

A Class-Driven Approach Based on Long Short-Term Memory Networks for Electricity Price Scenario Generation and Reduction

Bart Stappers, Nikolaos G. Paterakis , *Member, IEEE*, Koen Kok, *Senior Member, IEEE*, and Madeleine Gibescu, *Member, IEEE*

Abstract—Uncertainty characterization is an essential component of decision-making problems in electricity markets. In this work, a class-driven approach is proposed to describe stochasticity. The methodology consists of a three-step process that includes a class allocation component, a generative element based on a long short-term memory neural network and an automated reduction method with a variance-based continuation criterion. The system is employed and evaluated on Dutch imbalance market prices. Test results are presented, expressing the proficiency of the approach, both in generating realistic scenario sets that reflect the erratic dynamics in the data and adequately reducing generated sets without the need to explicitly and manually predetermine the cardinality of the reduced set.

Index Terms—Deep learning, imbalance prices, long short-term memory (LSTM), machine learning, recurrent neural network (RNN), scenario generation, scenario reduction.

I. INTRODUCTION

UNCERTAINTY constitutes an intricate and pervasive phenomenon in electricity markets. By no means would it be an overstatement to submit that both operators and market participants alike, ranging from utilities to retailers, are affected to lesser or greater extent by its presence. The existing capriciousness is exacerbated even further by the progressively increasing concentration of renewable energy sources, particularly wind and photovoltaic (PV) generation, in contemporary power systems [1]–[3]. To cope with the undesirable consequences

of uncertainty, participants may rely on deterministic operational planning procedures based on point forecasts. However, such prediction methods, merely representing a single summary statistic, are often highly inaccurate and contribute limited support for informed decision-making under uncertainty [4]. Therefore, rather than solely relying on the estimated mean of some stochastic process of interest to dictate market engagement, stochastic programming techniques have been adopted to explicitly model uncertainty and guide decision-makers in establishing optimal courses of action [5]–[8].

Such methods require elaborate contextual information on the stochastic processes to which the operator or market participant is exposed, i.e., stochastic optimization models need the latent uncertainty to be quantified. Commonly, the uncertainty underlying a stochastic process is characterized using a set of scenarios, comprising plausible realizations of the process throughout the decision-making horizon [9]. Generating a scenario set that appropriately captures the erraticity of a stochastic process poses a challenging problem.

A. Related Work

A variety of approaches have been proposed in the literature. In [10] sample paths generated based on parametric or non-parametric models were used to create scenario trees of predefined structure by means of cluster analysis or importance sampling. A comparable simulation-based methodology was presented in [11]. However, this approach differs from [10] in the sense that it provides a detailed algorithm for randomized clustering of simulations.

An optimization-based technique was proposed in [12], where a continuous-state stochastic process is approximated by a discrete scenario set that is optimal in terms of the approximation error. An optimal discretization is found by minimizing the approximation error, which, in casu, is defined as the Wasserstein distance between the optimized objective function values of the underlying and approximate problems.

Another research strand has explored the potential of exploiting statistical moments for the generation of representative scenarios. In [13], a moment-matching technique was proposed where the core idea is to find a limited set of generated scenarios that satisfies some pre-specified statistical properties. Extending the work in [13], a particularly efficient algorithm yielding a

Manuscript received June 27, 2019; revised October 15, 2019 and December 9, 2019; accepted December 29, 2019. Date of publication January 13, 2020; date of current version June 22, 2020. This work was supported by Scholt Energy B.V., The Netherlands. Paper no. TPWRS-00919-2019. (Corresponding author: Nikolaos Paterakis.)

B. Stappers is with the Department of Electrical Engineering, Eindhoven University of Technology, 5600 MB Eindhoven, The Netherlands, is also with the Scholt Energy B.V., 5555 XA Valkenswaard, The Netherlands (e-mail: b.stappers@tue.nl).

N. G. Paterakis is with the Department of Electrical Engineering, Eindhoven University of Technology, 5600 MB Eindhoven, The Netherlands (e-mail: n.paterakis@tue.nl).

K. Kok is with the Department of Electrical Engineering, Eindhoven University of Technology, 5600 MB Eindhoven, The Netherlands, and also with the The Netherlands Organisation for applied scientific research TNO, 9727 DW Groningen, The Netherlands (e-mail: k.kok@tue.nl).

M. Gibescu is with the Copernicus Institute of Sustainable Development, Utrecht University, 3584CB Utrecht, The Netherlands (e-mail: m.gibescu@uu.nl).

Color versions of one or more of the figures in this article are available online at <https://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TPWRS.2020.2965922

discrete joint distribution consistent with pre-defined moments and correlations was presented in [14].

Hybrid approaches have also gained attention. Relying on moment-matching methods, [15] first used simulation techniques to generate nodes of the scenario tree. Subsequently, these outcomes serve as input parameters for an optimization model, the objective of which is to determine the probabilities of the constituent scenarios such that they match some specific targets. A similar implementation was proposed in [11].

More recently, scenario generation approaches based on machine learning algorithms have become more prevalent in the literature. Using numerical weather predictions as input and a particle swarm optimization (PSO) algorithm for tuning purposes, [16] implemented radial basis function neural networks (RBFNNs) to generate wind power scenarios. A scenario generation method based on a feed-forward neural network (FFNN) was proposed in [17], where the authors employed a genetic algorithm rather than one of the more conventional gradient-based algorithms that are typically used for training purposes. A related neural network architecture was used in [4]. The authors proposed an iterative process based on an assimilation of FFNN outputs with randomly generated Gaussian white noise to create stochastic scenarios. The applicability of generative adversarial networks (GANs), a fairly novel branch in deep learning, in generating wind and photovoltaic power scenarios was investigated in [18]. A generative and a discriminative deep neural network are linked as adversaries in a minimax game. The former is tasked with generating fake samples of some process while the latter aims at discriminating between fake samples and real historical observations. Theoretically, GANs should produce scenarios that are indistinguishable from true historical observations after attaining the Nash equilibrium.

Several studies pay particular attention to scenario generation for electricity prices. In [19], the authors proposed an ARIMA-based model for generating regulation power price scenario trees and applied it in a case study related to the Nordic power market. In the context of optimal participation of electric vehicles in the electricity markets, Jensen *et al.* [20] also studied scenario generation for regulating power prices. They considered three methods to characterize the underlying uncertainty: time-series analysis of historical data, property matching, and copula matching. A two-stage approach combining an ARMA model and moment-matching for the generation of day-ahead price scenarios for the Midwest ISO (US) was presented in [21].

To the best of the authors' knowledge, no studies have been conducted on imbalance price scenario generation for power systems where price formation in the day-ahead market and the imbalance market are not directly linked, such as the Dutch and Belgian markets.

Accurately representing the uncertainty of a stochastic process with a discrete scenario set usually involves the generation of a very large number of potential realizations. Since computational burden increases with the number of scenarios in consideration, associated optimization problems rapidly become intractable as the cardinality of scenario sets increases. As a result, reduction techniques are needed to curtail the number of

scenarios considered, while minimizing the inevitable dilution of stochastic information contained in the original set.

Similar to the scenario generation case, several research domains can be distinguished in the literature related to scenario reduction. One strand attempts to reduce a set of scenarios such that certain statistical moments of the reduced set match those of the initial set [13], [14]. This approach has the benefit that it preserves desirable properties of the original set. However, as was shown in [22], moment-matching methods may lead to peculiar results.

Dupačová *et al.* [23] presented a methodology for tackling the optimal scenario reduction problem by minimizing the Kantorovich distance between the initial scenario set and the reduced scenario set. The Kantorovich distance is the optimal value of a linear problem known as the *Monge-Kantorovich mass transportation problem*. The authors showed that, under mild assumptions [9], the solution to this problem can be computed explicitly. Using this result, two heuristic algorithms to determine the optimal reduced set of predefined cardinality were derived. The first, a backward reduction method, iteratively removes scenarios from the original set until it reaches the specified cardinality and the remaining scenarios comprise the final approximation. Conversely, the forward selection algorithm recursively adds scenarios from the initial set to the reduced set until the latter totals a desired number of constituent members. Heitsch and Römisch [24] developed the fast forward selection (FFS) and simultaneous backward reduction (SBR) algorithms to improve the respective algorithms in [23]. In particular, FFS proved to have higher computational performance, both in terms of accuracy and running time, than its predecessor. SBR results in higher accuracy compared to vanilla backward reduction, albeit at the expense of running time. In [9], the authors presented an alternative reduction procedure for two-stage stochastic problems that is based on forward selection. The proposed methodology yielded the same level of stochastic information as alternative methods while significantly reducing the required number of scenarios. However, this increase in efficiency comes with comparatively higher computational burden. Broadly speaking, scenario reduction techniques that rely on the notion of forward selection appear to imply the lowest computational effort [25].

