^f$.
2. Define a price vector, $P_{demand,recon}$, that is the concatenation of the value of P_{max} (corresponding to $V = 0$), and $\hat{C}_t^f$.
3. Concatenate the two vectors together as $(V_{demand,assem}, P_{demand,assem})$ to obtain the fully assembled curve.

4.11.2. Supply curves

Similarly, the full supply curve is assembled in the following manner:

1. Define a volume vector, $V_{supply,assem}$, that is the concatenation of 0, and an equispaced grid of 200 points between $\mathbf{A}_{t,supply}^f$ and $\mathbf{B}_{t,supply}^f$.
2. Define a price vector, $P_{supply,assem}$, that is the concatenation of P_{max} (corresponding to the V coordinate of $V = \mathbf{B}_{t,supply}^f$), and $\hat{C}_t^f$.
3. Stack the two vectors together as $(V_{supply,assem}, P_{supply,assem})$ to obtain the fully assembled curve.

4.12. Optimal integration of the pieces in the assembled forecast: HC-MAE

The pieces A, C and B of the stacked structure have been obtained so far by means of uncoupled autoregressions, thus these pieces don't share information with one another. This was also hinted at in the last part of Section 4. One can then try to optimize their assembling: this will be called 'integration'.

This optimized assembling can be obtained by defining a measure for the distance, or dissimilarity, between the assembled forecast curve and the ground truth data, and minimizing with respect to it. This step will also prepare parameters to be used in the out-of-sample test phase. This simple distance will be called Heterogeneous Curve Mean Absolute Error (HC-MAE) metric.

The calculation of the dissimilarity between two DAM curves poses however a problem. Consider two curves, both of the same type, demand or supply, one the forecast and the other the ground truth. The curves are hence taken at subsequent times, for example $t = 1$ and $t = 2$: in a sloppy notation, one would have forecast (V_1, P_1) and ground truth (V_2, P_2). Consider the task of computing something like the RMSE between these two curves. Although they have similar data (V, P), they may not have the same number of V values or even the same V values. Hence, the curves cannot be directly compared, because some points in one curve might not even have a direct counterpart in the other. The proposed procedure, which can also be used to compare the performance of forecasting models of different origins, is the following:

1. Input: Begin with the two curves, (V_1, P_1) and (V_2, P_2).
2. Union: Compute and sort the union of V_1 and V_2 to form V_u.
3. Define Grid: Define a grid of U equispaced points that span from $\min(V_u)$ to $\max(V_u)$. Label this grid as V_{common}.
4. Interpolate: Perform step interpolation for P_1 and P_2 over this same V_{common}, obtaining the corresponding interpolated values. These could be called Π_1 and Π_2, defined on the same grid, and can be compared.
5. Compute the metrics: Calculate the Mean Absolute Error between Π_1 and Π_2 as follows:

$$\text{HC-MAE} = \frac{1}{U}\sum_{i=1}^{U} |\Pi_{1i} - \Pi_{2i}|.$$

This simple error against ground truth will be considered the error of the forecast. It is just a Mean Absolute Error adapted to the curve forecasting problem, a problem for which a standard MAE cannot be used.

Noticeably, in this procedure the choice of U is crucial as it determines the resolution at which the two curves are compared. A high value of U gives a comparison similar to calculating the area between the curves, though it has a significant computational cost. Conversely, if U is too low, the error metric may fail to capture local differences between the curves, resulting in a less accurate representation of their dissimilarity. In the experiments for this study, U was taken as 10,000, a number high enough to capture the local differences and low enough to not incur a high computational cost.

Going back to the task of optimally obtaining the assembled curve, one can also exploit in this computation some information about past lags again, doing it in four steps.

For clarity, with a slight abuse of notation in which special point symbols like $\mathbf{A}_{t-1}^f$ stand for their V coordinate:

1. Combine the forecasts made yesterday for today $\mathbf{A}_{t-1}^f$ and $\mathbf{B}_{t-1}^f$ with $\mathbf{A}_{t-7}$ and $\mathbf{B}_{t-7}$:

$$\mathbf{A}_{t-1,combined}^f = \theta_0 + \theta_1 \mathbf{A}_{t-1}^f + \theta_2 \mathbf{A}_{t-7} \tag{12}$$

$$\mathbf{B}_{t-1,combined}^f = \theta_0 + \theta_1 \mathbf{B}_{t-1}^f + \theta_2 \mathbf{B}_{t-7} \tag{13}$$

Namely, the previously made forecast $\mathbf{A}_{t-1}^f$ is adjusted by replacing it with a weighted sum of (the volumes of) $\mathbf{A}_{t-1}^f$ and one week prior data $\mathbf{A}_{t-7}$. Similarly, the previously made forecast $\mathbf{B}_{t-1}^f$ is adjusted by replacing it with a weighted sum of $\mathbf{B}_{t-1}^f$ and one week prior data $\mathbf{B}_{t-7}$, using the same coefficients θ_0, θ_1 and θ_2 from Eq. (12). This distributes information (otherwise obtained individually) between the two forecasts. This combination bolsters the forecast by repeating past information directly. Moreover, in a sense, this direct injection of information from $t-7$ in a higher layer of a forecasting hierarchy can be considered similar to the stabilizing use of skip layers in residual neural networks [44].

2. Do the same with $\mathbf{C}_t$, but with other coefficients.

$$\mathbf{C}_{t-1,combined}^f = \theta_3 + \theta_4 \mathbf{C}_{t-1}^f + \theta_5 \mathbf{C}_{t-7} \tag{14}$$

3. Assemble from $\mathbf{A}_{t-1,combined}^f$, $\mathbf{B}_{t-t,combined}^f$ and $\mathbf{C}_{t-1,combined}^f$ the demand and supply curves using the assembling algorithms described in Sections 4.11.1 and 4.11.2, respectively. The forecasts will contain the parameters θ_j, where $j = 0, \ldots, 7$.
4. Minimize the HC-MAE (between the forecasted and ground truth curves) with respect to the parameters θ.

The parameters θ_j of Eqs. (12)–(14) are learned using 'differential evolution' [24], an evolutionary optimization algorithm that can optimize non-differentiable objective functions, because the HC-MAE is non-differentiable.

From the optimal integration of $\hat{C}_M$ with $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$ in the training plus validation set, the parameters θ are hence obtained. They can be in turn used to assemble the forecasted curve in the test set. The HC-MAE, on its part, can be used to assess quality of these out-of-sample forecasts.

