Using Machine Learning to Enable Probabilistic Reliability Assessment in Operation Planning

Laurine Duchesne, Efthymios Karangelos, Louis Wehenkel
Montefiore Institute - Department of EE&CS
University of Liège, Liège, Belgium
{l.duchesne, e.karangelos, l.wehenkel}@uliege.be

***Abstract*—In the context of operation planning, probabilistic reliability assessment essentially boils down to predicting, efficiently and with sufficient accuracy, various economic and reliability indicators reflecting the expected performance of the system over a certain look-ahead horizon, so as to guide the operation planner in his decision-making. In order to speed-up the crude Monte Carlo approach, which would entail a very large number of heavy computations, we propose in this paper an approach combining Monte Carlo simulation, machine learning and variance reduction techniques such as control variates. We provide an extensive case study testing this approach on the three-area IEEE-RTS96 benchmark, in the context of day-ahead operation planning while using a security constrained optimal power flow model to simulate real-time operation according to the N-1 criterion. From this case study, we can conclude that the proposed approach allows to reduce the number of heavy computations by about an order of magnitude, without sacrificing accuracy.**

***Index Terms*—Reliability assessment, Operation Planning, Machine learning, Monte Carlo simulation, Variance reduction, Control variates, Security Constrained Optimal Power Flow**

I. Introduction

In the context of operational planning, power system *reliability management* aims at taking, as early as necessary, decisions facilitating the operation of the system over a future target horizon. For example, decisions may be taken to adjust market coupling capacities, to postpone planned outages, or to acquire flexibility resources for real-time operation. The question entails anticipating the decisions to be made by the real-time operator throughout the target horizon, as per the applicable reliability criterion, and complementing these by already taking suitable decisions ahead in time. In practice, such a question is typically addressed in a progressive manner using *reliability assessment* (*i.e.*, evaluating the anticipated outcomes of real-time operation) to inform *reliability control* (*i.e.*, selecting which decisions to commit to achieve a near optimal socio-economic tradeoff) [1].

In this paper we address the day-ahead operation planning problem as a 'template', and focus on reliability assessment while adopting a probabilistic approach. Specifically, we aim at evaluating over the 24-hours of the next day the expected costs associated to operating the system. Our approach relies on two fundamental components, namely a probabilistic model of the exogenous uncertainties and a computational model of real-time operation. The former describes the possible operating conditions that could be encountered the next day by the operator. The latter is a *Security Constrained Optimal Power Flow* (SCOPF) model, formalizing his choice of preventive (pre-contingency) and/or corrective (post-contingency) controls in real-time.

Using these two components, it is in principle possible to solve the operational planning reliability assessment problem by adopting a crude Monte Carlo approach: sample a suitable number of scenarios of next-day operating conditions according to the probabilistic model, run the SCOPF model to gather for each hour of each scenario values of operating costs, average these quantities over the set of simulated scenarios to yield an estimate of the expected operating costs. An obvious drawback of this approach is however its computational burden, due to the very large number of SCOPF computations that would typically be necessary.

A. Proposal & experimental setup

To make this approach more tractable, we propose to work along two complementary directions, namely i) speeding up the individual computations by leveraging Machine Learning to replace the SCOPF computations by a much faster proxy of real-time operator response, and ii) instead of using the crude Monte Carlo approach, leveraging variance reduction techniques (more specifically, control variates approaches), so as to reach the same accuracy while relying on fewer SCOPF computations. The proposed approach works as follows:

- During a first stage, we sample a number of scenarios and solve them with the SCOPF model in order to compute values of the cost function. It yields a dataset of input-output pairs which is then exploited according to the machine learning methodology presented in [2] in order to build simplified models (*proxies*) of real-time operation.
- During a second stage, we exploit the learnt proxies together with the control variates approaches, in order to estimate the expected values of the concerned cost components, while also exploiting a second (independent) sample of scenarios solved with the SCOPF model.

This research has been funded by the European FP7 project GARPUR. The scientific responsibility rests with the authors.

Comparing the accuracy obtained by the proposed approach to that of the crude Monte Carlo approach with an identical total budget of SCOPF computations, allows one to infer the potential computational speed-up of the proposed approach for a given target accuracy.

We test this approach on the (three-area) IEEE-RTS96, while modeling real-time operation according to the N-1 criterion, and in order to estimate the expected value of different components of the real-time operating cost, such as preventive control costs, and corrective control costs.

B. Paper organization

The rest of the paper is organised in four further sections: II. Problem statement, background and proposed methods, III. Case study, IV. Related works and contribution, V. Conclusion and further research.

Moreover, we also point the interested reader to the appendix for the detailed formulation of mathematical models and machine learning methods adopted in this work.

II. Problem statement, background, and methods

A. Problem statement

In day-ahead (da), the operation planner needs to assess how the operation of the system would turn out during the next day (nd), and if necessary takes some decision to ensure that real-time operation will turn out in the best possible way. While doing this, he faces uncertainties ξ_{nd} about the exogenous factors (renewable generation, weather conditions, demand, etc.) that will influence the outcome of reliability management during the next day, and he needs to anticipate how the control-room operators will react to them in real-time.

In this paper, we model the behaviour of the real-time control-room operators by a Security Constrained OPF model, aiming at meeting the N-1 criterion at the least cost.

Furthermore, in day-ahead conditions, we suppose that the operation planning engineer disposes of a generative model allowing him to sample scenarios of next-day conditions $\{\xi^1_{nd}, \xi^2_{nd}, \ldots\}$ and to plug them into a SCOPF computational module that will reveal the response of real-time operation based on the N-1 criterion[1]. We suppose that the planning engineer wants to evaluate the consequence of a given day-ahead decision, by estimating the resulting *expectation* of real-time operation cost components.

We thus focus on the following computational problem: *given a day-ahead decision, a generative model of exogenous uncertainties over a horizon of 24 next-day hourly time-steps, and a software implementing a sequence of SCOPF problems over a next-day scenario (or trajectory), how to minimize the number of SCOPF calls to obtain a sufficiently accurate estimate of the expected next-day operating costs.*

[1]These next-day scenarios are defined over a horizon of 24 hours via 24 time steps of one hour, and the SCOPF module is then actually applied for each such scenario in a sequential way over the 24 time steps, to reveal the outcome of real-time operation along such a scenario.

B. Background

1) Crude Monte Carlo approach: The crude Monte-Carlo approach [3] uses the generative model in order to sample next-day scenarios and then runs the SCOPF model on each one of them to compute the costs and constraint violation indicators. It averages such costs over a number n of scenarios until the accuracy is sufficient. Denoting by $y(\xi_{nd})$ an output of the SCOPF model of real-time operation[2] and by $\{\xi^1_{nd}, \xi^2_{nd}, \ldots, \xi^n_{nd}\}$ an *i.i.d* sample[3] of exogenous scenarios of next day conditions, the crude Monte Carlo approach estimates the mathematical expectation μ_y of y by:

$$\hat{\mu}_y = n^{-1} \sum_{i=1}^{n} y(\xi^i_{nd}), \tag{1}$$

and its standard deviation σ_y by

$$\hat{\sigma}_y = \sqrt{n^{-1} \sum_{i=1}^{n} \left(y(\xi^i_{nd}) - \hat{\mu}_y\right)^2}. \tag{2}$$

It is well known that the crude Monte Carlo estimator $\hat{\mu}_y$ is an unbiased estimator of μ_y [3]. Further, the standard error of $\hat{\mu}_y$ depends on the value of σ_y and on the sample size in the following way

$$\sigma_{\hat{\mu}_y} = \sqrt{n^{-1}}\sigma_y. \tag{3}$$

Thus, in the crude Monte-Carlo approach, one classically determines the required number of samples by checking the ratio $\sqrt{n^{-1}}\hat{\sigma}_y/\hat{\mu}_y$. Typically the number n of simulations is determined so that this quantity is smaller than 1%.

2) Variance reduction by using the control-variates approach: The crude Monte Carlo approach requires the computation of $y(\xi_{nd})$ over a set of next day scenarios that is large if the variance of y is large. Each one of these computations implies solving a sequence of SCOPF problems over the 24 hourly time-steps of the next day along a realization ξ^i_{nd} of exogenous variables. The total number of required samples n to reach a required level of accuracy is thus larger if the uncertainties in ξ_{nd} induce higher variabilities of costs and other indicators that need to be computed. In order to reduce the computational burden, various approaches have been proposed in the literature to reduce the variance of Monte Carlo methods. The one that we propose to investigate in this paper is called the "control variates" approach and is based on the following rationale.