A potential drawback of the described reduction techniques is the fact that the cardinality of the reduced scenario set is a parameter to be pre-specified by the user. Currently, the only study that addresses this issue is [26], in which a reduction algorithm based on sub-modular function optimization is proposed to endogenously determine the number of scenarios in the reduced set.

B. Motivation and Contributions

Given the context described above, this paper is motivated by several research gaps. First, it is to be noted that a large number of stochastic processes in electricity markets is considered continuous in nature, e.g., demand, price formation and wind power generation. As a result, the literature regarding scenario generation deals with finding appropriate discrete approximation sets for such continuous-state stochastic processes. However,

in many decision-making situations related to power markets, interest is not so much in exact underlying realizations of the constituent random variable as it is in value ranges it might be in. Under such circumstances, where one cares about the class or bin to which realizations of the random variable belong rather than the exact values they might take on, a different approach is warranted. Naturally, this opens the door to classification-based techniques for scenario generation. Remarkably, no studies have been performed adopting this perspective.

Second, and equally interesting, is the emergence of machine learning-inspired techniques in the scenario generation literature. Their recent influx signifies researchers' recognition of the virtues of data-driven methods that have greater ability for handling non-linear dependencies, without the need for explicit modeling and stringent assumptions that may or may not hold true. A considerable portion of studies in this area employ feed-forward neural networks, which appear to perform satisfactorily. Little or no attention, however, has been paid to neural networks that incorporate recurrent feedback elements. Such iterative elements essentially serve as artificial memory to the network, which increases its ability to capture (long-term) temporal dependencies. Therefore, this particular area of machine learning seems promising for scenario generation related to the stochastic processes that are typically studied in electricity markets research.

A third impetus that motivated this work is related to scenario reduction. As briefly discussed in Section I-A, the majority of existing techniques require parameters related to the termination of the reduction algorithm to be predefined. In the case of ubiquitously used forward selection-based procedures, this is typically the desired cardinality of the reduced set or the probability distance between the initial and reduced sets. For situations characterized by frequent and time-constrained decision-making, both instances have possible impeding effects. In particular, relying on some measure of probability distance presumes prior knowledge about the underlying problem. It requires, at the very least, a rudimentary notion of plausible and acceptable ranges for the computed distance to be bounded by. On the other hand, setting a cardinality parameter may offer a more intuitive alternative. However, doing so comes at the cost of not having an indication as to the informational value contained in the resulting reduced set. Setting the parameter too low may lead to an approximate set that is too distant from the original one, whereas setting the parameter too high unnecessarily increases running time.

In the light of the aforementioned observations, the contribution of this paper is threefold:

- 1) *Class-driven scenario generation*: This work proposes a method in which scenario generation related to a stochastic process is treated as a classification problem rather than a regression problem.
- 2) *Long short-term memory scenario generation*: This study takes a scenario generation approach that not only considers spatial depth as typical deep FFNNs do, but also takes temporal dynamics into account by relying on neural network memory components that are able to store information over long periods of time.

- 3) *Automated scenario reduction*: This paper proposes a scenario set reduction algorithm that strikes a balance between the accuracy offered by forward selection-based methods and running time constraints that are present in many operational settings. Relying on a variance-based continuation criterion, it requires minimal user input.

The remainder of the paper is organized as follows: Section II describes in detail the workings of recurrent neural networks, which are fundamental to this work. Section III develops the proposed implementation for uncertainty characterization. Section IV provides details on the experiments that were conducted to test the implementation and discusses results. Finally, Section V draws relevant conclusions.

II. RECURRENT NEURAL NETWORKS

This section provides relevant details on recurrent neural networks, the technology that constitutes the core of the proposed methodology presented in Section III.

A. Elementary Recurrent Neural Networks

Recurrent neural networks (RNNs) have been known as highly effective systems for a variety of sequence learning problems. A RNN can be considered a generalization of feed-forward neural networks to sequential data, e.g., text data or time series data [27].

In its traditional form, a RNN maps a sequence of inputs \mathbf{x} into a fixed-length output sequence \mathbf{y} through the computation of a *hidden state* sequence \mathbf{h} . To effectively perform this encoding, the core of a typical recurrent network architecture consists of multiple (layers of) *units* or *blocks*.

Consider a RNN layer that comprises N units and has input dimensionality M . Also, let $\mathbf{W} \in \mathbb{R}^{N \times M}$ be the input weights, $\mathbf{U} \in \mathbb{R}^{N \times N}$ the recurrent weights, $\mathbf{b} \in \mathbb{R}^N$ the bias vector, and let \mathbf{x}_t be the input vector at $t \in \{1, 2, \dots, T\}$. Then, the hidden state \mathbf{h} at time step t is defined as:

$$\tilde{\mathbf{h}}_t = \mathbf{W}\mathbf{x}_t + \mathbf{U}\mathbf{h}_{t-1} + \mathbf{b} \quad (1)$$

$$\mathbf{h}_t = f(\tilde{\mathbf{h}}_t) \quad (2)$$

where f is a nonlinear activation function. In most cases, the hyperbolic tangent, which compresses $\tilde{\mathbf{h}}_t$ to the range $[-1, 1]$, is used. The hidden state acts as a memory, allowing the block to capture relevant information from previous time steps of the input vector \mathbf{x}_t . Therefore, by design, RNNs should be more capable of modeling inter-temporal dynamics than feed-forward networks are.

Unfortunately, they are notoriously difficult to train properly, mainly due to the *vanishing* and *exploding gradient* problems that may arise when conventional gradient-based learning algorithms such as *backpropagation through time* (BPTT) are employed [28] (for a detailed analysis of these issues, see [29]).

B. Long Short-Term Memory Networks

To alleviate such training-related complications, Hochreiter and Schmidhuber [30] introduced the *long short-term memory* (LSTM) unit or block. The inner workings of such units are

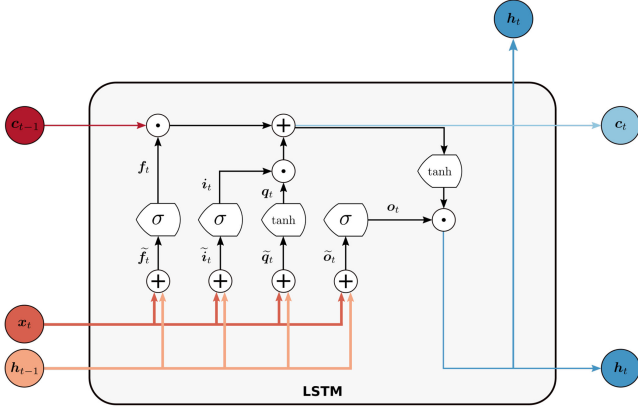


Fig. 1. Schematic of an LSTM unit. Using the input vector \mathbf{x}_t and hidden state of the previous unit \mathbf{h}_{t-1} , information is removed from (\mathbf{f}_t) or added to (\mathbf{i}_t and \mathbf{q}_t) the cell state \mathbf{c}_{t-1} to yield the updated memory \mathbf{c}_t . The cell state is combined with information from the output gate (\mathbf{o}_t) to arrive at the new hidden state, which flows to the next block (bottom right) or layer (top).

illustrated in Fig. 1. The principal idea behind an LSTM block is a modifiable memory cell that can store information over long periods of time. The state of this cell can be altered by various *gating units*, allowing the network to learn what information is relevant and “unlearn” information that has become obsolete with respect to the purpose at hand. The main element of each unit is the internal cell state \mathbf{c} . The internal state \mathbf{c}_{t-1} can be altered based on information contained in the input vector \mathbf{x}_t and the hidden state of a previous block \mathbf{h}_{t-1} . Potentially relevant information from these vectors is captured in \mathbf{q}_t and, subject to filtering by the *input gate* \mathbf{i}_t , added to the cell state to yield \mathbf{c}_t . The updated cell state is then passed through a hyperbolic tangent layer and weighted by the *output gate* \mathbf{o}_t to yield \mathbf{h}_t . The updated hidden state is subsequently passed onward, both temporally and a hierarchically, depending on the topology of the neural network. The initial concept of LSTM blocks was further advanced when [31] proposed to add a *forget gate* \mathbf{f}_t . By enabling the unit to reset its own cell state at appropriate moments, this gate was shown to aid in learning of continual tasks.