4.13. Forecasting methodology, recapitulated

In the proposed framework, forecasting a DAM curve as partly a sequence and partly an interpolation curve, is hence divided into four steps, namely:

1. Single out and model the interesting segment $\mathbf{C}_t$ of the curve, that is, the curve segment between the volume coordinates of the special points $\mathbf{A}_t$ and $\mathbf{B}_t$. Then forecast the volumes related to the points $\mathbf{A}_t$ and $\mathbf{B}_t$.
2. For the segment $\mathbf{C}_t$, disentangle the prices P from the volumes V, to obtain individual vectors $\hat{C}_t$ instead of curves.

3. Compress these vectors $\hat{C}_t$ to lower dimensional encodings via a monotonic autoencoder. The encodings are forecasted to obtain the curve at a future time step, and the decoder is used to reconstruct the curve. To fully ensure the monotonicity, the distance-weighted average of the five nearest neighbors of the forecasted interesting segment is obtained. Averaging is made according to Eq. (11). Step interpolation, appropriate for the IPEX data, is used in some of the computations. Since the extrema of $\hat{C}_t$ correspond to the P coordinates of $\mathbf{A}_t^f$ and $\mathbf{B}^f$, this completes the forecast of the special points.
4. In order to assemble the curve from the A, B, C pieces, perform an optimal integration of $\mathbf{C}_t^f$ with $\mathbf{A}_t^f$ and $\mathbf{B}^f$ by means of the parameters θ. These parameters are estimated in the validation set, and in the test set they are used as they were found in the validation set.

At training, step 1 allows for finding the EESN coefficients of the special points regressions, step 3 allows for estimating the monotonic autoencoder and finding the coefficients of its forecasting regressions, and step 4 allows for finding the parameters θ. At testing, EESN and autoencoder parameters, and the parameters θ found in the training phase are applied.

For clarity, it is to be recalled here that the architecture of the entire framework is shown in Fig. 5, and the relationship between the use of the dataset train, validation and test segments and the four steps described above is shown in Fig. 6. Notice that the combination of the autoencoder part and the EESN part forms a spatiotemporal module (the EESN modules are just temporal modules). This is because this part encapsulates both spatial information via the autoencoder, and temporal information via the EESN.

This setup allows the model to produce forecasts that are both monotonic, which is a feature not found in the literature, but still adaptable to external influences like exogenous drivers. As discussed in Section 4.5, the use of ESNs instead of standard RNNs makes computation times acceptably fast.

5. Experimental Setup

Numerical experiments will now be discussed aimed to compare the efficacy of the proposed approach to the most important foundational models present in the literature.

5.1. Dataset

As anticipated at the beginning of Section 2, the dataset used in this study consists of all available individual IPEX DAM bids for the 24 h of a given day, specific to the NORD zone, recorded by the GME for 730 days from January 1, 2018, to December 31, 2019. In Appendix A some more details are presented on how to download these data from the link in Ref. [20]. These raw data include all bid information necessary to create the aggregated curves as explained in Section 2. The way in which the dataset is partitioned in training, validation and test set was discussed in Section 4.6 and shown in Fig. 6. The chosen time frame depends on the fact that in August 2020 the IPEX market changed its regulation to shift P_{min} to -500 euros, and we wanted to perform the study on a fixed background of common P_{max} and P_{min} for all days without structure breaks.

Moreover, succeeding events like the COVID pandemic and the Ukraine invasion made European electricity time series much more complex, requiring further forecasting structure. Since we prefer to introduce and discuss first our forecasting framework in its original form, and compare it with benchmark models not developed with these new scenarios in mind, we defer an analysis of more recent data to next proposals. As context information, it should also be added that a total of 887 extreme weather events were recorded in Northern Italy in this time period [45], most notably a storm on October 29, 2019 [46]. However, these rare events do not show up in the dataset as price or volume outliers.

5.1.1. Benchmark models

For comparison, six models which we consider the most representative for this kind of studies were used as benchmarks. Their names are listed in the first row of Table 2, placed at the top of the columns as labels. They are the naive model, the smarter naive model, the PCA plus LASSO approach, the PCA plus LSTM approach, the functional-stochastic FAR model, and the X-model. These models were discussed in the literature section, Section 3, except for the two naive and smarter naive models to be discussed hereafter. Notice that, as discussed in Section 3, the X-model was not originally designed for curve forecasting, and the results presented here come from an adaptation extrapolated from the original paper.

In scalar electricity daily series autoregression forecasting of price or volume, an effective benchmark model, very difficult to beat and largely used, is the so-called 'naive' (or persistence) model [1]. Given the scalar data daily series $\{x_t\}$, the naive forecast issued at $t-1$ for today at t is obtained from the current value of the series, as $x_{t-1}^f = x_{t-1}$. The 'smarter naive' forecast is intended to improve the 'naive' forecast. It is given by $x_{t-1}^f = x_{t-7}$ if t is Saturday, Sunday or Monday, and $x_{t-1}^f = x_{t-1}$ in the other cases. This is because weekend day prices or volumes have a remarkably different behavior from other days and cannot be well predicted with the values of the other days type. If for example today is Saturday (a weekend day), yesterday's (that is, Friday's, a working day) demand is not too much suitable to forecast it, whereas Saturday's demand from one week ago can be more suitable. The same happens for Sunday (weekend against working day) and Monday (working day against weekend). In the case of daily curve forecasting a similar approach can be taken. In this case we can define the naive model as

$$X_{t-1}^f = X_{t-1} \tag{15}$$

where X_{t-1}^f represents the forecast issued at $t-1$ for the today's time t, obtained as a copy of X_{t-1}, the data curve of yesterday. Notice that the ground truth X_t and its forecast X_{t-1}^f are most surely defined on different volume points, hence they cannot be directly compared point by point. This is, again, why we have to use a special measure like the HC-MAE interpolation procedure for defining a meaningful forecasting error. In the same way the smarter naive model can be defined as

$$X_{t-1}^f = \begin{cases} X_{t-7} & \text{if } t \in \text{Saturday, Sunday, Monday,} \\ X_{t-1} & \text{otherwise,} \end{cases} \tag{16}$$

using notation in analogy to the scalar case.

Besides these two standard benchmarks, the third benchmark is a slightly modified version of the 'PCA plus LASSO linear autoregression' model proposed by Pelagatti in Ref. [13]. Since that model was primarily proposed for supply curves, it is modified so that it can forecast demand curves too (because its monotonically decreasing, not increasing).

The fourth model is a modified version of the 'PCA + LSTM model' proposed by Guo et al. in Ref. [7]. Since that model forecasts primarily supply curves, and specifically from the PJM market, the data preprocessing procedure of Ref. [13] was applied before applying PCA and, subsequently, LSTM.

The fifth model is the FAR stochastic functional autoregression proposed by Shah and Lisi in Ref. [11]. In this case, linear regression was used over a uniform grid of 500 points, and then the *far* [47] package of R was used to forecast the curves directly.