Suppose that we dispose of a "proxy" allowing us to compute at a very cheap cost an approximation $y_p(\xi_{nd})$ of the function $y(\xi_{nd})$. If $y_p(\xi_{nd})$ is very cheap to compute, we can estimate its mean μ_{y_p} by crude Monte Carlo, using a very large number of samples at negligible cost. Since $\mu_y = \mu_{y_p} + \mu_{y-y_p}$, we can thus reframe the estimation of μ_y by estimating separately μ_{y_p} and μ_{y-y_p}. If at the same

[2]In our case, this value y would be obtained as the sum of hourly values of some term of the SCOPF objective function computed for the 24 successive hourly time-steps composing a scenario ξ_{nd}.

[3]Note that it is not required that the successive *time-steps* in a given scenario are *i.i.d.* Rather the n different *scenarios* are assumed i.i.d.

time $x(\xi_{nd}) = y(\xi_{nd}) - y_p(\xi_{nd})$ has a small variance, we then can estimate $\mu_x = \mu_{y-y_p}$ by using the crude Monte Carlo approach, at sufficient accuracy, with a number of samples that will be relatively small (in the extreme case, $x(\xi_{nd})$ is almost constant, and its expectation could be estimated by using a very small number of samples).

Thus, the control variates approach consists of seeking a good proxy y_p ("good" meaning that σ_x is much smaller than σ_y) so that using the estimate:

$$\hat{\mu}_y^{y_p} = \mu_{y_p} + m^{-1} \sum_{i=1}^{m} x(\xi_{nd}^i) \tag{4}$$

would need a much smaller number m of samples than the crude Monte Carlo approach applied directly to y, while obtaining the same level of accuracy.

Indeed, the control variates estimator is by construction also an unbiased estimator of μ_y, since $\mu_x = \mu_y - \mu_{y_p}$ and since the second term of eq. (4) is itself an unbiased estimator of μ_x. On the other hand its standard error is given by

$$\sigma_{\hat{\mu}_y^{y_p}} = \sqrt{m^{-1}}\sigma_x, \tag{5}$$

so that for a same level of accuracy than the crude Monte Carlo approach this method would need a sample size m equal to

$$m = n\frac{\sigma_x^2}{\sigma_y^2}, \tag{6}$$

which will be (much) smaller than n if σ_x^2 is (much) smaller than σ_y^2. Notice that we do not suppose in any case that $\mu_{y_p} = \mu_y$; indeed if this could be guaranteed we could say in advance that $\mu_x = 0$, so that we would not need any additional samples to estimate this quantity.

C. Proposed methods

1) Building control variates by supervised machine learning: In some cases, a suitable proxy of the quantity y to estimate can be hand-crafted. In our context, it is unlikely that this can be done, given the complexity of the relationship between next day conditions ξ_{nd} and the considered variables y (cost indicators reporting the outcome of real-time operation).

We thus propose to use supervised machine learning in order to build, from a sample of pairs $(\xi_{nd}^i, y^i = y(\xi_{nd}^i))^{i=1,k}$, a proxy y_p, and then use the inferred proxy within the control variates approach. Given a total budget n of next-day possible scenarios for which we can compute the exact value of y, we investigate how to split them in two parts in the best way, the first k being used to learn a proxy, and the remaining $m = n - k$ being used in the control variates approach. We will also consider different settings for applying the machine learning approach, so as to build for a given learning sample size k the most accurate proxy y_p of y.

2) Stacked Monte Carlo approach (SMC): In the SMC approach [4] a sample of pairs $(\xi_{nd}^i, y^i)^{i=1,n}$ is used in a more intensive way to build an estimate of μ_y. The sample is first split into V folds of $k_v \approx \frac{n}{V}$ pairs, then for each fold $v \in \{1, \ldots, V\}$ a proxy y_p^v is built by using machine learning applied to the union of all other $V - 1$ folds, and then used to predict the value $x^v = y - y_p^v$ over the held-out fold only. Also, for each fold, the value of $\mu_{y_p^v}$ is estimated separately with high accuracy. The final SMC estimate is computed as follows

$$\hat{\mu}_y^{SMC} = V^{-1} \sum_{v=1}^{V} \left(\mu_{y_p^v} + k_v^{-1} \sum_{j=1}^{k_v} x^v(\xi^{v_j}) \right), \tag{7}$$

where v^j denotes the index of the j-th sample of fold v.

Rather than splitting the whole sample once in two folds, one for training a proxy and the rest for use in the control variates approach, this approach uses all available samples both for training and for the control variates estimate, while however avoiding to use any sample both in a certain training sample and in the corresponding control variate estimate. It is therefore likely to lead to an even better variance reduction.

D. Overall proposed study approach

In order to study the effectiveness of the above approaches, we first evaluate the baseline, i.e. the Crude Monte Carlo approach, in terms of the number n of samples required to yield a reasonable target precision, based on empirical simulations with a given test system and target quantity y to be evaluated in day-ahead conditions.

Next, we investigate how to exploit in the best way the set of n samples by studying different settings of the approaches proposed: i.e. different ways of splitting into k and $n - k$ samples, different machine learning methods used to build the proxies, and different ways of stating these machine learning problems. The goal of this study is to determine the gain in accuracy that can be obtained with the proposed approaches with respect to the crude Monte Carlo approach, while using a same budget for the detailed simulations (and SCOPF computations) of day-ahead conditions.

Finally, we translate the gain in accuracy in gain in computational requirements, by determining how many samples n' would be needed to obtain the same level of accuracy than our approaches, while using the crude Monte-Carlo approach.

III. Case study

A. Test system, uncertainties, and real-time operation model

We explore the applicability of this approach on the 3-area version of the IEEE RTS-96 [5], as modernized with the addition of 19 wind power generators by [6]. Our studies refer to the 1st day of the year, for a peak demand of 3135 MW, per area.

1) Horizon, uncertainties & temporal resolution: We place ourselves 12 hours before the start of the day under consideration (that is, at noon of the previous day) and assume that, at this point in time, the yet unresolved uncertainties restrict to the forecast errors of wind power injections and load demand. Our modeling approach for these exogenous uncertainties is based on our earlier work detailed in [7]. Here, we build on top of this model by further considering the spatial correlation between the forecast errors concerning

power injections/demands located in the same area of the 3-area system. To do so, we assume that the forecast error of each power injection/demand is composed by a global and a local term. The global term is common for all wind power generators/loads in the same area while the local term is distinctive per each individual power injection/demand. Adopting a one-hour interval as the time step for real-time operation, we exploit this model to generate via Monte Carlo simulations scenarios of 24 forecast error realizations per load demand and wind power generator.

2) Day-ahead planning model: In order to simulate day-ahead decision making, we regard the commitment status and economic dispatch of all dispatchable generators as well as the provisional curtailment of wind power generation. To fix such decisions, we solve a multi-period Security Constrained Optimal Power Flow (SCOPF) problem in anticipation of the demand values from the original system description [5] and the wind power "favorable" forecast from [6]. More specifically, we use the DC power flow approximation while taking into account the N-1 criterion concerning all transmission system components (*i.e.*, transmission lines, cables, and transformers). Further, as a preventive measure to address the wind/load uncertainties we impose 300 MW up-ward and down-ward spinning reserve capacity constraints. The full mathematical formulation is available in Appendix A-B.

3) Real-time operation model: To model the system trajectory within the day under consideration, we sequentially go through the 24 single period real-time operation instances. That is, for any such instance, we i) generate a set of wind power & load demand realizations, and, ii) re-compute real time preventive and/or corrective (alternatively, pre- and/or post-contingency) control actions to maintain conformity with the N-1 criterion[4]. These decisions should adjust to the most recent forecast error realizations, and are taken with a single-hour horizon. For the sake of simplicity, the real-time control decisions already applied within the trajectory do not constrain the candidate decision space. It is only potentially constrained by day-ahead decisions. Nevertheless, the existence of control actions to achieve the N-1 criterion is certainly not guaranteed at this stage. Rather, the decision maker would attempt a "best-effort" approach avoiding to the extent possible load shedding and wind curtailment to maintain the system operational under any postulated contingency within the considered set. We again resort to a DC-SCOPF to model such "best-effort" approach. More specifically, we consider the preventive and/or corrective re-dispatch of each generating unit as the available actions of first priority and, in our objective, take into account the respective marginal re-dispatch costs. Further, we model pre- and post-contingency load shedding and wind curtailment and treat these actions as options of last resort by means of appropriately high penalty cost coefficients in the objective function of the resulting optimization problem. The detailed mathematical formulation of this problem, notably including ramping constraints between any preventive re-dispatch action and the day-ahead dispatch for the forthcoming period for every generating unit is available to the reader in Appendix A-C.

[4]Notice that, at this stage we exclude from consideration the outages of the single lines feeding nodes 207 and 307; such outages would lead to the islanding of these nodes.

