

Machine Learning of Real-time Power Systems Reliability Management Response

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

Abstract—In this paper we study how supervised machine learning could be applied to build simplified models of real-time (RT) reliability management response to the realization of uncertainties. The final objective is to import these models into look-ahead operation planning under uncertainties. Our response models predict in particular the real-time reliability management costs and the resulting reliability level of the system. We tested our methodology on the IEEE-RTS96 benchmark. Among the supervised learning algorithms tested, extremely randomized trees, kernel ridge regression and neural networks appear to be the best methods for this application. Furthermore, by using feature “importances” computed by tree-based ensemble methods, we were able to extract the most relevant variables to predict the response of real-time reliability management, and thus obtain a better understanding of the system properties.

Index Terms—operation planning under uncertainties, reliability management, machine learning, SCOPF, proxy

I. INTRODUCTION

The increasing penetration of renewable energies coupled with energy markets lead to increasing levels of uncertainties in the context of look-ahead (a few hours to a few days ahead in time) operation planning. As a consequence, the traditional approach for determining operation planning decisions, based on a single ‘most likely’ forecast along the considered look-ahead horizon, is not appropriate anymore. Indeed, the observed real-time realizations are farther and farther from this forecasted trajectory, making compliance with the reliability target more and more difficult. To progress, one possibility is to plan operation over a representative set of possible future operating conditions while modeling the way the real-time operator would respond along these trajectories. The purpose would be to choose operation planning decisions making the compliance with real-time reliability targets feasible (with high enough probability) while minimizing (the expectation) of operating costs.

Real-time reliability management (i.e., assessment and control) essentially aims at ensuring that the system may survive (without too large service interruptions) any contingency within a postulated list of credible contingencies. In practice, this is expressed by the N-1 criterion, and reliability management in real-time can be modelled suitably by a so-called

‘Security Constrained Optimal Power Flow Problem’ (SCOPF) [1]. In the literature, probabilistic versions of this approach have as well been proposed, leading to a ‘Probabilistic-SCOPF’ formulation, where the covered list of contingencies is dynamically adjusted by taking into account the weather dependent probabilities of failures and the costs of potential service interruptions [2].

To take into account uncertainties in operation planning it is necessary to model in a suitable way the real-time reliability management strategy over many time steps and many look-ahead scenarios, which implies a challenging computational burden. To make this tractable, we propose to apply machine learning in order to automatically build simplified models (we call them *proxies*) of the outcome of real-time reliability management; these proxies should be orders of magnitude faster to compute, and at the same time sufficiently accurate, so that they can be used instead of the detailed SCOPF type of computational models in the context of operation planning reliability management.

Considering this, the contribution of this paper is twofold. First we investigate the use of machine learning (and in particular supervised learning) to predict some outputs of real-time reliability management such as the costs of real-time recourse decisions and the level of system reliability they induce. In particular different supervised learning algorithms are tested in order to evaluate which ones are the most appropriate for this problem and the time gain obtained with the machine learnt proxies with respect to the full SCOPF computation is evaluated. Then we show that with the by-products of some supervised learning algorithms (tree-based ensemble methods) we are able to analyze the relevance of features to predict the studied outputs.

The rest of this paper is organized as follows. Section 2 describes the proposed methodology for using (supervised) machine learning for the construction of proxies of real-time reliability management. Section 3 presents an empirical study of this proposal on the IEEE-RTS96 benchmark, and Section 4 discusses related works. Finally Section 5 concludes and outlines directions for future research.

II. RT-OPERATION PROXY BUILDING METHODOLOGY

We start by modeling RT-operation in the form of a SCOPF program, followed by the assessment of the resulting reliability level gotten via a cascade simulator. Then, in order to

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement No 608540, project acronym GARPUR. The scientific responsibility rests with the authors.

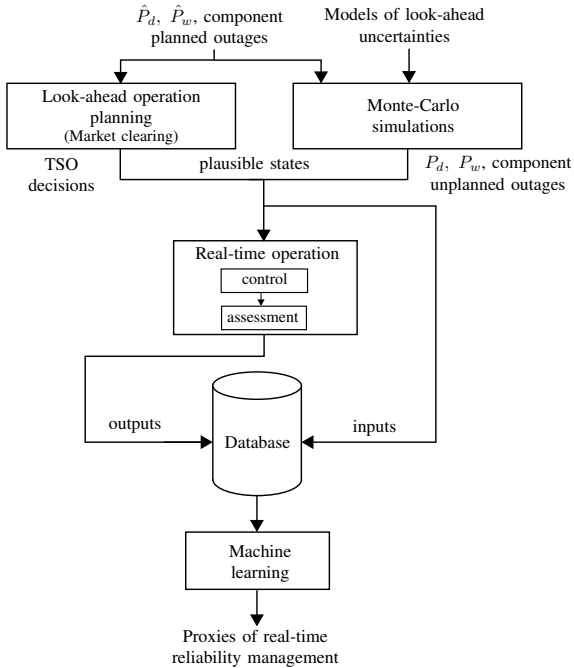


Fig. 1. Methodology used to build the proxies. \hat{P}_d and \hat{P}_w are the forecast load and wind generation while P_d and P_w are the realizations.