The sixth model follows the X-model proposed by Ziel et al. in Ref. [15]. It should be however noted that a key assumption of the X-model is that there must be at least one bid within each price class. This makes the number of price classes significant in the model's operation. As to that, one should notice that the number of price classes which one can get from the IPEX dataset restricted to the NORD zone is maybe too much small in the specific case of IPEX demand curves. This makes the applicability of the X-model somewhat limited.

6. Results

After training the proposed model in the training and validation set independently for the 24 h, the HC-MAE for the test set was computed for each model, with each hour's error averaged over the 180 days of the test set.

6.1. The curves as a whole

The results for demand curves are shown in Table 2. According to the HC-MAE proposed metrics these results show that, already at a first sight, the proposed model outperforms the FAR in capturing the whole shape of the curve, also because FAR violates very often monotonicity in large amount. The proposed model also outperforms the X-model, which in turn outperforms by large the PCA based models. Moreover, the proposed model gives better results than the naive and the smarter naive models (which of course never violate monotonicity).

For supply curves, the average HC-MAE results are shown in Table 3. Like in the demand curve case, the proposed model outperforms, in order, FAR, the X-model, and both the PCA-based models. It also outperforms both the naive and the smarter naive models.

A comparison with respect the more traditional Mean Absolute Error metrics will be discussed in the next section, Section 7.

More in detail, for demand curves the smarter naive model gives a performance comparable with that of the naive model for the night hours 1 to 6. For daylight hours from 7 to 24, the smarter naive model outperforms the naive model, especially for hours 7–18, which are typical working hours. This shows that the seasonal nature of the smarter naive model allows it to account for the sudden drop in demand from Fridays to the weekends and the sudden increase in demand from the weekends to Mondays. In addition, this highlights the general importance of keeping into account at least lag 7 in modeling these data. The PCA + LASSO and PCA + LSTM models do not perform well in this case, for all hours. Our proposed model in turn outperforms all the benchmark models significantly for all the hours.

For supply curves, the naive and smarter naive models have comparable performance for most hours, with each model outperforming the other in some hours. This is because the weekly pattern is much weaker in the supply curves, because most of the stochasticity in supply comes from renewable energy such as solar or wind energy, which is not linked to human weekly cycles. Even in the case of supply curves, the proposed model outperforms all benchmarks.

In order to better perceive where forecasting errors are generated, Fig. 12 shows a sample of the forecast demand curves for a choice of representative hours for the specific day of the 15th July, 2019, mainly focusing on the proposed model. In order not to clutter too much the Figures, the forecasts of the FAR (very bad) and the X-model are not reported, whereas the PCA+LSTM model is instead reported for two reasons. This is the most recent model available, and it is the only model using a deep learning structure, the LSTM network. Its poor score could indicate that deep architectures are not necessarily effective in this problem, or, at least, the depth of the structure has to be exploited in suitable parts of a model, like in the case of the autoencoder. However, no other deep learning models are available for curves, and developing them could be anyway interesting.

The smarter naive model is, as anticipated, very difficult to beat, but the proposed model competes, even in this specific difficult day, head to head with it, whereas all other models lag behind. From the figure, it is clear that the most difficult structure to forecast for all models is the beginning and ending points (that is, the special points) of the interesting segment. Missing their positions automatically generates large errors. This visual comparison should hence make more clearly compelling one of the main reasons of the presented approach. With the same spirit, Fig. 13 shows a sample of the forecasted supply curves for 15th July, 2019 for the same selected hours.

6.2. The interesting points

Since the proposed model has multiple parts that get optimally integrated in the final integration step, the performances in some of the steps are now compared and analyzed. A key step of the proposed framework is the forecasting of the V coordinate of the points $V_{\mathbf{A}}$ and $V_{\mathbf{B}}$. Good forecasts for $V_{\mathbf{A}}$ and $V_{\mathbf{B}}$ should intuitively lead to a better result for the overall curve forecast.

Tables 4 and 5 show comparisons by the Mean Average Error (MAE) of the forecast of $V_{A,Demand}$ and $V_{A,Supply}$ respectively. For demand curves, according to MAE, the smarter naive model, EESN, and the integrated model, have the best forecasts. However, the proposed model has a better curve forecasting result than the smarter naive model, even though it has a worse forecast for $V_{A,demand}$. In Table 5 a similar trend can be

Table 2
HC-MAE value comparison on the 24 h of the forecasting models discussed in Section 5.1.1 for the demand curve.

Hour	Naive	Smarter naive	PCA + LASSO	PCA + LSTM	FAR	X-Model	Proposed model
1	118.87	127.28	243.83	197.22	175.91	125.66	70.79
2	119.85	128.74	207.82	196.95	180.21	127.95	71.48
3	119.82	128.59	242.69	193.76	181.01	129.18	72.62
4	116.66	127.1	204.27	179.44	179.63	128.6	74.93
5	115.17	124.58	160.3	172.13	176.48	122.58	75.17
6	134.28	113.06	181.62	185.5	181.54	119.49	76.51
7	176.5	96.85	282.45	244.39	242.46	129.75	77.79
8	210.3	93.58	227.19	242.53	253.05	142.55	75.47
9	227.34	87.15	276.01	274.22	261.74	152.63	75.91
10	222.1	87.36	336.97	296.94	271.11	151.81	74.94
11	216.34	89.56	250.27	263.49	271.38	150.71	74.59
12	211.81	89.55	375.27	281.61	279.66	149.35	79.52
13	198	88.72	330.8	265.83	274.79	144.4	77.1
14	211.13	88.47	348.55	290.32	294.94	151.9	81.13
15	221.18	88.29	361.29	302.29	303.87	157.32	80.17
16	224.8	89.41	335.45	283.07	306.54	158.57	83.25
17	219.82	90.02	358.44	309.29	302.77	154.66	85.82
18	205.38	89.12	329.14	283.47	287.29	147.44	85.53
19	186.25	83.9	313.14	265.62	250.95	136.21	76.23
20	166.12	79.17	263.63	230.24	231.56	124.12	73.7
21	153.15	80.65	217.08	204.77	221.29	126.79	72.9
22	142.68	78.64	221.19	200.56	223.71	127.66	73.68
23	130.22	77.56	206.87	184.08	223.64	124.74	69.84
24	124.18	76.08	231.76	191.38	194.49	124.07	69.76

Table 3
HC-MAE value comparison on the 24 h of the forecasting models discussed in Section 5.1.1 for the supply curve.