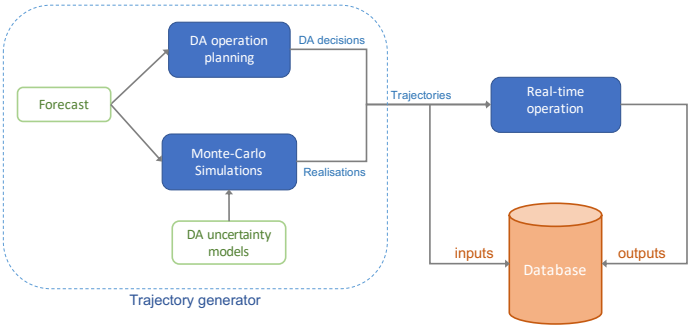


Figure 1. Methodology applied to the database used to learn the proxies y_p.

4) Operational cost assessment: Finally, given the decisions for the solution of this real-time SCOPF problem, we evaluate the respective hourly costs of operation. At this stage we adopt a penalty for wind-curtailment of 300€/MWh for any wind power generator and, consider an identical value of lost load for each load demand[5], which is an average of the coefficients published in [8], assuming a 1 hour supply interruption duration. Using this process, we build a database of 2400 trajectories. Figure 1 schematically summarises the database generation.

B. Machine learning settings and predictors

Once the database is generated, we can use it to learn the proxies. We study in particular two classes of regression predictors: Extremely Randomized Trees (which is a random forest method) with 1000 trees [9] and Neural Networks [10]. In the rest of the paper, we call them respectively ET and NN. The NN used in this paper is a multi-layer perceptron with a ReLU activation function. Both ET and NN are two well-known non-parametric methods to solve regression problems and are both non-linear. Furthermore they have complementary characteristics. A value predicted by the NN predictor is not bounded by the values seen in the training database, contrarily to the ET predictor. On the other hand, the ET predictor is smoother. Furthermore, with a random forest algorithm such as ET, we can exploit a by-product of the training models: the feature importances. This allows to rank the features by importance score to predict the target output and thus have a better understanding of the system studied.

1) Designing the predictors: In order to build the predictors, we divide our dataset into two sets: a learning and a test set. The proxies are built with the learning set and then used to predict the target output over the test set. The comparison between the true value of the test set and the prediction of the proxy allows us to assess how the proxy performs on unseen data drawn from the same distribution.

[5]In this study, we are only interested in the amount of load shedding and so we do not want to discriminate by the value of lost load in case load shedding is necessary.

Both methods depend on meta-parameters that can be tuned to improve the performance of the predictors. In order to find the best meta-parameters but still avoid overfitting, we use 5-fold cross-validation. We first divide the learning set into five folds. Then, for each possible combination of the meta-parameters, we use four of the five folds to learn a model and the left-out fold to assess the performance by computing a score (in this paper, we use the R^2-score, as in [2]). If we repeat the process five times, leaving out a different fold each time, we obtain five scores that we average to obtain a generalization score. We keep the combination of meta-parameters leading to the best generalization score. The tested meta-parameters are enumerated in Appendix C.

2) Choice of the setting for machine learning: In order to learn the proxies, we have investigated two different settings of applying machine learning to a dataset of trajectories solved with the SCOPF model.

Setting 1 was to use a dataset of (say k) trajectories, each one described by a set of features describing the uncertainty realisations for each of the 24 hours (as well as the generation schedule decided the day ahead), and as output the sum of hourly costs along that trajectory.

Setting 2, on the other hand, consisted of splitting each trajectory into 24 hourly snapshots, inferring on the basis of these $k\times 24$ hours a proxy of the hourly costs, and predicting then for a certain trajectory the total cost by the sum of its 24 hourly proxy predictions.

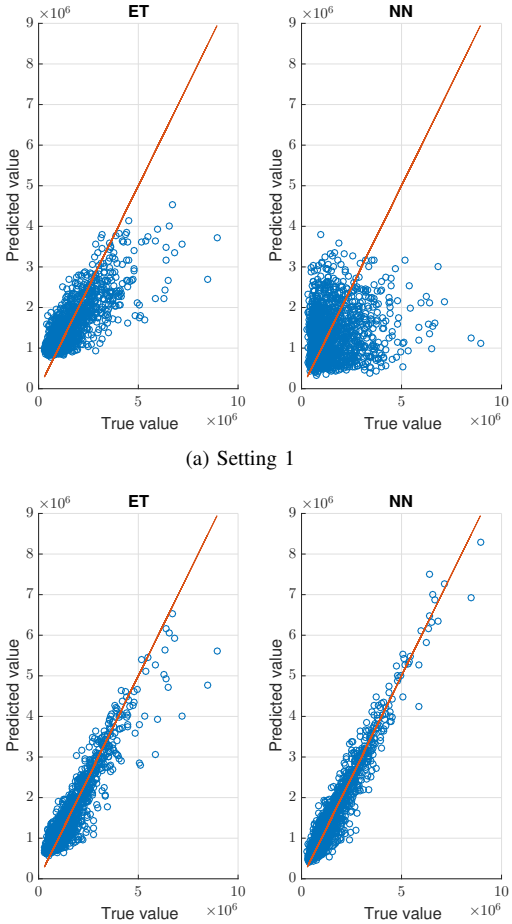
In a preliminary study, we applied these two approaches with both the ET and NN predictors, different cost terms and different training sample sizes k. By comparing the performances of these two settings on independent test samples, in terms of the overall accuracy of their predictions, we made two observations, namely i.) Setting 2 very clearly outperforms Setting 1, in all cases, and ii.) the NN predictor is often significantly more accurate than the ET predictor, but it needs a more careful tuning of its meta-parameters. As an example, Figure 2 shows four scatter plots comparing one of the terms of the real-time operating costs (the total preventive cost, in euro) against its predicted value over the test set for both settings and learning algorithms in the case of $k = 850$ learning trajectories (each one of 24 hours).

Hence in the rest of this paper we only report our results obtained with Setting 2, and we provide comparisons of the ET and NN predictors, when of interest.

C. Results

In our study we have investigated different terms of the real-time operating costs, namely total preventive control cost (*i.e.* the sum of preventive generation re-dispatch, load shedding and wind curtailment costs), expected corrective control cost (also composed, per contingency, of generation re-dispatch, load shedding and wind curtailment cost sub-components), preventive load-shedding and wind-curtailment costs.

1) Comparison of the approaches: To compare the different approaches, we first apply them to estimate the 'Total preventive control cost' (C^p_{tot}).



(a) Setting 1

(b) Setting 2

Figure 2. Scatter plots showing true vs predicted values for the total preventive cost over the test set, in case of (a) direct prediction of the trajectory cost and (b) sum of the 24 hourly proxy predictions. $k = 850$.

a) Computational budget: To start, Figure 3 shows the convergence of the crude Monte-Carlo approach when applied to the estimation of the expected value $\mu_{C^p_{tot}}$ of the total preventive control cost, as a function of the sample size n. With a total number $n = 2400$ of sampled scenarios, the standard error of this crude Monte Carlo approach is slightly above 2×10^4, i.e. about 1.4% of the estimated value of $\mu_{C^p_{tot}} \approx 1.44 \times 10^6$.

Next, we apply the control variates approach while using the first 850 scenarios to build our proxies (with Setting 2), and both ET and NN predictors. Figure 4 shows that in both cases this approach leads to a reduction of the standard error of about a factor 2, the NN predictor being more accurate than the ET predictor. With a total number of trajectories of $n = 2400$ $(k = 850, m = 1550)$, the standard errors of the control variates approaches is smaller than 1×10^4 (0.7% in relative terms).

Thus, while all estimators seem to converge to the same final value, with the crude Monte Carlo approach, we would need about 10,000 trajectories (i.e. about 240,000 SCOPF computations at the hourly basis) to reach the same level of

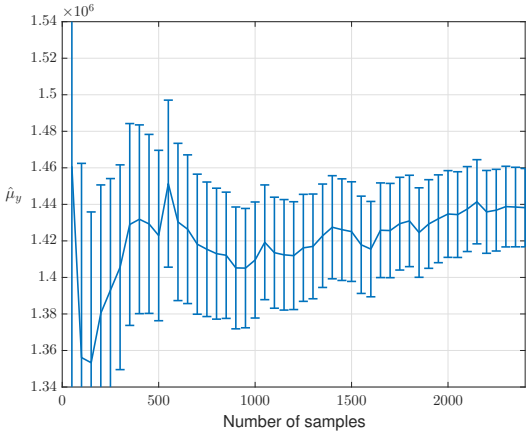


Figure 3. Convergence of the crude Monte-Carlo approach for C^p_{tot} with a growing number n (up to 2400) of scenario samples (error bars show the $\pm\hat{\sigma}/\sqrt{n}$ interval, where $\hat{\sigma}$ is the sample estimate of the standard deviation of C^p_{tot})

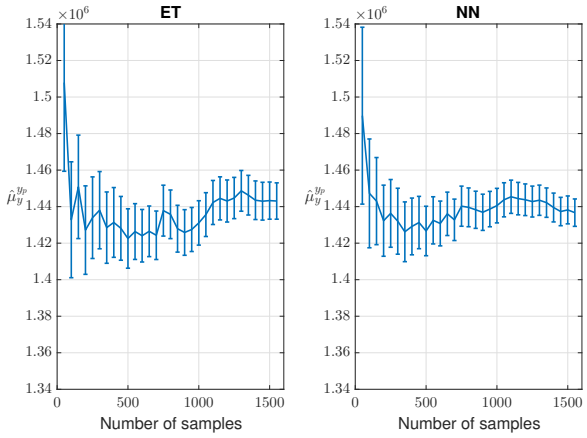


Figure 4. Convergence of the control variate approach applied to C^p_{tot} for a learning set size of $k = 850$, and up to $n - k = 1550$ additional samples

accuracy than the control variates approaches using only about 60,000 SCOPF computations.