Naturally, the features described above facilitate implementation of LSTM-based architectures in a wide array of problems involving sequential data.

III. PROPOSED METHODOLOGY

This section provides details on the methodology, comprising a system that is both iterative and sequential, in a step-wise fashion. The proposed workflow is illustrated in Fig. 2.

A. Step 1: Class Allocation

First, historical observations of a stochastic process of interest are aggregated into classes following a procedure similar as in [32]. Given a sequence of N^{steps} realizations $\mathbf{x}_t^{\text{real}}$ and vector \mathbf{v}^{cut} containing N^{cut} cut-off points, by implementing Algorithm 1, each observed value in $\mathbf{x}_t^{\text{real}}$ (line 2) is allocated to a class (line 3) and stored in vector \mathbf{v}_t (line 4), with support points \mathbf{v}^{cut} .

Algorithm 1: Determine Appropriate Class Allocation for each Observed Value in the Sequence Under Consideration.

Input: $\mathbf{x}_t^{\text{real}}, \mathbf{v}^{\text{cut}}$

- 1: Initialize \mathbf{v}_t with zeros everywhere on $\mathbb{N}_{\geq 0}^{|\mathbf{x}_t^{\text{real}}|}$
 - 2: **for** $i = 1, 2, \dots, |\mathbf{x}_t^{\text{real}}|$ **do**
 - 3: $j^* \leftarrow \arg \max_j v_j^{\text{cut}}, \text{ subject to } v_j^{\text{cut}} \leq x_{t,i}^{\text{real}}$
 - 4: $v_{t,i} \leftarrow j^*$
 - 5: **end for**
- Output:** \mathbf{v}_t

B. Step 2: Iterative Generation

In the second (generative) step, three sequential substeps can be distinguished: data processing, parameter prediction and distribution sampling.

1) *Data Processing:* The elements in \mathbf{v}_t are transformed into a binary categorical representation by applying a one-hot encoding, yielding $\mathbf{x}_t^{\text{hot}} \in \mathbb{N}_{\{0,1\}}^{N^{\text{steps}} \times N^{\text{cut}}}$. It is assumed that the dimensionality of the discrete structure does not raise concerns with regard to over-fitting. For high-dimensional structures, one-hot encoding may be substituted with an alternative procedure that results in a more dense representation, e.g., [33]. After appropriate scaling, exogenous features and time-indexing inputs may be incorporated, if relevant. Let N^{exo} and N^{time} be the number of exogenous and time-indexing features, respectively. To ensure consistency with $\mathbf{x}_t^{\text{hot}}$ the feature values are presented in rank-2 tensors: $\mathbf{x}_t^{\text{exo}} \in \mathbb{R}^{N^{\text{steps}} \times N^{\text{exo}}}$ and $\mathbf{x}_t^{\text{time}} \in \mathbb{R}^{N^{\text{steps}} \times N^{\text{time}}}$. The results are concatenated to produce the array $\mathbf{x}_t \in \mathbb{R}^{N^{\text{steps}} \times N^{\text{feat}}}$, where $N^{\text{feat}} = N^{\text{exo}} + N^{\text{time}}$.

2) *Parameter Prediction:* A deep recurrent neural network architecture is employed to provide a one-step ahead parameter prediction of a distribution over N^{cut} classes. An overview of a possible network hierarchy is shown in Fig. 3. The network takes as input a 3-D tensor, given by reshaping \mathbf{x}_t into shape $1 \times N^{\text{steps}} \times N^{\text{feat}}$. The input is processed sequentially by a long short-term memory network, with weights reflecting learned temporal dependencies and dynamics in history. Note that, by design, recurrent neural networks are deep in time, as the hidden state at each time step contains information about previous hidden states. Spatial depth, on the other hand, can be attained by stacking multiple recurrent hidden layers on top of each other [34]. Each layer receives the learned input representation of its predecessor. Therefore, as information flows upwards in the hierarchy, increasing levels of representational abstraction are achieved. The recurrent layers are followed by dense layers, both to attain further depth and to ensure consistency between the dimensionality of the network output and the target. The final layer is an implementation of the softmax function:

$$\varphi(z_k) = \frac{\exp(z_k)}{\sum_{k' \in \mathbf{v}^{\text{cut}}} \exp(z_{k'})}, \quad (3)$$

where $\mathbf{z} \in \mathbb{R}^{N^{\text{cut}}}$ is the output vector of the dense layer preceding the softmax layer. It normalizes the entries in \mathbf{z} , such that $\sum_{k=1}^{N^{\text{cut}}} \varphi(z_k) = 1$. Consequently, the output of the neural

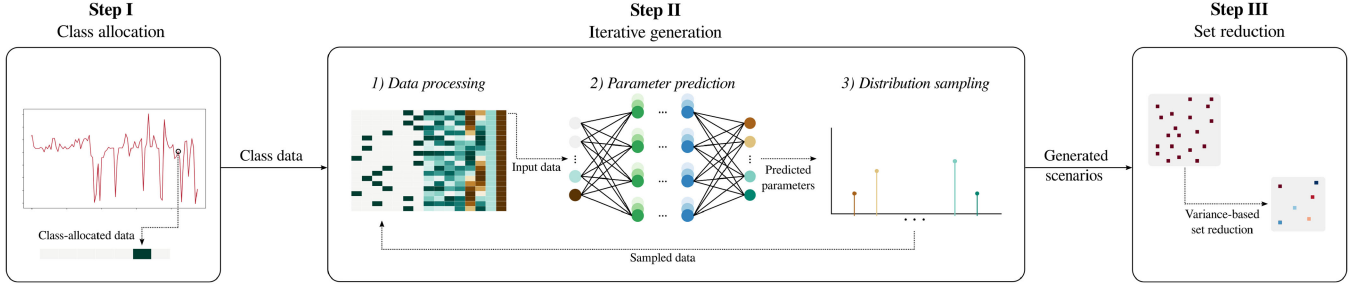


Fig. 2. Schematic of the proposed approach. Three main steps can be discerned: class allocation, iterative scenario generation and scenario set reduction. The generative step, in turn, consists of three iterative components, i.e., data processing, parameter prediction and distribution sampling.

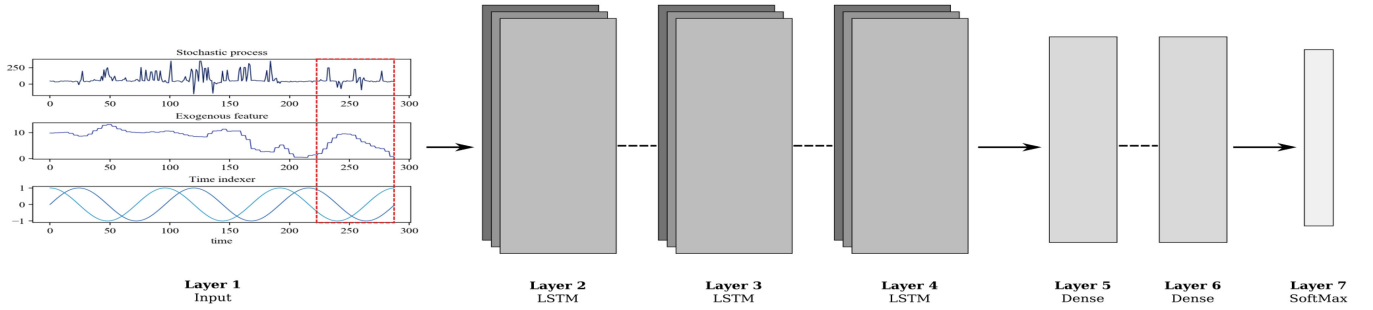


Fig. 3. Full neural network configuration. Note that this is a specific architectural instance, similar to the one used in Section IV. Network input consists of historical realizations of the stochastic process, but may also include exogenous and time-indexing variables. The input is presented to the LSTM as a rank-3 tensor. The output layer comprises a dense layer with softmax activation function, resulting in a conditional distribution over N^{cut} different classes.

network can be interpreted as predicted parameters that define the probability distribution of the target y_t .