use supervised learning to predict the outcome of real-time operation, we build a database whose inputs describe the real-time operating conditions and whose outputs are the outputs of this SCOPF program and of this reliability assessment program. Fig. 1 depicts the whole methodology. Section II-A describes how the database may be generated with the help of Monte-Carlo simulations and in a look-ahead context, while Section II-B describes how supervised learning algorithms can be used to build the proxies and how these algorithms should be evaluated.

A. Database generation

Let us, for the sake of explanation, consider the day-ahead operation planning context. Our database in such case would include a look-ahead horizon spanning the full 24 hours of the upcoming day and several plausible states for each hour. Such states would be generated by combining the day-ahead market clearing outcome with chosen TSO day-ahead decisions (if any) and stochastic models of day-ahead uncertainties. Particularly, stochastic models of load, wind generation and component forced outage (generators, lines, phase-shifting transformers, etc.) would be needed to capture the difference between the anticipated state of the system at the day-ahead and the realized state of the system in real-time.

Once a dataset of possible real-time states is generated, we apply to each state a simulator of real-time reliability control followed by an assessment of its outcome. The purpose of real-time control is to adapt to the uncertainty realization (different from the forecast due to the unavoidable forecast errors) to ensure that the real-time reliability target is met. The outputs of real-time control are in particular the real-time control costs

and decisions. Then the assessment stage computes a measure of the system reliability level.

B. Use of supervised machine learning

In order to predict continuous outputs, we compared the following regression algorithms:

- random forest (RF) [3] with 500 trees,
- extremely randomized trees (ET) [4], also with 500 trees,
- ridge regression (RR) and kernel ridge regression (KRR) [5],
- support vector regression (SVR) [6],
- fully connected neural networks (NN) [5] with 3 hidden layers, 100 neurons per layer and rectified linear units (ReLU) as activation functions.

For the KRR and SVR algorithms, we tested three different kernels: linear, polynomial and gaussian. Note that the KRR algorithm with a linear kernel actually corresponds to the ridge regression algorithm.

In order to train and then evaluate the accuracy of the resulting proxies, we divided the dataset into two subsets: the learning set which contains 80% of the samples and the test set which contains the remaining 20%. The learning set is used to build the proxies and we tested how these proxies predict unseen data with the test set. To assess the performance of a model, we used the R^2 -score (coefficient of determination), which is computed on the basis of N cases by [15]:

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

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.

We also used a 5-fold cross-validation [5] over the learning set to find the best meta-parameters associated with each algorithm. A 5-fold cross-validation consists in dividing the learning set in 5 folds and then using 4 folds to build a model and the fifth one to test it. If you repeat that 5 times while changing the fold left out, you will obtain 5 scores that you can average to form the cross-validation score which is an estimate of the generalization performance of the model. We kept the combination of meta-parameters leading to the best cross-validation score for each algorithm.

Note that for the RR, KRR, SVR and NN methods the inputs and outputs should be standardized, since these algorithms are not scale-invariant.

III. CASE STUDY ON THE IEEE-RTS96 BENCHMARK

All the experiments were run on a Toshiba Satellite computer with an Intel®Core™ i7-3610QM @ 2.3GHz as processor and 6GB of RAM. The assessment and the control problems were implemented respectively with Matlab [7] and with GAMS [8]. Finally, the supervised learning library used in this work is Scikit-learn [9].

We tested our methodology on a modified version of the IEEE-RTS96 single area network [10]. We added nine wind