In order to construct the above estimators, we have used the crude Monte Carlo approach to estimate μ_{yp} of the proxy predictions on a sample of 20,000 trajectories (without any SCOPF computation, of course). Figure 5 shows, for the NN predictor, how this side computation converges with the number of (side) samples. We also see from this figure that using this proxy alone, even with a much larger number of additional side scenario samples (say about 10,000), would lead to an estimator that would be quite biased (in the present case underestimating the actual value of the expected total preventive cost around 1.42×10^6) while also having a higher variance than the two estimators using the control variates approach. Notice that with the ET method we obtain a quite similar curve (not shown), with the main difference that in this case the bias is positive and even a bit larger (with a mean μ_{yp} converging towards 1.47×10^6).

From these results, we observe that for a same budget of 2400 scenarios solved with the SCOPF computations, we can significantly improve the crude Monte Carlo approach by

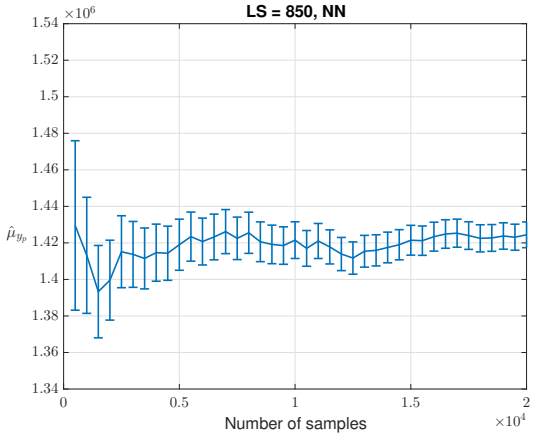


Figure 5. Estimation by crude Monte Carlo of $\hat{\mu}_{yp}(NN)$ from side samples (here the proxy is based on NN and was built using 850 training trajectories).

using part of the sample to build a control variate by machine learning, and the rest to reduce the bias of these proxies in a systematic way.

b) Size of the learning sample: Of course, the budget of trajectories could possibly be used in different ways. To investigate this aspect, we have made further analyses. Table I reports numerical results. The first line gives the average and its standard error as obtained with the crude Monte Carlo approach for a total budget of 2400 scenarios solved with the SCOPF model, each one with 24 hourly time steps; the next lines show the results of the proposed approach, for different ways of splitting this budget into learning sample (k), the $2400-k$ being used for the control variates approach. For each value of k and for both ET and NN predictors, we indicate the value of their mean ($\hat{\mu}_{yp}$) estimated from 20,000 side samples (and its standard error), and the mean and standard error of using it in the control variates approach. We observe from the values in this table that, in spite of the biased values of $\hat{\mu}_{yp}$ and for all settings, the control variates approach is yielding a non biased estimator with a factor two improvement of the standard error with respect to the crude Monte Carlo approach. We also see that the most accurate setting seems to be the NN approach with $k = 750$ (with a standard error of about 8×10^3), but the variation of accuracy with k and the predictor used is actually not very strong, the worst setting (ET with $k = 1000$) and the best one (NN with $k = 750$) yielding a difference in standard error of less than 20% (both are highlighted in bold in the table).

c) The Stacked-Monte Carlo approach: In order to conclude our analysis of variance reduction methods, let us report some preliminary experiments using the SMC approach. The results of Figure 6 were obtained in the following way: for a given number of samples $n \in [250; 2400]$, we used the SMC method with the NN predictor and $V = 10$ folds. We can observe from this graph that with a budget of only $n = 1000$ trajectories, we already obtain a non biased estimate of $\mu_{C^p_{tot}}$ with a standard error comparable to those obtained with the above control variate approaches when they were exploiting a budget of more than 2000 trajectories.

TABLE I

ACCURACIES OF DIFFERENT SETTINGS OF THE CONTROL-VARIATE APPROACH TO ESTIMATE THE EXPECTED VALUE OF C^p_{tot}

Method	Mean	Std Err
CMC - $n = 2400$	1.438e+06	21.45e+03
$\hat{\mu}_{yp}(ET)$ - $k = 250$	1.524e+06	5.87e+03
$\hat{\mu}_{yp}(NN)$ - $k = 250$	1.416e+06	7.14e+03
MC with control variate (ET) - $k = 250$	1.440e+06	9.57e+03
MC with control variate (NN) - $k = 250$	1.436e+06	9.03e+03
$\hat{\mu}_{yp}(ET)$ - $k = 500$	1.499e+06	6.23e+03
$\hat{\mu}_{yp}(NN)$ - $k = 500$	1.426e+06	7.13e+03
MC with control variate (ET) - $k = 500$	1.437e+06	8.93e+03
MC with control variate (NN) - $k = 500$	1.431e+06	8.97e+03
$\hat{\mu}_{yp}(ET)$ - $k = 750$	1.478e+06	6.33e+03
$\hat{\mu}_{yp}(NN)$ -$k = 750$	1.428e+06	7.09e+03
MC with control variate (ET) - $k = 750$	1.440e+06	9.68e+03
MC with control variate (NN) - $k = 750$	1.434e+06	**8.07e+03**
$\hat{\mu}_{yp}(ET)$- $k = 1000$	1.474e+06	6.41e+03
$\hat{\mu}_{yp}(NN)$ -$k = 1000$	1.419e+06	7.01e+03
MC with control variate (ET) - $k = 1000$	1.441e+06	**9.85e+03**
MC with control variate (NN) - $k = 1000$	1.431e+06	8.19e+03

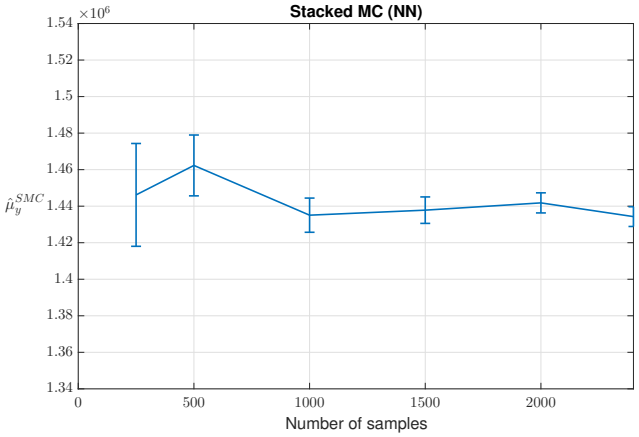


Figure 6. Convergence of the expected value of C^p_{tot} by the Stacked MC method, with 10 folds and NN as learning algorithm

2) Application to the estimation of other terms of the cost function: For the sake of completeness we also briefly report here on results obtained from the application of these approaches to the sub-components of the real-time cost function. More specifically, we present results on estimating the cost of preventive load shedding, preventive wind curtailment as well as expected corrective cost (that is, sum of expected corrective generation re-dispatch, load shedding and wind curtailment) respectively.

For each one of these sub-components, tables II – IV present the performance of the crude Monte Carlo approach (first row), with the control variate (third row) while using 850 trajectories to construct proxies via the NN predictor, which as reported earlier was found to out-perform the ET predictor. The middle row per sub-component presents the mean value and standard error of the NN predictor, once again estimated using 20,000 side-samples.