3) *Distribution Sampling*: Leveraging the softmax transformation in (3), the LSTM output is used to parametrize a multinomial distribution \mathcal{P} with:

$$p(y_t = k \mid \varphi(z)) = \varphi(z_k), \forall k \in \mathbf{v}^{cut}. \quad (4)$$

A one-step ahead prediction \hat{y}_t is then obtained by sampling from probability distribution \mathcal{P} . At time $t + 1$, the prediction is incorporated in the class input sequence, which results in $\mathbf{v}_{t+1} = \{v_{t,i-N_{steps}+2}, \dots, v_{t,i}, \hat{y}_t\}$. The three substeps are reiterated $(T - 1)$ times to obtain a scenario of the form $\{\hat{y}_t, \hat{y}_{t+1}, \dots, \hat{y}_T\}$, where T is a pre-specified time horizon. A scenario set Z is then generated by repeating the entire procedure N_Z times, with N_Z denoting the desired cardinality.

C. Step 3: Scenario Set Reduction

In the final step, the generated scenario set is reduced to a subset by means of an iterative, greedy algorithm that is premised on the notion of forward selection. Building on earlier results (e.g., [23], [24], [35]), a novel and automated set reduction procedure with a variance-based continuation criterion is proposed. Let $\Omega = \{1, 2, \dots, N_Z\}$ be an index set to Z , such that $Z = \bigcup_{\omega \in \Omega} Z_\omega$. Along with Ω and the probability distribution π of Z , Algorithm 2 takes as input a threshold θ and a cost matrix $\Gamma \in \mathbb{R}^{N_Z \times N_Z}$. The elements of Γ are given by:

$$\Gamma_{\omega, \omega'} = \|Z_\omega - Z_{\omega'}\|, \forall \omega, \omega' \in \Omega, \quad (5)$$

where $\|\cdot\|$ denotes the Euclidean norm. Note that it is implicitly assumed that the stochastic process of interest can be appropriately characterized by the scenario set Z .

After initializing the counter (line 1) and the cost matrix (line 2), the procedure commences by determining the scenario ω that minimizes the distance between itself and all other scenarios in Ω (line 3). This scenario is used to split the initial set into a reduced set Ω_s and a set comprising the remaining scenarios Ω_c (lines 4 and 5, respectively). The cost matrix is adjusted (line 9) and a non-preserved scenario is selected (line 10), before Ω_s and Ω_c are updated accordingly (lines 11 and 12). Note that, at each iteration, the scenario $\omega \in \Omega_c$ is selected that minimizes the probability distance between Ω and Ω_s if it would be included in the preserved scenario set. The variance of the scenarios in the reduced set, averaged over the temporal dimension with time horizon H , is calculated for the previous and current iteration (lines 13 and 14). When defined (lines 17 and 20), the relative change of $\bar{\sigma}^2$ between iterations is computed (lines 18 and 21) and stored (lines 19 and 22). For $i > n$, the moving average of $\bar{\sigma}^2$ over the previous n iterations, λ , is calculated and updated (line 23). The iterative process is terminated when the continuation criterion (line 7) is rendered invalid, i.e., when λ undershoots the predefined threshold θ .

As the scenario reduction method proposed in this work relies heavily on the (fast) forward selection procedure, it also inherits its properties. As was noted in [9], the forward selection algorithm does not guarantee the reduced set to be closest in the Kantorovich distance to the generated scenario set over all

Algorithm 2: Automated Scenario Set Reduction Using Variance-Based Continuation Criterion.

Input: Γ , π , Ω , θ , n

- 1: $i \leftarrow 0$
- 2: $\Gamma^{[0]} \leftarrow \Gamma$
- 3: $\omega^{[0]} \leftarrow \arg \min_{\omega \in \Omega} \sum_{\omega' \neq \omega}^{|\Omega|} \pi_{\omega'} \Gamma_{\omega, \omega'}^{[0]}$
- 4: $\Omega_s^{[0]} \leftarrow \omega^{[0]}$
- 5: $\Omega_c^{[0]} \leftarrow \Omega \setminus \omega^{[0]}$
- 6: $\lambda \leftarrow \theta + 1$
- 7: **while** $\theta \leq \lambda$ **do**
- 8: $i \leftarrow i + 1$
- 9: $\Gamma_{\omega, \omega'}^{[i]} \leftarrow \min\{\Gamma_{\omega, \omega'}^{[i-1]}, \Gamma_{\omega, \omega'}^{[i-1]}\}, \forall \omega, \omega' \in \Omega_c^{[i-1]}$
- 10: $\omega^{[i]} \leftarrow \arg \min_{\omega' \in \Omega_c^{[i-1]}} \sum_{\omega'' \in \Omega_c^{[i-1]} \setminus \{\omega'\}} \pi_{\omega''} \Gamma_{\omega', \omega''}^{[i]}$
- 11: $\Omega_s^{[i]} \leftarrow \Omega_s^{[i-1]} \cup \omega^{[i]}$
- 12: $\Omega_c^{[i]} \leftarrow \Omega_c^{[i-1]} \setminus \omega^{[i]}$
- 13: $\bar{\sigma}_{[i-1]}^2 \leftarrow \frac{1}{H} \sum_{t=1}^H \sum_{\omega=1}^i (\xi_{t\omega} - \mu_t)^2$
- 14: $\bar{\sigma}_{[i]}^2 \leftarrow \frac{1}{H(i+1)} \sum_{t=1}^H \sum_{\omega=1}^{i+1} (\xi_{t\omega} - \mu_t)^2$
- 15: **if** $i = 1$ **then**
- 16: $\Delta^{[1]} \leftarrow 1$
- 17: **else if** $1 < i \leq n$ **then**
- 18: $\delta^{[i]} \leftarrow (\bar{\sigma}_{[i]}^2 - \bar{\sigma}_{[i-1]}^2) / \bar{\sigma}_{[i-1]}^2$
- 19: $\Delta^{[i]} \leftarrow \Delta^{[i-1]} \cup \delta^{[i]}$
- 20: **else**
- 21: $\delta^{[i]} \leftarrow (\bar{\sigma}_{[i]}^2 - \bar{\sigma}_{[i-1]}^2) / \bar{\sigma}_{[i-1]}^2$
- 22: $\Delta^{[i]} \leftarrow \Delta^{[i-1]} \cup \delta^{[i]}$
- 23: $\lambda \leftarrow \lambda + \frac{\Delta^{[i]}}{n} - \frac{\Delta^{[i-n]}}{n}$
- 24: **end if**
- 25: **end while**

Output: $\Omega_c^{[i]}$, $\Omega_s^{[i]}$

reduced sets of the same cardinality. Similar to the forward selection method, Algorithm 2 should therefore merely be considered a useful heuristic.

Following [35], an optimal redistribution of the probabilities in π is carried out. Let Ω_s^* and Ω_c^* , respectively, be the reduced set and the set of non-selected scenarios that result from implementing Algorithm 2. Then, the probability of each scenario in Ω_s^* is defined as:

$$\pi_{\omega}^* = \pi_{\omega} + \sum_{\omega' \in C_{\omega}} \pi_{\omega'}, \forall \omega \in \Omega_s^*, \quad (6)$$

with:

$$C_{\omega} = \{\omega' \in \Omega_c^* \mid \omega = \arg \min_{\omega'' \in \Omega_s^*} \Gamma_{\omega, \omega''}\}, \quad (7)$$

where C_{ω} denotes the subset of Ω_c^* for which the associated probabilities are to be transferred to ω .

IV. RESULTS

To appraise the proficiency of the proposed implementation, it is applied to the case of generating and reducing day-ahead scenarios for Dutch imbalance market prices. Not only are these prices extremely volatile, they are highly uncertain by their very

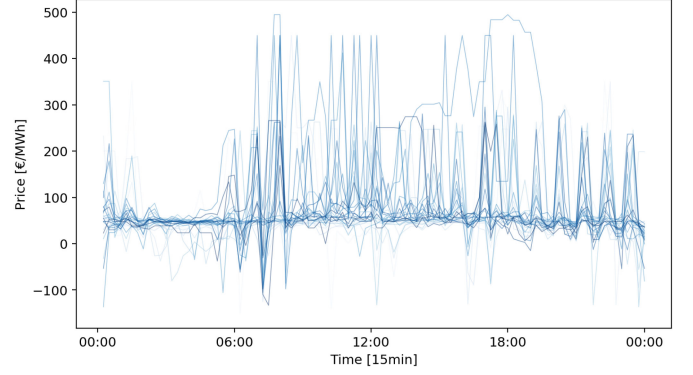


Fig. 4. Observed offtake prices for each day in November 2017.

nature, as they are to a large extent dictated by all unpredictable events in the electricity system and its external influences. Fig. 4 illustrates the erratic nature of Dutch imbalance market prices. It can be observed that scenario generation for this random process is a non-trivial task. Therefore, it provides an interesting test case for the methodology presented in this work. The results of the experiments are outlined in this section.