Hour	Naive	Smarter naive	PCA + LASSO	PCA + LSTM	FAR	X-Model	Proposed model
1	79.52	78.36	418.69	334.17	200.36	334.85	73.5
2	77.64	76.28	485.53	381.47	200.98	339.59	72.15
3	77.21	76.38	504.24	366.04	201.43	339.44	69.85
4	74.73	75.49	497.56	384.29	200.88	337.76	68.46
5	69.91	73.13	404.97	299.07	202.21	340.77	64.96
6	64.87	70.63	389.35	317.93	203.14	341.10	63.17
7	58.11	66.09	447.41	323.09	202.58	335.38	57.36
8	71.82	68.55	450.68	308.55	205.07	313.12	61.67
9	80.88	69.2	362.35	231.05	203.40	306.98	65.94
10	80.83	71.7	389.79	232.69	205.18	303.11	66.22
11	81.09	75.24	403.11	258.98	203.73	303.79	70.02
12	80.58	79.04	398.49	267.09	199.13	303.40	69.68
13	76.46	78.89	353.86	245.77	203.16	304.50	71.91
14	77.13	79.65	396.76	264.41	203.97	307.06	71.87
15	75.94	77.46	358.18	245.54	206.58	305.80	68.06
16	73.19	70.61	356.05	242.51	204.96	305.06	62.91
17	71.99	63.75	370.88	261.06	208.70	310.31	57.88
18	69.96	62.78	384.23	265.58	211.56	307.41	55.84
19	66.82	60.4	373.64	222.43	204.82	305.66	55.61
20	61.28	60.75	390.41	233.1	205.38	312.10	53.75
21	60.42	62.29	372.78	204.38	204.06	309.63	57.37
22	54.37	62.38	349.06	166.55	207.37	313.50	53.12
23	53.55	66.88	423.15	279.78	205.27	326.13	50.9
24	53.05	65.01	382.99	273.1	204.07	326.18	51.36

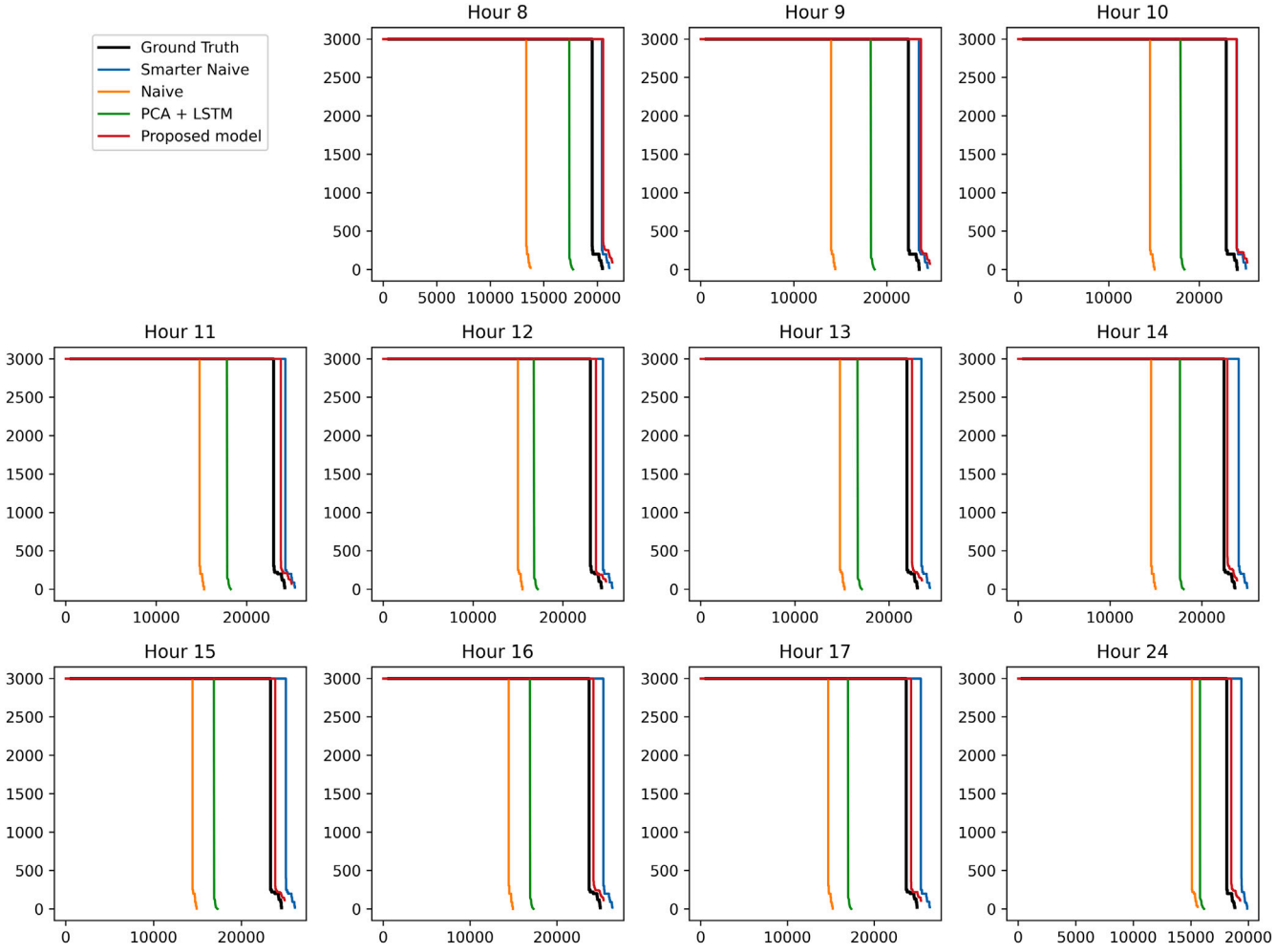


Fig. 12. Visual comparison of the forecasts for the demand curves on 15th July, 2019, for selected hours.