Tables II – IV verify in general the applicability of the proposed approach in estimating the various sub-components of the real-time cost function. It is of interest to comment here that such sub-components are also indicative of the type of problems to be anticipated in real-time operation. For instance, increased preventive load shedding & wind curtailment costs

TABLE II

ACCURACIES OF ESTIMATING THE EXPECTED VALUE OF PREVENTIVE LOAD SHEDDING COST – C^{LS}

Method	Mean	Std Err
CMC - $n = 2400$	8.6157e+05	2.1374e+04
$\hat{\mu}_{yp}(NN)$ - $k = 850$	8.7661e+05	7.1728e+03
MC with control variate (NN) - $k = 850$	8.5968e+05	7.6484e+03

TABLE III

ACCURACIES OF ESTIMATING THE EXPECTED VALUE OF PREVENTIVE WIND CURTAILMENT COST – C^{WC}

Method	Mean	Std Err
CMC - $n = 2400$	4.5849e+05	2.9559e+03
$\hat{\mu}_{yp}(NN)$ - $k = 850$	4.5644e+05	1.0053e+03
MC with control variate (NN) - $k = 850$	4.5714e+05	1.1766e+03

TABLE IV

ACCURACIES OF ESTIMATING THE EXPECTED VALUE OF EXPECTED TOTAL CORRECTIVE CONTROL COST – $\hat{C}^{CC}_{tot}$

Method	Mean	Std Err
CMC - $n = 2400$	2.8057e+03	4.3067
$\hat{\mu}_{yp}(NN)$ - $k = 850$	2.8037e+03	1.4425
MC with control variate (NN) - $k = 850$	2.8158e+03	2.9275

are indicative of the fact that the network may be inadequate to securely accommodate the range of potential wind power and demand injections, that is a lack in transmission capacity and operational flexibility. These indicators can thus be exploited to point towards the type of necessary operational planning decisions. Likewise, increased expected corrective control costs may indicate that exogenous factors (*e.g.*, weather conditions) and/or the system loading conditions increase the likelihood and/or potential impact of contingencies. From a planning perspective such finding can be exploited, for instance, by considering to make more preventive real-time flexibility resources available in order to reduce reliance on corrective control, taking into account that in practice it may turn out not to perform as expected.

Last but not least, these preliminary results on the sub-components establish the potential for further investigation on estimating the broad range of indicators that can be relevant to describe the operability of the system in real-time, such as load shedding and wind curtailment costs referring to specific loads and wind generators of interest, the achievabililty of the considered reliability criterion *etc.*.

IV. RELATED WORKS AND CONTRIBUTION

In this section, we position our contribution with respect to the literature of related works.

On the one hand, we refer the reader to the text-book [3] for an up to date introduction to Monte Carlo methods, and the huge body of variance reduction techniques that have been proposed over the years in this context. We also refer to [10] for an explanation of the prominent role of the bias-variance tradeoff in the design of modern machine learning algorithms. In direct relation to the methods developed in this paper, we refer to [4], [11] which are studying from a theoretical point of view the use of machine learning to build control variates for the Monte Carlo approach. In particular, [11] shows that by using a suitable class of non-parametric regression methods

(such as the ET and NN predictors used in our work), "Super-root-n" convergence may be achieved for a very large class of complex Monte Carlo integration problems.

On the other hand, the general idea of using machine learning in order to speed up power systems security and reliability assessment dates back to the 1970s (see e.g. the bibliography of [12]). Since its origins, this idea has continuously attracted the interest of power system engineers, with an intensification in the recent years, due to the significant progresses in machine learning and the strong increase in computing power and data gathering possibilities. A good bibliography review about all these recent works is however missing at this stage, and beyond the scope of this paper.

The more specific idea of using machine learning to build proxies of shorter-term decision-making contexts to be used when solving longer-term reliability assessment problems has been proposed and studied only recently [13], [1], [14], [2]. Within this context, the method presented in this paper, using machine learning to build control variates to speed up the Monte Carlo approach, is to our best knowledge entirely novel.

V. Conclusion and further research

In this paper we have explored the use of machine learning in order to speed up the Monte Carlo simulation approach in the context of uncertainty aware reliability assessment in operation planning.

The Monte Carlo simulation approach has two nice features in this context: i) it lends itself almost trivially to massive parallel computing architectures, ii) it is free of strong assumptions about the uncertainty and real-time operation models and hence very generally applicable. The crude Monte Carlo approach is nevertheless highly compute-intensive, and in order to scale it to real-life use, speeding it up is therefore highly desirable.

The approach investigated in this paper has solid theoretical guarantees: it is un-biased and may yield 'Super-root-n" convergence. On the basis of a systematic case study on the three-area RTS96 benchmark, we found that when estimating the expected value of various costs terms of next-day real-time operation it allows one to reduce the standard error of the crude Monte Carlo approach by a factor of about 3 to 4, while using the same number of sampled trajectories of next-day operation conditions. In computational terms, this means a speed-up of a factor 9 to 16, for a given target accuracy.

There are many possible directions of future research to broaden the potential of the proposed approach. Among them we mention the following ones:

- How to adapt the approach in order to estimate the probabilities of extreme situations that could occur in the next day, e.g. leading to the inoperability of the system?
- How to enhance the approach to rank different possible day-ahead decisions, rather than just estimating the cost incurred by one of them, in order to more directly support a planner in his decision-making?
- How to adapt the approach to estimate the gradient of the expected value of the different cost terms or of extreme situations' probabilities with respect to the calibration of day-ahead decisions, in order to provide further hints for optimal day-ahead decision making?

In addition, there is also room for further improving the proposed approach by exploiting various existing machine learning algorithms (such as deep learning methods), and by finding out other settings than the two that we have investigated in this paper.

References

[1] GARPUR Consortium, "D2.2 - guidelines for implementing the new reliability assessment and optimization methodology," 2016, available at: http://www.garpur-project.eu/deliverables.

[2] L. Duchesne, E. Karangelos, and L. Wehenkel, "Machine learning of real-time power systems reliability management response," *PowerTech Manchester 2017 Proceedings*, 2017.

[3] R. Y. Rubinstein and D. P. Kroese, *Simulation and the Monte Carlo method*. John Wiley & Sons, 2016, vol. 10.

[4] B. D. Tracey and D. H. Wolpert, "Reducing the error of Monte Carlo algorithms by learning control variates," *arXiv preprint arXiv:1606.02261*, 2016.

[5] P. Wong, P. Albrecht, R. Allan, R. Billinton, Q. Chen, C. Fong, S. Haddad, W. Li, R. Mukerji, D. Patton *et al.*, "The ieee reliability test system-1996. a report prepared by the reliability test system task force of the application of probability methods subcommittee," *Power Systems, IEEE Transactions on*, vol. 14, no. 3, pp. 1010–1020, 1999.

[6] H. Pandzic, Y. Dvorkin, T. Qiu, Y. Wang, and D. Kirschen, "Unit Commitment under Uncertainty - GAMS Models," Library of the Renewable Energy Analysis Lab (REAL), University of Washington, Seattle, USA, [Online]. Available at: http://www.ee.washington.edu/research/real/gams_code.html.

[7] L. Duchesne, "Machine learning of proxies for power systems reliability management," Master's thesis, Université de Liège, Liège, Belgique, 2016.

[8] M. Fotuhi-Firuzabad and R. Billinton, "Impact of load management on composite system reliability evaluation short-term operating benefits," *IEEE Transactions on power systems*, vol. 15, no. 2, pp. 858–864, 2000.

[9] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine Learning*, vol. 63, no. 1, pp. 3–42, 2006.

[10] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. Springer Series in Statistics, 2009.

[11] C. J. Oates, M. Girolami, and N. Chopin, "Control functionals for Monte Carlo integration," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 79, no. 3, pp. 695–718, 2017.

[12] L. A. Wehenkel, *Automatic learning techniques in power systems*. Springer Science & Business Media, 2012.

[13] L. Wehenkel, "How to combine observational data sources with first principles of physics to build stable and transportable models for power system design and control?" Plenary presentation at "Analytic Research Foundations for the Next-Generation Grid". A workshop of the National Research Council of the National Academies, https://books.google.be/books?isbn=0309378591, 2015.

[14] G. Dalal, E. Gilboa, S. Mannor, and L. Wehenkel, "Unit commitment using nearest neighbor as a short-term proxy," *arXiv preprint arXiv:1611.10215*, 2016.

[15] A. J. Wood and B. F. Wollenberg, *Power generation, operation, and control*. John Wiley & Sons, 2012.

[16] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[17] Scikit-learn developers, "Model evaluation: quantifying the quality of predictions," [Online], Available at http://scikit-learn.org/stable/modules/model_evaluation.html, accessed on 2016-05-11.

APPENDIX A
DAY-AHEAD AND REAL-TIME OPERATION MODELS

This section details the implementation of the day-ahead and real-time operation simulators. It begins with introducing the notations used in the mathematical models, then it presents the day-ahead decision-making program and finally it describes the real-time SCOPF.