A. Input Data

As stated in the Dutch imbalance pricing system (IPS) [36], prices are determined at regular time intervals with 15-minute resolution, called program time units (PTUs). The system principally adheres to a single pricing mechanism [37]. However, under specific circumstances, the Transmission System Operator (TSO) may deviate and impose dual pricing, i.e., separate settlement prices for offtake and injection.

The experiments in this study pertain to offtake prices. More precisely, rather than using observed offtake prices at the PTU level, this study relies on minute-by-minute estimations of these prices. Using the IPS, signals indicating the balance state of the grid are combined with information from the bid ladder to arrive at such estimations. The intuition behind doing so is rooted in the enhanced insight in the actual price formation process that is gained by considering minute-by-minute intervals. It should be noted that, as per the rules defined in the IPS, price sequences contain temporal dependencies. The input data used in this study spans the period 2014–2017. Estimated prices are clipped to be in the range $[-70, 150]$ €/MWh and are subsequently allocated to classes with a 1 €/MWh width. This interval contains approximately 96% of the historical data.

Following [4], various seasonal effects are captured by introducing time indexing features $x_k^{sin} = \sin(2\pi k/T)$ and $x_k^{cos} = \cos(2\pi k/T)$, where T denotes the period (e.g., for an input indicating the minute within a PTU, $T = 15$) and $k \in \{1, \dots, T\}$. Time indexing inputs are constructed to reflect intra-PTU, intra-day, and intra-year dynamics.

Concatenating the price class data and time indices results in the final input dimensionality of 227. In this particular case, there are no exogenous variables involved. However, as pointed out in Section III-B, they easily be incorporated as part of the network input when warranted. Scenarios with a time horizon of

148 PTUs (i.e., 37 hours) are generated, such that each scenario spans the period between 11:00 a.m. of the current day D until 12 midnight of day $D + 2$. Note that this particular time window implies compatibility with day-ahead decision-making problems (e.g., day-ahead market trading).

B. LSTM Network Details

In order to generate scenarios for Dutch balancing market prices, a deep long short-term memory neural network as depicted in Fig. 3 is built and trained. Network hyperparameters reported below are optimized using grid search.

1) *Architecture*: The number of time steps in each input sequence is set to 60. Before being transformed to a 3-D tensor to incorporate the batch size, the network input at time t is a 60×227 matrix $X_t \in \mathbb{R}^{60 \times 227}$. The first LSTM layer consists of 96 hidden units, applies a hyperbolic tangent (tanh) activation function and returns the complete output sequence. The second LSTM layer comprises 64 hidden units, but is otherwise equal to its predecessor. The last LSTM layer contains 48 hidden units, has a tanh non-linearity and returns only the output of the final time step. The recurrent layers are followed by two dense layers. To reflect the dimensionality of the target, both layers consist of 221 hidden nodes. The first dense layer uses a rectified linear unit (ReLU) [38], [39] as activation function. The softmax layer at the end of the neural network takes the unprocessed logits of the second dense layer as input.

2) *Training*: Due to the large number of parameters in the model, the potential for over-fitting is implicitly present. To mitigate this concern, both dense layers are preceded by dropout [40] layers during the training phase. The dropout rate is set to 0.2. As an additional preventive measure, the learning procedure is stopped before convergence, i.e., when the increase in validation accuracy stagnates. To reduce the risk of terminating the training process in a local maximum, the patience parameter is set to 50 epochs. Input sequences are presented to the network in batches of size 500. The RMSProp [41] optimizer is employed to minimize the categorical cross-entropy loss, with the learning rate set to 0.001. The system was implemented in Keras [42] using TensorFlow [43] as backend, and trained on a single GPU (NVIDIA Tesla V100, 61 GB RAM).

C. Evaluation Metrics

As this study does not consider a specific optimization problem, the evaluation of the scenario sets warrants alternative metrics. To start with, the first four statistical moments of the generated and reduced scenario sets are calculated and compared to those of the price observations. Furthermore, as interest is primarily in the overall quality of the scenario sets, the adoption of a skill score is desirable. As such, the quality of the generated and reduced sets is assessed using the energy score (ES) [44]:

$$ES = \frac{1}{N_\Omega} \sum_{\omega=1}^{N_\Omega} \|\mathbf{y} - \xi_\omega\| - \frac{1}{2(N_\Omega)^2} \sum_{\omega'=1}^{N_\Omega} \sum_{\omega=1}^{N_\Omega} \|\xi_{\omega'} - \xi_\omega\|, \quad (8)$$

where $\|\cdot\|$ denotes the Euclidean norm and \mathbf{y} the realization of the random process under study. Aside from being a general type

of scoring rule, another important motivation for choosing the ES is the fact it constitutes a strictly proper skill score. Additionally, it allows for direct comparison between scenario sets. The ES is negatively-oriented, i.e., it is inversely related to the skill of the scenario set. The ES for both steps is calculated at the PTU level rather than on a minute-by-minute basis.

D. Performance Benchmarks

The performance of LSTM-based scenario generation is compared to the performance of scenario sets generated with a deep multi-layer perceptron (MLP). MLPs are a specific class of feed-forward ANNs that can incorporate multiple hidden layers between the input and output layers, thereby endowing the FFNN with deep learning capabilities. The MLP-based generation procedure is analogous to the one described in Section III-B, with a few minor deviations due to differences between recurrent and feed-forward ANNs. The network topology and hyperparameters are tuned using grid search. The same training process as outlined in Section IV-B2 is used.

To compare the overall skill of sets that have been reduced from a common initial set, this study proposes a relative energy score (RES), which will be defined as:

$$RES_{\Omega_r} = \frac{ES_{\Omega_r}}{ES_\Omega}, \quad (9)$$

where ES_{Ω_r} and ES_Ω respectively denote the energy score of the reduced set and the generated set.

E. Test Results

The method proposed in this study was tested over the 14-day period from January 3, 2018 through January 16, 2018. For each day, a set of 500 scenarios is generated. It is to be noted that, as mentioned in Section IV-A, each scenario does not only span the operational day, but also the 52 preceding PTUs.

1) *Distributions of Scenario Sets*: Fig. 5(a) depicts the percentile bounds containing 80% and 95% of the generated scenarios as well as the realized offtake prices for several days in the test period. It can be visually observed that the majority of realized prices are well-encapsulated by the generated scenario sets. On several occasions the observed prices are not covered by the percentile bounds, which is caused by the chosen imbalance price range $[-70, 150]$ €/MWh. Approximately 6% of the observations are outside this price range over the entire test period.

Subsequently, each of the 14 initially generated scenario sets is reduced by means of the variance-based reduction method outlined in Algorithm 2. Equivalently to Fig. 5(a), the percentile bounds of the reduced scenario sets for days 3, 10 and 14 are shown in Fig. 5(b). It can be seen that the methodology allows for a drastic reduction in scenario set cardinality without a significant loss in coverage. Computed over the complete test period, the average cardinality of the reduced sets is 36.4, constituting a reduction of 93% relative to the initially generated sets with a cardinality of 500.