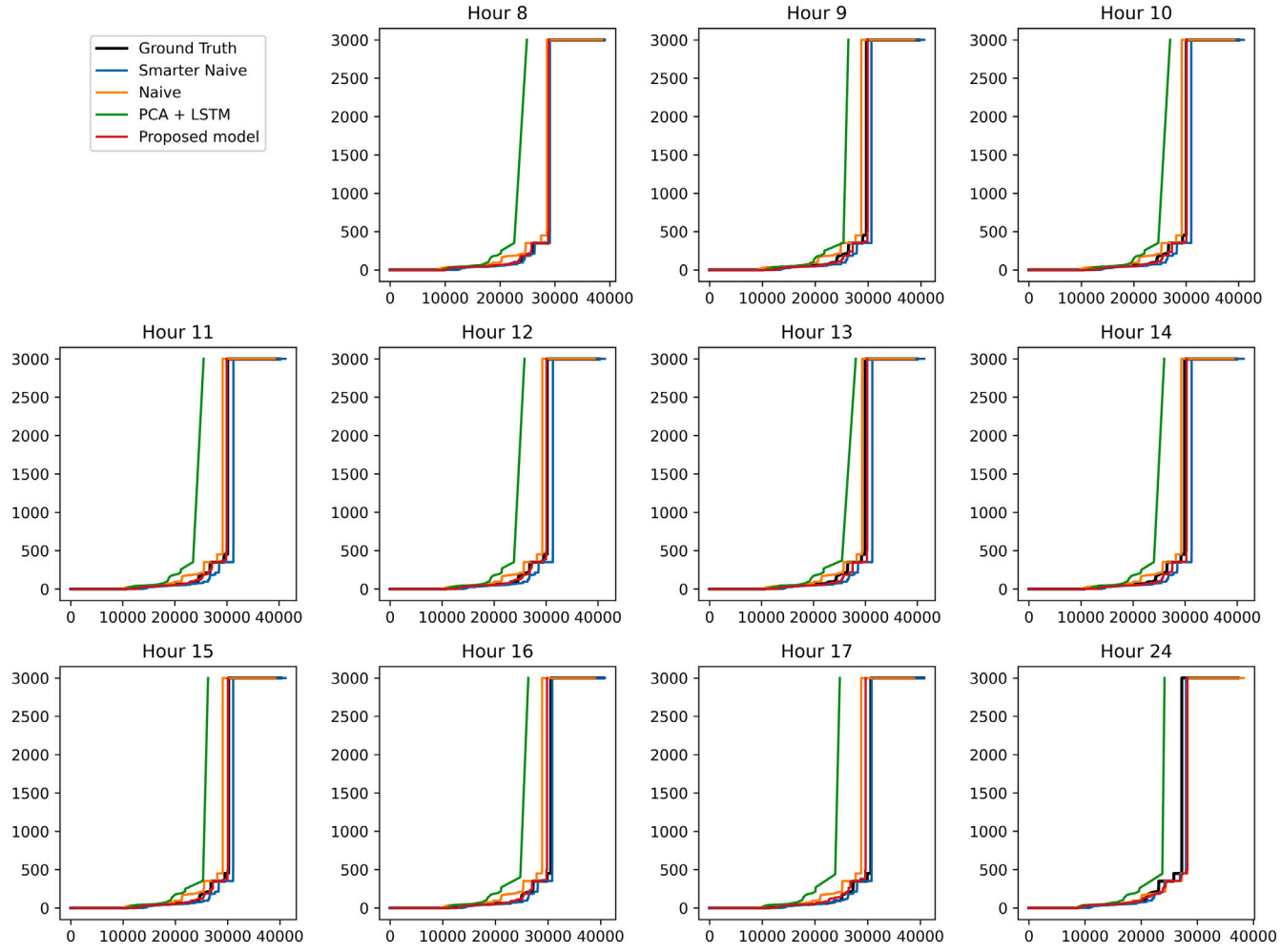


Fig. 13. Visual comparison of the forecasts for the supply curves on 15th July, 2019, for selected hours.

Table 4
MAE value comparison of the different models for $V_{A,Demand}$.

Hour	Naive	Smarter naive	PCA + LSTM	EESN	Proposed
1	867.0	773.0	1424.0	473.0	454.0
2	819.0	744.0	1345.0	455.0	439.0
3	782.0	725.0	1289.0	453.0	437.0
4	764.0	710.0	1252.0	456.0	433.0
5	776.0	695.0	1258.0	445.0	425.0
6	957.0	659.0	1286.0	484.0	476.0
7	1569.0	664.0	2026.0	619.0	650.0
8	2247.0	747.0	2757.0	759.0	816.0
9	2740.0	774.0	3309.0	903.0	955.0
10	2704.0	785.0	3385.0	887.0	949.0
11	2608.0	801.0	3113.0	859.0	974.0
12	2533.0	800.0	3150.0	935.0	980.0
13	2205.0	754.0	2835.0	803.0	845.0
14	2403.0	767.0	3200.0	874.0	928.0
15	2621.0	788.0	3481.0	924.0	951.0
16	2681.0	803.0	3471.0	941.0	972.0
17	2642.0	815.0	3560.0	971.0	1019.0
18	2373.0	787.0	3134.0	918.0	967.0
19	2061.0	734.0	2824.0	758.0	814.0
20	1745.0	682.0	2489.0	688.0	753.0
21	1506.0	666.0	2169.0	663.0	701.0
22	1322.0	631.0	1959.0	643.0	702.0
23	1092.0	589.0	1655.0	574.0	633.0
24	959.0	540.0	1461.0	519.0	596.0

Table 5
MAE value comparison of the different models for $V_{A,Supply}$.

Hour	Naive	Smarter naive	PCA + LSTM	Only EESN	Proposed
1	409.0	472.0	1372.0	617.0	410.0
2	393.0	454.0	1320.0	367.0	383.0
3	391.0	453.0	1277.0	361.0	374.0
4	375.0	437.0	1227.0	484.0	342.0
5	379.0	421.0	1478.0	440.0	335.0
6	415.0	442.0	1474.0	372.0	374.0
7	570.0	494.0	1748.0	496.0	502.0
8	769.0	539.0	2058.0	669.0	609.0
9	923.0	576.0	2275.0	640.0	594.0
10	989.0	634.0	2347.0	728.0	686.0
11	990.0	676.0	2187.0	683.0	691.0
12	983.0	693.0	2099.0	869.0	695.0
13	950.0	694.0	2204.0	641.0	668.0
14	922.0	668.0	2148.0	723.0	656.0
15	932.0	657.0	2107.0	670.0	630.0
16	921.0	611.0	2248.0	648.0	619.0
17	928.0	567.0	2234.0	684.0	591.0
18	904.0	542.0	2284.0	570.0	580.0
19	826.0	489.0	2525.0	640.0	533.0
20	702.0	462.0	2159.0	709.0	511.0
21	670.0	473.0	2208.0	630.0	502.0
22	619.0	491.0	2225.0	565.0	479.0
23	516.0	478.0	1974.0	523.0	441.0
24	499.0	478.0	1778.0	507.0	419.0

seen for supply curves too. The $V_{A,supply}$ forecasts as shown in Table 5 follow a similar trend. This is attributed to the fact that even though good $V_{A,demand}$ and $V_{A,supply}$ forecasts should intuitively lead to better curve forecasts, this is not necessarily the case because a curve forecast does not depend only on the forecast of a single V point by itself. It depends on the good forecast of that V point with respect to other points on the curve. This is shown by comparing the results between Tables 3 and 5. For supply curves, the smarter naive model tends to have better $V_{A,supply}$ forecasts than the naive model. However, the naive model tends to have better curve forecasts than the smarter naive model.