A. Notations

a) Indices:

c	Index of contingencies
d	Index of demands
g	Index of generating units
k	Index of piece-wise linear dispatchable generation cost curve segments
l	Index of transmission elements (lines, cables and transformers)
n	Index of nodes
t	Index of hours in a day
w	Index of wind power generators

b) Sets:

$\mathcal{C}$	Set of contingencies
$\mathcal{D}$	Set of demands
$\mathcal{D}_n$	Subset of demands connected at node n
$\mathcal{G}$	Set of dispatchable units
$\mathcal{K}$	Set of piece-wise linear dispatchable generation cost curve segments
$\mathcal{L}$	Set of transmission elements (lines, cables and transformers)
$\mathcal{N}$	Set of nodes
$\mathcal{W}$	Set of wind power generators
$\mathcal{W}_n$	Subset of wind power generators connected at node n

c) Parameters:

$P_{d,t}^{forecast}$	Forecast of load active power of demand d at time t
$P_{w,t}^{forecast}$	Forecast of generation of wind power generator w at time t
$P_{d,t}^{RT}$	Realisation of load active power of demand d at time t
$P_{w,t}^{RT}$	Realisation of generation of wind power generator w at time t
on_g^{init}	Initial status of generating unit g at the beginning of the day-ahead decision-making (1 if started up, 0 otherwise)
$t_g^{up,init}$	Minimum number of time periods generating unit g must stay up at the beginning of the considered day
$t_g^{dn,init}$	Minimum number of time periods generating unit g must stay down at the beginning of the considered day
$t_g^{up,min}$	Minimum number of time periods generating unit g must stay up once started up
$t_g^{dn,min}$	Minimum number of time periods generating unit g must stay down once shut down
c_g	Redispatch marginal cost of generating unit g
c_g^0	Start-up cost of generating unit g
$c_{g,k}^{inc}$	Marginal running cost of generating unit g at the segment k of its piece-wise linear curve
P_g^{max}	Capacity of generating unit g
P_g^{min}	Minimum stable output of generating unit g
ΔP_g^-	Ramp-down limit of generating unit g (for 60min)
ΔP_g^+	Ramp-up limit of generating unit g (for 60min)
$\Delta P_g^{-,c}$	Ramp-down limit of generating unit g in case of corrective actions (for 20min)
$\Delta P_g^{+,c}$	Ramp-up limit of generating unit g in case of corrective actions (for 20min)
$P_{g,k}^{inc,max}$	Maximum power output of generating unit g at the segment k of its piece-wise linear curve
v_d	Voll of demand d in €/MWh
p_w	Wind penalty for curtailment of wind power generator w in €/MWh
R^+	Minimum up spinning reserve required per hour for one area
R^-	Minimum down spinning reserve required per hour for one area
f_l^{max}	Long-term thermal rating of transmission element l
r_l	Ratio of the short-term thermal rating to the long-term thermal rating of transmission element l ($r_l \geq 1$)
X_l	Reactance of transmission element l
$\beta_{n,l}$	Element of the flow incidence matrix, taking a value of one if node n is the sending node of element l, a value of minus one if node n is the receiving node of element l, and a zero value otherwise.
$a_{l,c}$	Binary parameter taking a zero value if element l is unavailable under contingency c.

d) Variables:

$P_{g,t}^{DA}$	Dispatch of generating unit g at time t as per the day-ahead decision-making
$on_{g,t}^{DA}$	Binary variable representing the status of generating unit g as per the day-ahead decision-making (1 if started up, 0 otherwise)
$st_{g,t}^{up}$	Binary variable indicating when generating unit g is started-up (value 1 when started up, 0 otherwise)
$st_{g,t}^{dn}$	Binary variable indicating when generating unit g is shut down (value 1 when shut down, 0 otherwise)
$WC_{w,t}^{DA}$	Provisional curtailment of wind power generator w at hour t in day-ahead
$R_{g,t}^+$	Upward redispatch flexibility provided by generating unit g at time t in day-ahead
$R_{g,t}^-$	Downward redispatch flexibility provided by generating unit g at time t in day-ahead

$f_{l,t}^{DA}$	Power flowing through transmission element l at time t under the pre-contingency state in day-ahead
$f_{l,t,c}^{DAST}$	Power flowing through transmission element l at time t following contingency c in day-ahead
$\theta_{n,t}^{DA}$	Voltage angle at node n under the pre-contingency state in day-ahead
$\theta_{l,t,c}^{DAST}$	Voltage angle at node n following contingency c in day-ahead.
$^{+}P_{g,t}^{RTp}$	Preventive ramp-up of generator g in real-time at hour t
$^{-}P_{g,t}^{RTp}$	Preventive ramp-down of generator g in real-time at hour t
$LS_{d,t}^{RTp}$	Preventive load shedding of demand d in real-time at hour t
$WC_{w,t}^{RT^p}$	Preventive wind curtailment of wind power generator w in real-time at hour t
$^{+}P_{g,t,c}^{RTc}$	Corrective ramp-up of generator g in real-time at hour t following contingency c
$^{-}P_{g,t,c}^{RTc}$	Corrective ramp-down of generator g in real-time at hour t following contingency c
$LS_{d,t,c}^{RTc}$	Corrective load shedding of demand d in real-time at hour t following contingency c
$WC_{w,t,c}^{RTc}$	Corrective wind curtailment of wind power generator w in real-time at hour t following contingency c
$f_{l,t}^{p}$	Power flowing through transmission element l under the pre-contingency state
$f_{l,t,c}^{ST}$	Power flowing through transmission element l following contingency c and prior to the application of corrective control.
$f_{l,t,c}^{c}$	Power flowing through transmission element l following contingency c and the successful application of corrective control.
$\theta_{n,t}^{p}$	Voltage angle at node n under the pre-contingency state
$\theta_{n,t,c}^{ST}$	Voltage angle at node n following contingency c and prior to the application of corrective control.
$\theta_{n,t,c}^{c}$	Voltage angle at node n following contingency c and the successful application of corrective control.

All the variables are continuous, except for $on_{g,t}$, $st_{g,t}^{dn}$ and $st_{g,t}^{up}$ which are binary variables. Powers flowing through transmission elements and voltage angles are continuous in $\mathbb{R}$ and the remaining variables are positive.

B. Day-ahead decision-making

We simulate day-ahead decision-making with a multi-period SCOPF in order to commit and dispatch the generating units of the system and also determine the provisional wind curtailment. We use the DC approximation [15] and consider as reliability criterion the N-1 criterion for transmission elements only.

The objective function minimizes generation cost as well as provisional wind curtailment:

$$\text{minimize} \quad \sum_{t=1}^{24}\left(\sum_{g\in\mathcal{G}}\left(c_g^0 * st_{g,t}^{up} + \sum_{k\in\mathcal{K}} c_{g,k}^{inc} * P_{g,k,t}^{inc}\right) + \sum_{w\in\mathcal{W}} p_w * WC_{w,t}^{DA}\right) \tag{8}$$

The first set of constraints (9-19) of the day-ahead program concerns the minimum time a generating unit must stay up or down, either at the beginning of the day or during the day.

For $t = 1$, $\forall g \in \mathcal{G}$:

$$st_{g,t}^{up} - st_{g,t}^{dn} = on_{g,t}^{DA} - on_g^{init} \tag{9}$$

$$st_{g,t}^{up} + st_{g,t}^{dn} \leq 1 \tag{10}$$

$\forall t = 2, ..., 24$, $\forall g \in \mathcal{G}$:

$$st_{g,t}^{up} - st_{g,t}^{dn} = on_{g,t}^{DA} - on_{g,t-1}^{DA} \tag{11}$$

$$st_{g,t}^{up} + st_{g,t}^{dn} \leq 1 \tag{12}$$

(13)

$\forall g \in \mathcal{G}$:

$$\sum_{t'=1}^{t_g^{up,init}} \left(1 - on_{g,t'}^{DA}\right) = 0 \tag{14}$$

$$\sum_{t'=1}^{t_g^{dn,init}} on_{g,t'}^{DA} = 0 \tag{15}$$

(16)

$\forall g \in \mathcal{G}$, $\forall t = 1, ..., \left(24 - t_g^{up,min}\right)$:

$$\sum_{t'=t}^{t+t_g^{up,min}} on_{g,t'}^{DA} \geq st_{g,t}^{up} \cdot t_g^{up,min} \tag{17}$$

(18)

$\forall g \in \mathcal{G}$, $\forall t = 1, ..., \left(24 - t_g^{dn,min}\right)$:

$$\sum_{t'=t}^{t+t_g^{dn,min}} \left(1 - on_{g,t'}^{DA}\right) \geq st_{g,t}^{dn} \cdot t_g^{dn,min} \tag{19}$$

(20)

The following set of constraints limits the power output of each generating unit between its minimum stable output and its maximum capacity, computes the upward and downward redispatch flexibility of each generator and also imposes ramping constraints to go from one committed dispatch to the one of the next period in one hour:

$\forall g \in \mathcal{G}$, $\forall t = 1, ..., 24$:

$$-P_{g,t}^{DA} + R_{g,t}^{-} \leq -P_g^{min} \cdot on_{g,t}^{DA} \tag{21}$$

$$P_{g,t}^{DA} + R_{g,t}^{+} \leq P_g^{max} \cdot on_{g,t}^{DA} \tag{22}$$

$$P_{g,t+1}^{DA} - P_{g,t}^{DA} \leq \Delta P_g^{+} \cdot on_{g,t}^{DA} + P_g^{max}(1 - on_{g,t}^{DA}) \tag{23}$$

$$-\left(P_{g,t+1}^{DA} - P_{g,t}^{DA}\right) \leq \Delta P_g^{-} \cdot on_{g,t}^{DA} + P_g^{max}(1 - on_{g,t+1}^{DA}) \tag{24}$$

We assume a piece-wise linear cost function of $|\mathcal{K}|$ segments for the marginal running cost of a generating unit g, which gives eq. (25) and (27).
$\forall g \in \mathcal{G}$, $\forall k \in \mathcal{K}$, $\forall t = 1, ..., 24$:

$$P^{inc}_{g,k,t} \leq on^{DA}_{g,t} \cdot \ P^{inc,max}_{g,k} \tag{25}$$

$$\tag{26}$$

$\forall g \in \mathcal{G}$, $\forall t = 1, ..., 24$:

$$P^{DA}_{g,t} = \sum_{k=1}^{K} P^{inc}_{g,k,t} \tag{27}$$

Equation (28) represents the balancing of the system and equations (30)-(32) the transmission constraints in case of the DC approximation.