Fig. 6 provides an intra-day perspective by exhibiting distribution plots of generated and reduced scenarios for a selection

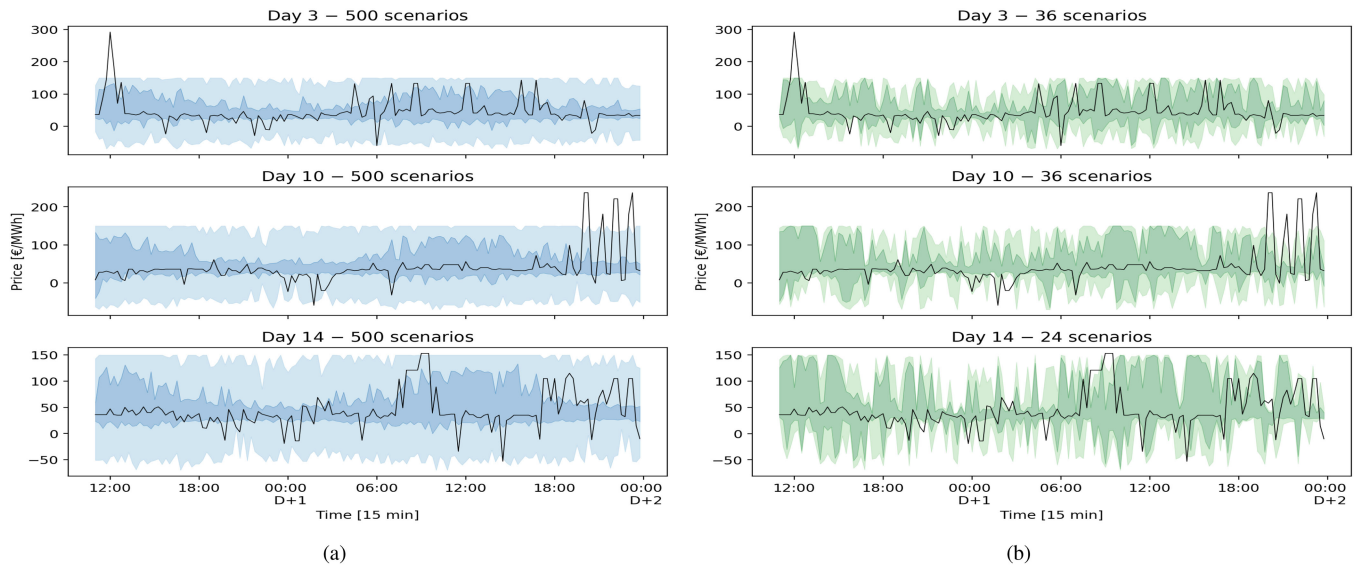


Fig. 5. Scenario sets versus offtake prices for a selection of days. The dark and light shaded areas respectively represent the 80% and 95% percentile bounds. Set cardinality is indicated at the top of each subplot. Realized prices are shown by the black line. (a) Generated scenario sets. (b) Reduced scenario sets.

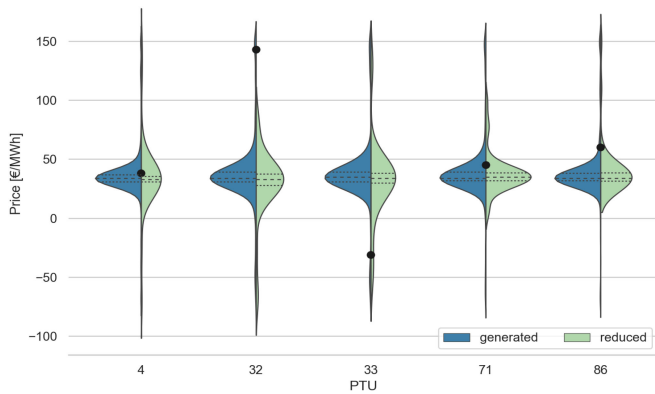


Fig. 6. Distribution comparison between generated and reduced sets for various PTUs of January 10, 2018. The dashed lines represent the quartiles of the distribution. Observed offtake prices are denoted by the black dots.

of PTUs of a particular day, i.e., January 10, 2018. Realized offtake prices, along with the median and the interquartile range of the distributions, are shown as well. It can be observed that the quartiles of the reduced scenarios are quite similar to those of the generated scenarios. Additionally, the distributions are more leptokurtic during PTUs 32, 33 and 71. Indeed, this appears to reflect price dynamics of particular PTUs that are typically characterized by high volatility and strong price ramping, e.g., the PTUs immediately before and after usual business hours.

2) *Scenario Generation Benchmark*: Fig. 7 shows the ES for each day in the 14-day test period for both the proposed method of scenario generation and the MLP benchmark. Having a lower energy score at each instance, LSTMs can be said to perform better and generate scenarios sets with greater skill than MLPs do. Measured over the complete test period, the former yield scenario sets with, on average, 13% lower ES than the latter. The differences between the two range from 2% (day 8) to 28% (day 9). The relative outperformance of LSTMs can be

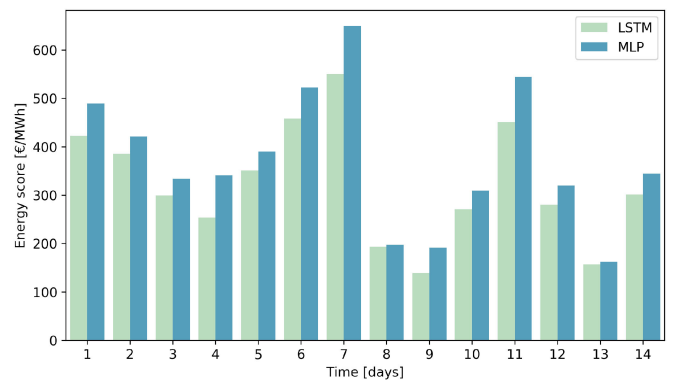


Fig. 7. Energy score comparison between scenario sets generated with an LSTM-based recurrent neural network and those resulting from an MLP-based feed-forward network for each day in the test period.

TABLE I
STATISTICAL MOMENTS OF OBSERVED PRICES, GENERATED SETS AND REDUCED SETS, COMPUTED OVER 14-DAY TEST PERIOD

	(1) Mean	(2) Variance	(3) Skewness	(4) Kurtosis	N
Observed	42.05	1514	0.92	2.83	1344
LSTM	39.94	1268	1.15	4.61	672 000
MLP	33.14	665	2.08	9.65	672 000
Reduced	40.92	1397	1.03	3.27	48 960

attributed to their internal memory capacity and the ability to store information over multiple time steps.

As a means of further comparison, the first four statistical moments for the offtake price realizations and the generated scenario sets are presented in Table I. The reported values are calculated over the entire test period and are based only on the 96 data points of the operational day as a preemptive measure

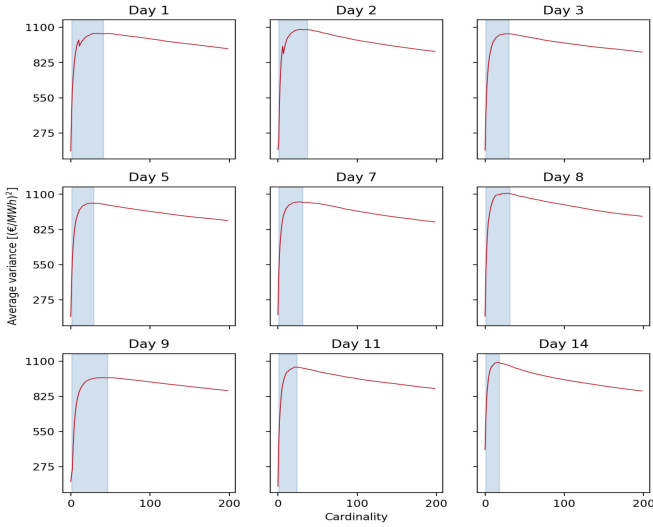


Fig. 8. Evolution of the average variance $\bar{\sigma}^2$ as more scenarios are selected and added to the reduced set. The average variance is defined as in Algorithm 2. The shaded area represents the cardinality of the reduced set that is found by terminating the reduction process when the continuation criterion is rendered invalid. Each subplot is a selected day in the test period.

against possible misrepresentation of distributional characteristics. In spite of the fact that the proposed approach is not grounded in moment-matching techniques, Table I provides further corroborative indications on the feasibility of taking a class-driven, LSTM-based approach in finding acceptable approximations of distributions of interest.

3) *Scenario Reduction Benchmark*: The scenario reduction component of the proposed methodology inherently contains the assumption that variance, or more precisely, the evolution of its change, may serve as a proxy to informational value or content. Hence, scenario reduction with a variance-based continuation criterion may be viewed as exhausting the informational value of the original set, i.e., within the bounds set by the algorithmic parameters. In other words, reducing the original set to a size lower than the variance-based cardinality might imply missing out on relevant sample paths. Conversely, setting a cardinality higher than the one obtained by the variance-based procedure is expected to offer negligible incremental gains in terms of information content.