6.3. The interesting segment

Figs. 14 and 15 show comparisons of the forecasted $\mathbf{C}_t$ for demand and supply with the ground truth curves for hour 11 of 15th July, 2019. These figures show that the forecasted curves are indeed monotonic in nature, but also show that supply is easier to forecast than demand, and that forecasting well the interesting segment is very hard indeed. These last two issues are in fact very difficult to take on within all models currently available, and improvements in techniques aimed at taking on them are ultimately the main motivation of the development of forecasting by means of the 'moving box and its internal reduced dynamics' framework presented here.

Finally, the monotonic autoencoder compresses demand or supply curve data more effectively than PCA. This suggests that the curves have characteristics which are complex. Unlike PCA, which relies on linear combinations to capture variance, autoencoders can learn patterns and dependencies that are not purely additive or proportional. This difference indicates that the demand and supply curves are likely influenced by non-linear interactions.

This improved compression also reveals intricate, non-redundant details in the curve that PCA may overlook. For example, demand and supply curves in electricity markets often exhibit subtle variations—small shifts due to behavioral changes, seasonal effects, or gradual trends — that require a non-linear model to be fully captured. The autoencoder's learned representation isolates these critical features providing a clear view of the core dynamics. This not only highlights the essential features within the curve but also suggests that the data's structure is more intricate than what linear techniques alone can reveal.

6.4. Supply vs. Demand

The proposed model treats the demand and supply curves as independent, to simplify the forecasting process and individually focus on the unique factors that influence each side of the market. While it is true that, in real markets, demand and supply curves are often interrelated, modeling them independently allows for capturing distinct dynamics that may affect each curve separately. For example, the demand curve is often driven by consumer behavior, weather patterns, and external conditions, while supply is influenced by factors such as production costs, resource availability, and regulatory policies. By forecasting them independently, these factors can be isolated and leveraged for forecasting. Furthermore, the interdependencies between the demand and supply curves are dynamic and not easy to quantify.

By treating demand and supply independently, the model provides flexibility to analyze each side individually and potentially identify unique trends or anomalies. This approach can also be adapted to incorporate interdependencies in a later work, if needed, by introducing additional terms or models that capture how changes in one curve may impact the other.

7. Discussion

The performance of the benchmark models reveals some insights into the models themselves. The PCA + Linear model, the PCA + LSTM model, and the X-model, perform significantly worse than the naive and smarter naive models. This might be attributed to the fact that these models rely on approximating the curve using a limited number of

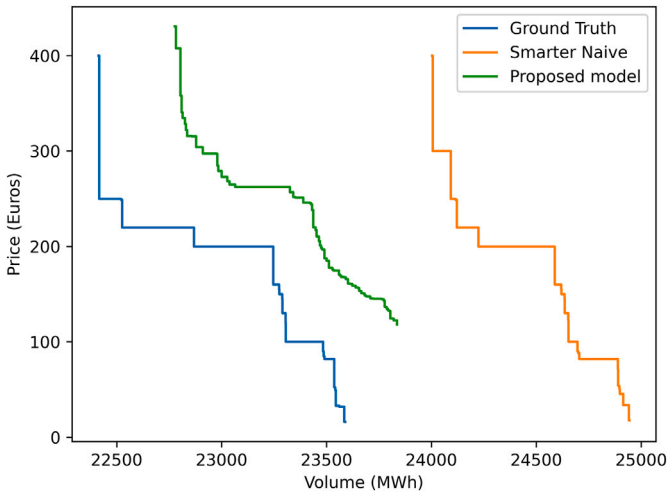


Fig. 14. Comparison of the forecast for $\hat{C}_{t,Demand}$ with the true curve for monotonicity.

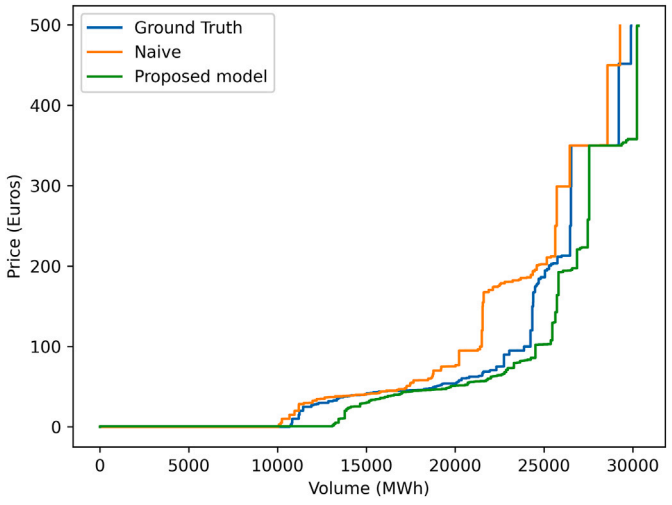


Fig. 15. Comparison of the forecast for $\hat{C}_{t,Supply}$ with the true curve for monotonicity.

points. Specifically, the PCA models use percentiles based on bid prices, and the X-model creates price classes based on the mean bid curve. As a result, the information in the curve cuts down to mostly around the interesting segment of the curve. The information lost from the rest of the curve is never retrieved. While the FAR model does consider the entire curve, it is hard to include exogenous information in the FAR scheme. In addition, in this scheme the forecasting model is linear. In comparison to these models, the proposed model keeps the curve information outside of the interesting segment, and the forecasting part is independent of the curve modeling part, hence any forecasting model can be used after the modeling of the curve.

In all, as shown in Section 6, the proposed model outperforms all the benchmark models significantly. As said, a major reason for this good performance comes certainly from the fact that the model is specifically designed for tracking the box where the interesting segment lives. However, it is also important to highlight a second possible reason. Unlike the models already present in the literature, which either use V as the support and predict P, or use P as the support and predict V, the proposed model has components for predicting both V and P, which can be another advantage in terms of efficiently using available information. In fact, in the first step the two V values, namely $V_{\mathbf{A}}$ and $V_{\mathbf{B}}$, are forecast, a step which essentially forecasts the interesting segment of the

individual curve. The second step forecasts the P values by forecasting $\hat{C}_t$. This feature may be critical for achieving the better performance of the proposed model.

In addition, there is still a third feature for which the proposed model could be especially interesting, which we however comment just in passing. In the proposed model the interesting segment is dealt with special care by means of the autoencoder, and the window in which it is tracked seems to work accurately. Hence, the crossing of the curves put out by our model, crossings which forecast the clearing volume and price, could be more precise than those put out by all other models.