$\forall t = 1, ..., 24$, $\forall n \in \mathcal{N}$:

$$\begin{aligned} \sum_{w \in \mathcal{W}_n} (P^{forecast}_{w,t} - WC^{DA}_{w,t}) + \sum_{g \in \mathcal{G}_n} P^{DA}_{g,t} \\ - \sum_{l \in \mathcal{L}} \beta_{n,l} \cdot f^{DA}_{l,t} = \sum_{d \in \mathcal{D}_n} P^{forecast}_{d,t} \end{aligned} \tag{28}$$

$\forall t = 1, ..., 24$, $\forall w \in \mathcal{W}$:

$$0 \leq WC^{DA}_{w,t} \leq P^{forecast}_{w,t} \tag{29}$$

$\forall t = 1, ..., 24$, $\forall l \in \mathcal{L}$:

$$f^{DA}_{l,t} = \frac{1}{X_l} \sum_{n \in \mathcal{N}} \beta_{n,l} \cdot \theta^{DA}_{l,t} \tag{30}$$

$$f^{DA}_{l,t} \leq f^{max}_{l} \tag{31}$$

$$-f^{DA}_{l,t} \leq f^{max}_{l} \tag{32}$$

Equations (33)-(36) force the system to still be secure in the case of the loss of one transmission element.

$\forall t = 1, ..., 24$, $\forall c \in \mathcal{C}, \forall n \in \mathcal{N}$:

$$\begin{aligned} \sum_{w \in \mathcal{W}_n} (P^{forecast}_{w,t} - WC^{DA}_{w,t}) + \sum_{g \in \mathcal{G}_n} P^{DA}_{g,t} \\ - \sum_{l \in \mathcal{L}} \beta_{n,l} \cdot f^{DAST}_{l,t,c} = \sum_{d \in \mathcal{D}_n} P^{forecast}_{d,t} \end{aligned} \tag{33}$$

$\forall t = 1, ..., 24$, $\forall c \in \mathcal{C}, \forall l \in \mathcal{L}$:

$$f^{DAST}_{l,t,c} = a_{l,c} \cdot \frac{1}{X_l} \sum_{n \in \mathcal{N}} \beta_{n,l} \cdot \theta^{DAST}_{l,t,c} \tag{34}$$

$$f^{DAST}_{l,t,c} \leq a_{l,c} \cdot f^{max}_{l} \tag{35}$$

$$-f^{DAST}_{l,t,c} \leq a_{l,c} \cdot f^{max}_{l} \tag{36}$$

Equations (37)-(42) determine the minimum size of the up and down spinning reserves per area in the system (in this work, we have three areas and the same spinning reserve requirements per area).

$\forall t = 1, ..., 24$, $\forall g = \in \mathcal{G}$

$$\sum_{g \in \mathcal{G}_{area1}} R^{+}_{g,t} \geq R^{+} \tag{37}$$

$$\sum_{g \in \mathcal{G}_{area1}} R^{-}_{g,t} \geq R^{-} \tag{38}$$

$$\sum_{g \in \mathcal{G}_{area2}} R^{+}_{g,t} \geq R^{+} \tag{39}$$

$$\sum_{g \in \mathcal{G}_{area2}} R^{-}_{g,t} \geq R^{-} \tag{40}$$

$$\sum_{g \in \mathcal{G}_{area3}} R^{+}_{g,t} \geq R^{+} \tag{41}$$

$$\sum_{g \in \mathcal{G}_{area3}} R^{-}_{g,t} \geq R^{-} \tag{42}$$

C. Real-time operation

In order to simulate real-time operation along a system trajectory, we solve sequentially the 24 hourly steps of the trajectory. That is we solve 24 single period problems corresponding to the 24 hours of one day.

We model real-time operation with a SCOPF problem with the N-1 reliability criterion, again considering only transmission elements. We consider preventive (pre-contingency) as well as corrective (post-contingency) actions and we do not forget the intermediate state after the occurrence of a contingency but before any corrective action can be applied, that we call short-term post-contingency state.

Note that continuous variables from the day-ahead decision-making program are parameters for this problem.

1) Objective function:

The objective function (43) minimizes the redispatch cost (upward and downward) as well as load shedding and wind curtailment, both in preventive and corrective modes. The value of lost load and wind penalty should be such that load shedding and wind curtailment are used only where no other solution exists. In order to favour corrective actions over preventive ones, we multiply the total preventive cost by a large factor M.

minimize

$$\begin{aligned} M * \left(\sum_{g \in \mathcal{G}} c_g \left({}^{+}P^{RTp}_{g,t} + {}^{-}P^{RTp}_{g,t} \right) + \sum_{d \in \mathcal{D}} v_d * LS^{RTp}_{d,t} + \sum_{w \in \mathcal{W}} p_w * WC^{RTp}_{w,t} \right) \\ + \sum_{c \in \mathcal{C}} \left(\sum_{d \in \mathcal{D}} v_d * LS^{RTc}_{d,t,c} + \sum_{w \in \mathcal{W}} p_w * WC^{RTc}_{w,t,c} + \sum_{g \in \mathcal{G}} c_g \left({}^{+}P^{RTc}_{g,t,c} + {}^{-}P^{RTc}_{g,t,c} \right) \right) \end{aligned} \tag{43}$$

2) Pre-contingency state:

The following equations determine the preventive actions.The possible redispatch of generating units is limited by maximum and minimum output power of generating units as well as by ramping constraints of one hour. Equations (48) and (49) also impose that with the re-dispatch of a unit g, it is still possible to go in one hour to the dispatch of the generating unit g at time $t + 1$ as per the day-ahead decision-making.

$\forall g \in \mathcal{G}$:

$$P_{g,t}^{DA} + ({}^{+}P_{g,t}^{RTp} -^{-} P_{g,t}^{RTp}) \geq P_g^{min} \cdot on_{g,t}^{DA} \tag{44}$$

$$P_{g,t}^{DA} + ({}^{+}P_{g,t}^{RTp} -^{-} P_{g,t}^{RTp}) \leq P_g^{max} \cdot on_{g,t}^{DA} \tag{45}$$

$${}^{+}P_{g,t}^{RTp} -^{-} P_{g,t}^{RTp} \leq \Delta P_g^{+} \tag{46}$$

$$-({}^{+}P_{g,t}^{RTp} -^{-} P_{g,t}^{RTp}) \leq \Delta P_g^{-} \tag{47}$$

$$P_{g,t+1}^{DA} - \left(P_{g,t}^{DA} + ({}^{+}P_{g,t}^{RTp} -^{-} P_{g,t}^{RTp}\right) \leq \Delta P_g^{+} \tag{48}$$

$$-\left(P_{g,t+1}^{DA} - \left(P_{g,t}^{DA} +^{+} P_{g,t}^{RTp} -^{-} P_{g,t}^{RTp}\right)\right) \leq \Delta P_g^{-} \tag{49}$$

The next constraints correspond to the classical DC approximation.