For a selection of days in the test period, Fig. 8 shows the evolution of the mean variance $\bar{\sigma}^2$, averaged over the temporal dimension (see Algorithm 2, line 13), in the reduced set as its cardinality iteratively increases. Initially, the variance increases steeply. It then reaches a maximum before gradually decreasing as more scenarios of the original set are added to reduced set. The shaded area illustrates the impact of the continuation criterion on the size of the reduced set when using Algorithm 2 ($\theta = 0.01$, $n = 5$). To investigate how this affects the relative performance of the proposed reduction method, its RES score is compared to the RES of various reduced sets that are obtained by applying a standard forward selection procedure with size equal to a chosen proportion of the variance-based cardinality. Fig. 9 presents the RES during the test period for a variety of such proportions. In the top two subplots, the sets reduced by

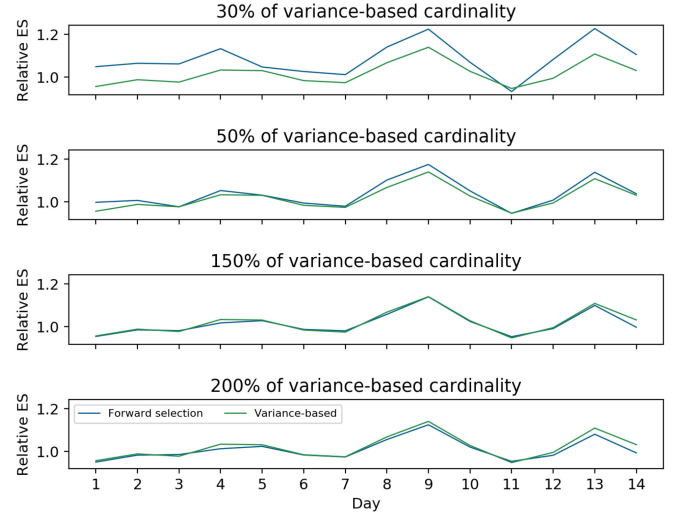


Fig. 9. Relative energy scores during the test period for sets reduced with both the variance-based reduction method proposed in this study and the standard forward selection procedure with proportional pre-defined cardinality.

the forward selection method contain less scenarios than ones reduced with Algorithm 2. It can be seen that the variance-based method yields lower relative energy scores for most days, which is in accordance with expectations.

From the bottom two subplots, where the forward selection-based reduced sets have greater cardinality than the variance-based reduced sets, it appears that increasing the cardinality above the level obtained using the proposed continuation criterion does not yield significant improvements in terms of the RES. A probable interpretation of this result is that virtually all relevant information contained in the generated set is extracted and transported to the reduced set once the continuation criterion ($\theta < \lambda$) is rendered invalid; a reading that is also substantiated by Fig. 8. It appears that although the sets that are reduced by means of standard forward reduction comprise, respectively, 50% and 100% more scenarios, these extra scenarios add but scant information to the reduced sets. This finding corroborates the hypothesis that continuing to add scenarios to the reduced set becomes futile past a certain cardinality, at least with respect to the RES. Indeed, the ability of the proposed scenario reduction algorithm seems to be supported by the results in Fig. 9.

4) *Minor Test Results*: Finally, several tests have been performed to evaluate the impact of bin width on the presented results. Without providing further details, it should be stated that relative energy scores are impacted unfavorably when the chosen bin width increases. A potential explanation for this can be found in the fact that increasing the bin width is expected to increase the first term in (8) and, therefore, the energy score.

V. CONCLUSION

This paper proposed a novel method for the generation and subsequent reduction of scenario sets to adequately represent price uncertainty as related to decision-making problems in electricity markets. A class-driven implementation facilitates the employment of modern and powerful deep recurrent neural

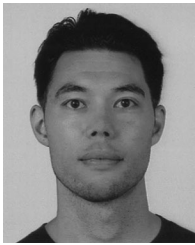
network structures for scenario generation. The reductive component of the methodology builds on well-established forward selection procedures and extends it by the inclusion of a variance-based continuation criterion which allows for automated scenario reduction. The presented system was tested on Dutch imbalance prices. The results of the experiments indicate that the approach is able to generate realistic scenarios which reflect highly erratic dynamics in the data. Additionally, they express the capability of the suggested reduction method to select an adequate subset of scenarios without the need to explicitly predetermine its cardinality. The approach can be used in combination with stochastic optimization or other uncertainty-aware decision-making methods, such as (deep) reinforcement learning.

In the light of future research, one potentially interesting path of inquiry might be to investigate whether the proposed scenario generation method can be extended to high-dimensional contexts. For example, when relying on one-hot encoding for the representation of classes, the number of dimensions and parameters increase linearly with the number of classes under consideration. When the class-space is very large, this may lead to several issues, e.g., over-fitting or large memory requirements.