8. Conclusions

The proposed framework aims at improving DAM supply and demand curve forecasting, by keeping a broad eye on the many possible details typical of the electricity market complex dynamics, and by strategically using all available DAM bid curve data and data knowledge. It also aims to maintain the results economically interpretable.

As discussed, the present study started from analyzing the aggregation procedure behind these monotonic curves, and taking into consideration many ideas already present in the related literature, which is still unfortunately rather limited in extension.

Within this framework, the curves were broken into three key parts that are structurally meaningful and easily interpretable. These were the two points $\mathbf{A}_t$ and $\mathbf{B}_t$, and the segment $\mathbf{C}_t$, which is the vector of points between them. From a computer vision point of view, this interesting segment can be viewed as living in a box, sliding back and forth in time. This suggested the possibility of first, separately forecasting the box position, then, taking care of the dynamics 'inside the box' by using a suitably developed monotonic autoencoder approach as a driving pattern for a data average, and finally optimally gluing the three forecasted pieces together by the use of a special non-differentiable measure called HC-MAE which required a non-gradient optimization technique. In passing, a variant of the ESN architecture, called the EESN, was used to forecast the interesting points and the embeddings inside the autoencoder. The forecasted $\mathbf{A}_t$, $\mathbf{B}_t$, and $\mathbf{C}_t$ were integrated using an integration model that also incorporates curve information at times $t-1$ and $t-7$ to produce a curve forecast aimed at lowering HC-MAE. In practice, ideas from the skip layer architecture were included to make the forecast more robust.

The evaluation of our framework's forecasting performance was carried out using as comparison benchmarks the most relevant literature models for both supply and demand. The framework turned out to be rather competitive against these benchmarks.

This comparison was interesting also because it could put into evidence which directions the past and current literature, including the presented framework, have not yet explored. It tried in any case to explore the overall difficult relationship between the typical curve data, coming from heterogeneous sets, and the different available ways for encoding them and for working with them.

It is important not to forget that, besides the IPEX, there are several other similar exchanges around the world, such as the EPEX, the JEPX (Japanese Electric Power Exchange), the PJM (Pennsylvania-New Jersey-Maryland Interconnection), the Australian markets, and new DAMs that are growing all over the world. All of these exchanges could be supposed to have similar market curves for their respective day-ahead markets, but that is not true. Hence, having a forecasting framework able to encompass all these subtly different behaviors could be of great benefit for the energy community, which is anyhow linked by common technology and consumption patterns, although declined in different local ways. Hence, it is hoped that the results presented in this paper are general enough to become useful to these other markets as well. Verifying that, is matter for further research.

Further research can also go in other directions. As mentioned in the preceding section, the behavior of crossing points can be an issue to explore as soon as possible. Moreover, all current models, including the presented framework, assume that the bids on different hours are uncoupled. This is assumed in order to make numerical experiments feasible, especially for functional or econometric models. However, because the core step of the presented framework boils down to forecast at first just two points of each curve, it could be easy to set up a forecasting system that forecasts these two points for all the 24 curves at once, thus including interactions of bids on different hours. Individual or coupled autoencoders could forecast at once the interesting segments as well. Finally, inclusion of exogenous dynamic information like weather and electric load forecasts (available at the moment of forecasting) is another item which should be explored as soon as possible.

CRediT authorship contribution statement

Nabangshu Sinha: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis. **Carlo Lucheroni:** Writing – review & editing, Supervision, Resources, Project administration, Methodology, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

C.L. acknowledges that, as to him, this work was funded by the European Union - NextGenerationEU under the Italian Ministry of University and Research (MUR) National Innovation Ecosystem grant ECS00000041 - VITALITY - CUP J13C22000430001.

Appendix A. How to collect the IPEX bid data

The raw IPEX data used in the paper consist of daily bid information, and can be freely downloaded from the GME website [20]. In the specific, by following the URL found in Ref. [48], one is led to a page where the options to be selected are "MGP" (Mercato del Giorno Prima), then "Downloads", and then "Public data". A date must then be entered in order to download the data which are needed for forming the price/volume curves of the DAM ('Mercato del Giorno Prima', MGP) for that day. The file is downloaded as "[yyyymmdd]MGPOffertePubbliche.zip" ('MGP Public Offers'). Once unzipped, one gets an XML file that needs to be opened with an Excel compatible program. In the XML file, once filtered for the NORD market zone, the bids come in two categories, BID (demand) and OFF (supply). The prices are under the column "ENERGY_PRICE_NO" and the quantities are under "AWARDED_QUANTITY_NO".

Noticeably, on this page one can also find a lot of other interesting information, like data related to the other markets managed by the IPEX besides the MGP, like seven Intra-Day Markets (MI) for intra-day adjustments, the Energy Imports and Exports Management Platform (MPEG) for cross-border transactions, the Renewable Energy Withdrawal Market (MRR) for renewable energy integration, and the Dispatching Services Market (MSD) for real-time balancing and grid stability.

Data availability

Data will be made available upon request.

References

[1] Weron R. Electricity price forecasting: a review of the state-of-the-art with a look into the future. Int J Forecast 2014;30(4):1030–81.
[2] Ziel F, Weron R. Day-ahead electricity price forecasting with high-dimensional structures: univariate vs. multivariate modeling frameworks. Energy Econ 2018;70:396–420.
[3] Percebois J, Pommeret S. Reform of the European electricity market: should we prefer a price based on a weighted average of marginal costs with cross-subsidies? Electr J 2024;37(1):107364.