$\forall n \in \mathcal{N}$:

$$\begin{aligned} \sum_{w \in \mathcal{W}_n} (P_{w,t}^{RT} - WC_{w,t}^{DA} - WC_{w,t}^{RT^p}) \\ + \sum_{g \in \mathcal{G}_n} \left(P_{g,t}^{DA} + ({}^{+}P_{g,t}^{RTp} -^{-} P_{g,t}^{RTp})\right) \\ - \sum_{l \in \mathcal{L}} \beta_{n,l} \cdot f_{l,t}^{p} \\ = \sum_{d \in \mathcal{D}_n} (P_{d,t}^{RT} - LS_{d,t}^{RT^p}) \end{aligned} \tag{50}$$

$\forall l \in \mathcal{L}$:

$$f_{l,t}^{p} = \frac{1}{X_l} \sum_{n \in \mathcal{N}_n} \beta_{n,l} \cdot \theta_{l,t}^{p} \tag{51}$$

$$f_{l,t}^{p} \leq f_l^{max} \tag{52}$$

$$-f_{l,t}^{p} \leq f_l^{max} \tag{53}$$

Finally, we ensure that we do not shed more load and wind generation than what is possible:

$\forall d \in \mathcal{D}$:

$$0 \leq LS_{d,t}^{RT^p} \leq P_{d,t}^{RT} \tag{54}$$

$\forall w \in \mathcal{W}$:

$$0 \leq WC_{w,t}^{DA} + WC_{w,t}^{RT^p} \leq P_{w,t}^{RT} \tag{55}$$

3) Short-term post-contingency state:

In this stage, a contingency occurred but the operator has not reacted yet. Since we are in emergency state, the line thermal ratings correspond to the short-term ones.

$\forall c \in \mathcal{C}, \forall n \in \mathcal{N}$:

$$\begin{aligned} \sum_{w \in \mathcal{W}_n} (P_{w,t}^{RT} - WC_{w,t}^{DA} - WC_{w,t}^{RT^p}) \\ + \sum_{g \in \mathcal{G}_n} \left(P_{g,t}^{DA} + ({}^{+}P_{g,t}^{RTp} -^{-} P_{g,t}^{RTp})\right) \\ - \sum_{l \in \mathcal{L}} \beta_{n,l} \cdot f_{l,t,c}^{ST} \\ = \sum_{d \in \mathcal{D}_n} (P_{d,t}^{RT} - LS_{d,t}^{RT^p}) \end{aligned} \tag{56}$$

$\forall c \in \mathcal{C}, \forall l \in \mathcal{L}$:

$$f_{l,t,c}^{ST} = a_{l,c} \cdot \frac{1}{X_l} \sum_{n \in \mathcal{N}_n} \beta_{n,l} \cdot \theta_{l,t,c}^{ST} \tag{57}$$

$$f_{l,t,c}^{ST} \leq a_{l,c} \cdot r_l \cdot f_l^{max} \tag{58}$$

$$-f_{l,t,c}^{ST} \leq a_{l,c} \cdot r_l \cdot f_l^{max} \tag{59}$$

4) Corrective control:

Finally, corrective actions can be applied to keep the system secure.

$\forall c \in \mathcal{C}, \forall g \in \mathcal{G}$:

$$\begin{aligned} P_{g,t}^{DA} + ({}^{+}P_{g,t}^{RTp} -^{-} P_{g,t}^{RTp}) + ({}^{+}P_{g,t,c}^{RTc} -^{-} P_{g,t,c}^{RTc}) \\ \geq P_g^{min} \cdot on_{g,t}^{DA} \end{aligned} \tag{60}$$

$$\begin{aligned} P_{g,t}^{DA} + ({}^{+}P_{g,t}^{RTp} -^{-} P_{g,t}^{RTp}) + ({}^{+}P_{g,t,c}^{RTc} -^{-} P_{g,t,c}^{RTc}) \\ \leq P_g^{max} \cdot on_{g,t}^{DA} \end{aligned} \tag{61}$$

$${}^{+}P_{g,t,c}^{RTc} -^{-} P_{g,t,c}^{RTc} \leq \Delta P_g^{+,c} \tag{62}$$

$$-({}^{+}P_{g,t,c}^{RTc} -^{-} P_{g,t,c}^{RTc}) \leq \Delta P_g^{-,c} \tag{63}$$

$\forall c \in \mathcal{C}, \forall n \in \mathcal{N}$:

$$\begin{aligned} \sum_{w \in \mathcal{W}_n} (P_{w,t}^{RT} - WC_{w,t}^{DA} - WC_{w,t}^{RT^p} - WC_{w,t,c}^{RT^c}) \\ + \sum_{g \in \mathcal{G}_n} \left(P_{g,t}^{DA} + ({}^{+}P_{g,t}^{RTp} -^{-} P_{g,t}^{RTp}) + ({}^{+}P_{g,t,c}^{RTc} -^{-} P_{g,t,c}^{RTc})\right) \\ - \sum_{l \in \mathcal{L}} \beta_{n,l} \cdot f_{l,t,c}^{c} = \sum_{d \in \mathcal{D}_n} (P_{d,t}^{RT} - LS_{d,t}^{RT^p} - LS_{d,t,c}^{RT^c}) \end{aligned} \tag{64}$$

$\forall c \in \mathcal{C}, \forall d \in \mathcal{D}$:

$$0 \leq LS_{d,t}^{RT^p} + LS_{d,t,c}^{RT^c} \leq P_{d,t}^{RT} \tag{65}$$

$\forall c \in \mathcal{C}, \forall w \in \mathcal{W}$:

$$0 \leq WC_{w,t}^{DA} + WC_{w,t}^{RT^p} + WC_{w,t,c}^{RT^c} \leq P_{w,t}^{RT} \tag{66}$$

$\forall c \in \mathcal{C}, \forall l \in \mathcal{L}$:

$$f_{l,t,c}^{c} = a_{l,c} \cdot \frac{1}{X_l} \sum_{n \in \mathcal{N}_n} \beta_{n,l} \cdot \theta_{l,t}^{ST} \tag{67}$$

$$f_{l,t,c}^{c} \leq a_{l,c} \cdot f_l^{max} \tag{68}$$

$$-f_{l,t,c}^{c} \leq a_{l,c} \cdot f_l^{max} \tag{69}$$

APPENDIX B
CASE STUDY: THE IEEE-RTS96 - SHORT DESCRIPTION OF THE DATA

We test our methodology on the IEEE-RTS96 benchmark [5], where 19 windfarms have been added as in [6].

We consider in this case study the first day of the year, with a peak demand per area of 3135MW. Demand, generating units and line ratings data come from [5], while the forecast wind generation ('favorable') as well as the initial states of the generating units are borrowed from [6]. Note that line ratings have been reduced by a factor of 20%.

Concerning the reliability criterion, we use the N-1 criterion for transmission elements only (transmission lines, cables

and transformers) and thus we have 120 contingencies. It is important to note that for the real-time simulator we do not consider contingencies of lines 49 and 82 in order to avoid islanding of nodes 207 and 307. Therefore, we have only 118 contingencies in the real-time problem.

We choose as minimum up and down spinning reserve per area 300 MW. The wind penalty is 300€/MWh and the voll is an average of the coefficients from [8] converted in € and is thus equal to 4018.2€/MWh.

Concerning the objective function of the real-time simulator, we set M=150($> |\mathcal{C}|$) in order to be sure that corrective actions will always be favoured over preventive ones.

APPENDIX C
MACHINE LEARNING SETTINGS AND PREDICTORS

In this section, we begin by introducing briefly the learning algorithms used in the paper, then we describe the procedure used to train and test the models and we list the values of the meta-parameters tested to improve the performance of our models.

A. Two regression algorithms: extremely randomized trees and neural networks

We tested two types of predictors : extremely randomized trees (ET) [9] and artificial neural network (NN)[10].

The ET algorithm is a Random Forest algorithm [16]. It is an ensemble of regression trees where each tree is built with some randomness. The final prediction is the average of the predictions of each tree in the forest. This method has three meta-parameters: the number of trees in the forest, the number k of features selected randomly at each split and the minimum number of samples required to split a node n_{min}.

The artificial neural network we studied is a multi-layer perceptron with a ReLU activation function. We tuned the number of hidden layers and the number of neurons per layer.

B. Description of the learning procedure

In order to avoid overfitting, we divide the dataset randomly into two sets: a learning set and a test set. Each model is learnt with the learning set and its performances are assessed on the test set. It allows to see how well each model generalises on unseen data. The measure used to compare each model is the R^2-score (coefficient of determination), which is computed on the basis of N cases by [17]:

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2},$$

where y_i is the true output of case i, $\hat{y}_i$ is the predicted output, and $\bar{y}$ is the mean of the N true values. The best possible score is 1 and corresponds to a model that perfectly predicts all the target output values of the dataset used to estimate its value.

As said in the previous section, each model has some meta-parameters that we can tune in order to improve their performance. In order to select the best meta-parameters, we used a 5-fold cross validation. For the ET algorithm, we tested the following parameters: $k = 1, p/3, p/2, p$, where p is the total number of features and $n_{min} = 2, 4, 6, 8, 10, 20$. The number of trees was set to 1000, which is good trade-off between performance and time needed to train and predict. For the NN algorithm, we tried the following configurations: two or three hidden layers with 10, 50 or 100 neurons per layer. The best meta-parameters vary in function of the output we want to predict or the setting used.