REFERENCES

- [1] A. Fabbri, T. G. San Román, J. R. Abbad, and V. H. M. Méndez Quezada, "Assessment of the cost associated with wind generation prediction errors in a liberalized electricity market," *IEEE Trans. Power Syst.*, vol. 20, no. 3, pp. 1440–1446, Aug. 2005.
- [2] R. Doherty and M. O'Malley, "A new approach to quantify reserve demand in systems with significant installed wind capacity," *IEEE Trans. Power Syst.*, vol. 20, no. 2, pp. 587–595, May 2005.
- [3] F. Bouffard and F. D. Galiana, "Stochastic security for operations planning with significant wind power generation," *IEEE Trans. Power Syst.*, vol. 23, no. 2, pp. 306–316, May 2008.
- [4] S. I. Vagropoulos *et al.*, "ANN-based scenario generation methodology for stochastic variables of electric power systems," *Electr. Power Syst. Res.*, vol. 134, pp. 9–18, May 2016.
- [5] M. A. Plazas, A. J. Conejo, and F. J. Prieto, "Multimarket optimal bidding for a power producer," *IEEE Trans. Power Syst.*, vol. 20, no. 4, pp. 2041–2050, Nov. 2005.
- [6] J. Cabero *et al.*, "A medium-term integrated risk management model for a hydrothermal generation company," *IEEE Trans. Power Syst.*, vol. 20, no. 3, pp. 1379–1388, Aug. 2005.
- [7] T. Li and M. Shahidehpour, "Risk-constrained generation asset arbitrage in power systems," *IEEE Trans. Power Syst.*, vol. 22, no. 3, pp. 1330–1339, Aug. 2007.
- [8] A. J. Conejo, R. García-Bertrand, M. Carrión, Á. Caballero, and A. de Andrés, "Optimal involvement in futures markets of a power producer," *IEEE Trans. Power Syst.*, vol. 23, no. 2, pp. 703–711, May 2008.
- [9] J. M. Morales, S. Pineda, A. J. Conejo, and M. Carrión, "Scenario reduction for futures market trading in electricity markets," *IEEE Trans. Power Syst.*, vol. 24, no. 2, pp. 878–888, May 2009.
- [10] J. Dupačová, G. Consigli, and S. W. Wallace, "Scenarios for multi-stage stochastic programs," *Ann. Operations Res.*, vol. 100, pp. 25–53, Dec. 2000.
- [11] N. Gülpinar, B. Rustem, and R. Settergren, "Simulation and optimization approaches to scenario tree generation," *J. Econ. Dyn. Control*, vol. 28, pp. 1291–1315, Apr. 2004.
- [12] G. C. Pflug, "Scenario tree generation for multiperiod financial optimization by optimal discretization," *Math. Program.*, vol. 89, pp. 251–271, Jan. 2001.
- [13] K. Høyland and S. W. Wallace, "Generating scenario trees for multistage decision problems," *Manage. Sci.*, vol. 47, pp. 295–307, Feb. 2001.
- [14] K. Høyland, M. Kaut, and S. W. Wallace, "A heuristic for moment-matching scenario generation," *Comput. Optim. Appl.*, vol. 24, pp. 169–185, Feb. 2003.
- [15] P. Beraldi, F. De Simone, and A. Violi, "Generating scenario trees: A parallel integrated simulation-optimization approach," *J. Comput. Appl. Math.*, vol. 233, pp. 2322–2331, Mar. 2010.
- [16] G. Sideratos and N. D. Hatziaargyriou, "Probabilistic wind power forecasting using radial basis function neural networks," *IEEE Trans. Power Syst.*, vol. 27, no. 4, pp. 1788–1796, Nov. 2012.
- [17] M. Cui, D. Ke, Y. Sun, D. Gan, J. Zhang, and B. M. Hodge, "Wind power ramp event forecasting using a stochastic scenario generation method," *IEEE Trans. Sustain. Energy*, vol. 6, no. 2, pp. 422–433, Apr. 2015.
- [18] Y. Chen, Y. Wang, D. Kirschen, and B. Zhang, "Model-free renewable scenario generation using generative adversarial networks," *IEEE Trans. Power Syst.*, vol. 33, no. 3, pp. 3265–3275, May 2018.
- [19] M. Olsson and L. Söder, "Generation of regulating power price scenarios," in *Proc. Int. Conf. Probabilistic Methods Appl. to Power Syst.*, 2004, pp. 26–31.
- [20] I. G. Jensen, N. F. Møller, G. Pantuso, and N. Juul, "A comparison of scenario generation methods for the participation of electric vehicles in electricity markets," *Int. Trans. Electr. Energy Syst.*, vol. 29, no. 4, pp. 1–11, 2019.
- [21] Q. Zhou, L. Tesfatsion, and C. C. Liu, "Scenario generation for price forecasting in restructured wholesale power markets," in *Proc. IEEE PES Power Syst. Conf. Expo.*, 2009, pp. 1–8.
- [22] R. Hochreiter and G. C. Pflug, "Financial scenario generation for stochastic multi-stage decision processes as facility location problems," *Ann. Oper. Res.*, vol. 152, pp. 257–272, Jul. 2007.
- [23] J. Dupačová, N. Gröwe-Kuska, and W. Römisch, "Scenario reduction in stochastic programming - An approach using probability metrics," *Math. Program.*, vol. 95, no. 3, pp. 493–511, Mar. 2003.
- [24] H. Heitsch and W. Römisch, "Scenario reduction algorithms in stochastic programming," *Comput. Optim. Appl.*, vol. 24, pp. 187–206, Feb. 2003.
- [25] Y. Dvorkin, Y. Wang, H. Pandzic, and D. Kirschen, "Comparison of scenario reduction techniques for the stochastic unit commitment," in *Proc. IEEE Power Energy Soc. Gen. Meeting*, 2014, pp. 1–5.
- [26] Y. Wang, Y. Liu, and D. S. Kirschen, "Scenario reduction with submodular optimization," *IEEE Trans. Power Syst.*, vol. 32, no. 3, pp. 2479–2480, May 2017.
- [27] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, 2014, vol. 2, pp. 3104–3112.
- [28] Y. Bengio, P. Simard, and P. Frasconi, "Learning long term dependencies with gradient descent is difficult," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, Mar. 1994.
- [29] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proc. 30th Int. Conf. Mach. Learn.*, 2013, vol. 28, pp. 1310–1318.
- [30] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, pp. 1735–1780, Nov. 1997.
- [31] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural Comput.*, vol. 12, no. 10, pp. 2451–2471, Oct. 2000.
- [32] B. J. Claessens, P. Vrancx, and F. Ruelens, "Convolutional neural networks for automatic state-time feature extraction in reinforcement learning applied to residential load control," *IEEE Trans. Smart Grid*, vol. 9, no. 4, pp. 3259–3269, Jul. 2018.
- [33] T. Chen, M. R. Min, and Y. Sun, "Learning k-way d-dimensional discrete codes for compact embedding representations," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, vol. 80, pp. 854–863.
- [34] A. Graves, A. R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2013, vol. 38, pp. 6645–6649.
- [35] A. J. Conejo *et al.*, *Decision Making Under Uncertainty in Electricity Markets*, 1st ed. New York, NY, USA: Springer, 2010.
- [36] TenneT TSO B.V., "The imbalance pricing system," Tech. Rep., Oct. 2016.
- [37] R. A. van der Veen and R. A. Hakvoort, "The electricity balancing market: Exploring the design challenge," *Util. Policy*, vol. 43, pp. 186–194, Dec. 2016.
- [38] R. H. Hahnloser, R. Sarapeshkar, M. A. Mahowald, R. J. Douglas, and H. S. Seung, "Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit," *Nature*, vol. 405, pp. 947–951, Jun. 2000.
- [39] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?" in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, 2009, pp. 2146–2153.
- [40] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, Jun. 2014.

- [41] T. Tieleman and G. Hinton, "RMSPProp: Divide the gradient by a running average of its recent magnitude," Coursera: Neural networks for machine learning, Tech. Rep. 6, 2012.
- [42] F. Chollet, "Keras: The Python deep learning library," 2015. [Online]. Available: <https://github.com/fchollet/keras>
- [43] M. Abadi *et al.*, "TensorFlow: A system for large-scale machine learning on heterogeneous distributed systems," in *Proc. 12th USENIX Conf. Oper. Syst. Des. Implement.*, 2016, pp. 265–283.
- [44] T. Gneiting, L. I. Stanberry, E. P. Grimit, L. Held, and N. A. Johnson, "Assessing probabilistic forecasts of multivariate quantities, with an application to ensemble predictions of surface winds," *Test*, vol. 17, pp. 211–235, Aug. 2008.



Bart Stappers received the M.Sc. degree in finance from Tilburg University, Tilburg, The Netherlands, in 2014. After graduating, he has been involved in applied machine learning research for energy trading purposes at Scholt Energy, Valkenswaard, The Netherlands. He is currently with Scholt Energy and the Eindhoven University of Technology, Eindhoven, The Netherlands, where he is working toward the Ph.D. degree in electrical engineering. His current research interests include machine learning, electricity markets, algorithmic trading, reinforcement learning, and mathematical optimization.



Nikolaos G. Paterakis (Member, IEEE) received the Dipl.Eng. degree in electrical and computer engineering from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 2013, and the Ph.D. degree in industrial engineering and management from the University of Beira Interior, Covilhã, Portugal, in 2015. From October 2015 to March 2017, he was a Postdoctoral Fellow with the Department of Electrical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands, where he is currently an Assistant Professor. His current research interests include electricity markets, power system operations, and applications of machine learning and optimization techniques. He is an Associate Editor of the *IET Renewable Power Generation*, an Editor of *MDPI Applied Sciences*, and a Review Editor of *Frontiers in Energy Research (Smart Grids)*. He is also a Reviewer of more than 30 journals, while he was recognized as an Outstanding Reviewer of IEEE TRANSACTIONS ON SUSTAINABLE ENERGY (2016) and as one of the Best Reviewers of IEEE TRANSACTIONS ON SMART GRID (2015 and 2017).



Koen Kok (Senior Member, IEEE) received the B.Sc. degrees in electrical engineering and in technical informatics, the M.Sc. degree in computer science from the University of Groningen, Groningen, The Netherlands, and the Ph.D. degree in computer science from the Vrije Universiteit Amsterdam, Amsterdam, The Netherlands, for his thesis on smart grid coordination mechanisms based on distributed software technology. He is currently a Senior Scientist with TNO, The Hague, The Netherlands, the largest applied research institute in Netherlands, and a Full

Professor with the Electrical Energy Systems Group, Eindhoven University of Technology, Eindhoven, The Netherlands. Formerly, he was with the Energy Research Center of the Netherlands, VU University Amsterdam, and the Technical University of Denmark. He has extensive research experience in the fields of market-based control of power systems, smart-grid ICT architectures, and integration of distributed energy resources and demand response in the electricity system. The key results of his research have been field deployed, commercialized, and/or made available in open-source.



Madeleine Gibescu (Member, IEEE) received the Dipl.Eng. degree in power engineering from University Politehnica, Bucharest, Romania, in 1993, and the M.Sc. and Ph.D. degrees in electrical engineering from the University of Washington, Seattle, WA, USA, in 1995 and 2003, respectively. She was a Research Engineer with ClearSight Systems, and as a Power Systems Engineer with Alstom Grid, in Washington and California, USA. In 2007, she joined the Department of Electrical and Sustainable Energy, Delft University of Technology, as an Assistant Pro-

fessor, and between 2013 and 2018, she was an Associate Professor of smart power systems with the Electrical Energy Systems Department, Eindhoven University of Technology, The Netherlands. Since September 2018, she has been a Professor in the area of integration of intermittent renewable energy with the Copernicus Institute of Sustainable Development, Utrecht University, Utrecht, The Netherlands.