[4] Schwarz P. Energy economics. Routledge; 2022.
[5] Kazemi M, Zareipour H, Ehsan M, Rosehart WD. A robust linear approach for offering strategy of a hybrid electric energy company. IEEE Trans Power Syst 2016;32(3):1949–59.
[6] Chen R, Paschalidis IC, Caramanis MC, Andrianesis P. Learning from past bids to participate strategically in day-ahead electricity markets. IEEE Trans Smart Grid 2019;10(5):5794–806.
[7] Guo H, Chen Q, Zheng K, Xia Q, Kang C. Forecast aggregated supply curves in power markets based on lstm model. IEEE Trans Power Syst 2021;36(6):5767–79.
[8] Kazmi H, Tao Z. How good are TSO load and renewable generation forecasts: learning curves, challenges, and the road ahead. Appl Energy 2022;323:119565.
[9] Pinhão M, Fonseca M, Covas R. Electricity spot price forecast by modelling supply and demand curve. Mathematics 2022;10(12):2012.
[10] Yıldırım S, Khalafi M, Güzel T, Satık H, Yılmaz M. Supply curves in electricity markets: a framework for dynamic modeling and Monte Carlo forecasting. IEEE Trans Power Syst 2022;38:3056–69.
[11] Shah I, Lisi F. Forecasting of electricity price through a functional prediction of sale and purchase curves. J Forecast 2020;39(2):242–59.
[12] Ciarreta A, Martinez B, Nasirov S. Forecasting electricity prices using bid data. Int J Forecast 2023;39(3):1253–71.
[13] Pelagatti M. Supply function prediction in electricity auctions. In: Complex models and computational methods in statistics. Springer; 2012. pp. 203–13.
[14] Tang Q, Guo H, Zheng K, Chen Q. Forecasting individual bids in real electricity markets through machine learning framework. Appl Energy 2024;363:123053.
[15] Ziel F, Steinert R. Electricity price forecasting using sale and purchase curves: the X-model. Energy Econ 2016;59:435–54.
[16] Ziel F, Steinert R. Probabilistic mid-and long-term electricity price forecasting. Renew Sustain Energy Rev 2018;94:251–66.
[17] Haben S, Caudron J, Verma J. Probabilistic day-ahead wholesale price forecast: a case study in Great Britain. Forecasting 2021;3(3):596–632.
[18] Ghelasi P, Florian Z. Hierarchical forecasting for aggregated curves with an application to day-ahead electricity price auctions. Int J Forecast 2024;40(2):581–96.
[19] Mestre G, Portela J, San Roque AM, Alonso E. Forecasting hourly supply curves in the Italian day-ahead electricity market with a double-seasonal SARMAHX model. Int J Electr Power Energy Syst 2020;121:106083.
[20] GME. Italian 'gestore dei mercati energetici' website. Available from: https://www.mercatoelettrico.org/en-us/Home. [Accessed January 2024].
[21] Li Z, Alonso AM, Elías A, Morales JM. Clustering and forecasting of day-ahead electricity supply curves using a market-based distance. Int J Electr Power Energy Syst 2024;158:109977.
[22] N. Sinha, C. Lucheroni, BME model: Forecasting electricity supply and demand curves using disentangled prices and volumes 2025 (submitted).
[23] Cover T, Hart P. Nearest neighbor pattern classification. IEEE Trans Inf Theory 1967;13(1):21–27.
[24] Storn R, Price K. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. J Global Optim 1997;11(4):341–59.
[25] Kramer MA. Nonlinear principal component analysis using autoassociative neural networks. AlChE J 1991;37(2):233–43.
[26] Wang Y, Yao H, Zhao S. Auto-encoder based dimensionality reduction. Neurocomputing 2016;184:232–42.
[27] Sakurada M, Yairi T. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In: Proceedings of the MLSDA 2014 2nd workshop on machine learning for sensory data analysis; 2014. p. 4–11.
[28] Zabalza J, Ren J, Zheng J, Zhao H, Qing C, Yang Z, et al. Novel segmented stacked autoencoder for effective dimensionality reduction and feature extraction in hyperspectral imaging. Neurocomputing 2016;185:1–10.
[29] Pfeiffer J, Broscheit S, Gemulla R, Göschl M. A neural autoencoder approach for document ranking and query refinement in pharmacogenomic information retrieval. In: Proceedings of the BioNLP 2018 workshop; 2018. p. 87–97.
[30] Jaeger H. Echo state network. Scholarpedia 2007;2(9):2330.
[31] Vlachas PR, Pathak J, Hunt BR, Sapsis TP, Girvan M, Ott E, et al. Backpropagation algorithms and reservoir computing in recurrent neural networks for the forecasting of complex spatiotemporal dynamics. Neural Netw 2020;126:191–217.
[32] Sinha N, Lucheroni C. Double-reservoir deep echo state network architecture for short-term electricity demand forecasting. In: 2022 18th international conference on the European energy market (EEM); IEEE; 2022. p. 1–6.
[33] Deihimi A, Showkati H. Application of echo state networks in short-term electric load forecasting. Energy 2012;39(1):327–40.
[34] Zheng K, Qian B, Li S, Xiao Y, Zhuang W, Ma Q. Long-short term echo state network for time series prediction. IEEE Access 2020;8:91961–74.
[35] Hu H, Wang L, Lv S-X. Forecasting energy consumption and wind power generation using deep echo state network. Renew Energy 2020;154:598–613.
[36] Kitouni O, Nolte N, Williams M. Expressive monotonic neural networks. arXiv:2307.07512 [Preprint]. 2023
[37] Wehenkel A, Louppe G. Unconstrained monotonic neural networks. arXiv:1908.05164 [Preprint]. 2023.
[38] Runje D, Shankaranarayana SM. Constrained monotonic neural networks. In: Proceedings of the 40th international conference on machine learning 2023; Vol. 202. PMLR; 2023. p. 29338–53.
[39] Karmakar P, Teng SW, Lu G. Transformer with enhanced encoder and monotonic decoder for automatic speech recognition. In: International conference on digital image computing: techniques and applications (DICTA), Sydney, Australia, 2022; 2022. p. 1–5.
[40] Dong C, Zhao J. An augmented autoencoder with multi-head attention for tool wear prediction in smart manufacturing. IEEE Access 2024;12:79128–37. https://doi.org/10.1109/ACCESS.2024.3406568
[41] Maarten M, Van Baelen Q, Ooijevaar T, Karsmakers P. Constraint guided autoencoders for joint optimization of condition indicator estimation and anomaly detection in machine condition monitoring. arXiv:2409.11807 [Preprint]. 2024.
[42] Fong Y, Xu J. Forward stepwise deep autoencoder-based monotone nonlinear dimensionality reduction methods. J Comput Graph Stat 2021;30(3):519–29.
[43] Lang S. Introduction to linear algebra. Springer Science & Business Media; 2012.
[44] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition. In: Conference on computer vision and pattern recognition, https://doi.org/10.1109/CVPR.2016.90, and also in arXiv:1512.03385 (2016).
[45] Grazzini F, Craig GC, Keil C, Antolini G, Pavan V. Extreme precipitation events over Northern Italy. Part I: A systematic classification with machine-learning techniques. Quart J Roy Meteor Soc 2020;146(726):69–85.
[46] Cavaleri L, Bajo M, Barbariol F, Bastianini M, Benetazzo A, Bertotti L, et al. The October 29, 2018 storm in Northern Italy–an exceptional event and its modeling. Prog Oceanogr 2019;178:102178.
[47] Serge DJG. Far: modelization for functional AutoRegressive processes, R package version 0.6-6; 2022. Available from: https://CRAN.R-project.org/package=far.
[48] GME/download. Download data from gme. Available from: https://www.mercatoelettrico.org/en-us/Home/Results/Electricity/MGP/Download/Download?valore=OffertePubbliche. [Accessed January 2024].