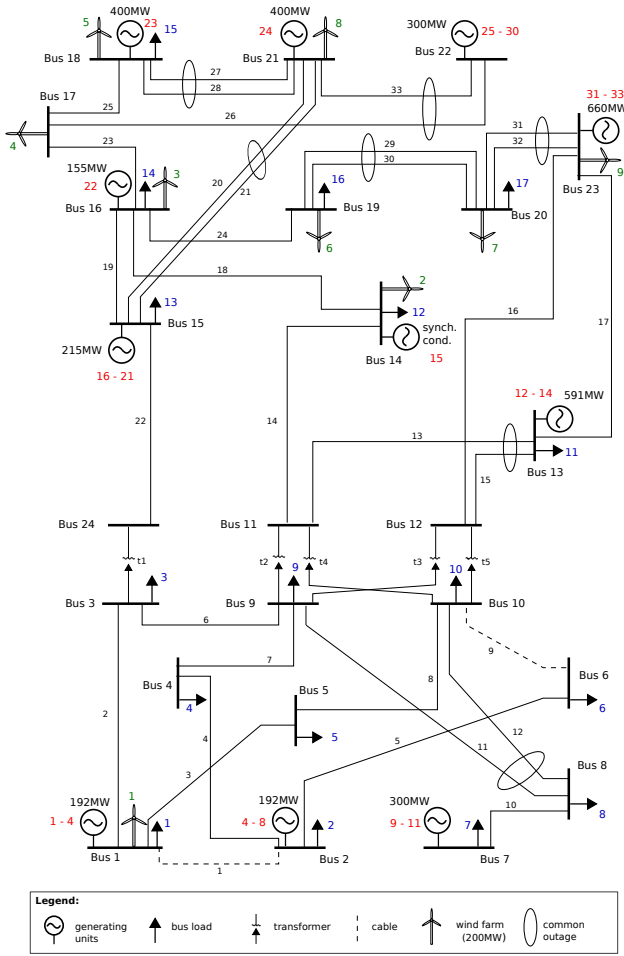


Fig. 2. Modified version of the IEEE-RTS96 network.

farms as specified in [11]. However, we limited the capacity of each wind farm to 300MW. The one-line diagram of the system is depicted in Fig. 2.

A. Database generation in a day-ahead context

We chose to generate our database in a day-ahead context. We considered one particular day for which we have load and wind generation forecasts and no planned outage. In order to generate multiple plausible scenarios for this day, we considered the realizations of load and wind generation based on day-ahead forecast error and also possible unplanned outages of transmission lines, transformers and generating units. We refer the reader to [12] for more details on the database generation used in this paper.

The day-ahead market clearing is simulated with a unit-commitment imposing N-0 constraints and based on the forecast of load and wind generation. The upward and downward reserves have a capacity at least as large as the largest market output power among the generating units in service, in order for the system to be able to withstand the loss of any thermal generating unit.

For real-time control, we used the N-1 criterion but we only looked at lines and transformers contingencies. We

considered only preventive generation rescheduling and if needed preventive wind curtailment and load shedding. These preventive actions are determined by a preventive SCOPF and the DC network-model was used to express the network constraints. The set of constraints comprises maximum and minimum generation for each generator, ramp-up and ramp-down constraints, nodal power balance and transmission line thermal ratings. We allow continuous load shedding and wind curtailment. The objective function minimizes load shedding and wind curtailment costs, that are respectively defined as the sum of load shed multiplied by the average value of lost load and the sum of wind curtailed multiplied by a wind penalty. The aim is to avoid load shedding and wind curtailment as much as possible. Finally, the redispatch cost is defined as:

$$C_{redispatch} = \sum_{g \in \mathcal{G}} C_g^p (P_g^{up} + P_g^{down}),$$

where \mathcal{G} is the set of generating units, C_g^p is the redispatch cost of generating unit g per MW and P_g^{up} and P_g^{down} are respectively the amount of preventive ramp-up and ramp-down of g . Whenever the SCOPF program is infeasible for a sample (despite load shedding and wind curtailment), we simply discard it. Note that at the day-ahead stage, generating units 9 and 10 on bus 7 were forced to be on in order to avoid load shedding. Indeed, in case line 10 is in outage, if at least two of the generating units are not on, load 7 must be shed and since we only defined preventive actions, we prefer to prevent that by forcing these two units to be in service.

The assessment program is a more detailed cascade simulator based on [13], allowing us to compute the expected amount of load shed over the trajectories of the system induced by a set of possible contingencies, which in addition to N-1 events also comprises some common mode double outages (circled in Fig. 2). In particular, it computes the *risk*, which is defined as [2]

$$\text{Risk} = \sum_{c \in \mathcal{C}} \pi_c(w_0) \sum_{d \in \mathcal{D}} \text{voll}(d) * LS_c(d), \quad (1)$$

where \mathcal{C} is the set of considered contingencies, \mathcal{D} is the set of loads, π_c is the probability of contingency c and depends on the weather w_0 , $\text{voll}(d)$ ¹ is the value of lost load for load d and $LS_c(d)$ is the amount of load d shed at the end of the cascade of phenomena following contingency c .

Our dataset contains 4000 samples. For each sample, it takes in average 0.04s to solve the real-time control problem and then 0.89s to solve the real-time assessment problem.

We studied in particular 5 outputs: total cost of preventive actions, cost of preventive generation rescheduling, cost of preventive load shedding and cost of preventive wind curtailment as outputs of the reliability control program and risk as output of the reliability assessment program.

¹Note that we use a *voll* depending of the load d because we want the risk to be a precise estimate of the level of reliability of the system. In contrast, we define the load shedding cost of the control program with an average *voll* to avoid discriminating by this factor in situations where load shedding is necessary.

TABLE I
TEST SCORES (TS) AND LEARNING SCORES (LS) FOR EACH ESTIMATOR.

Outputs	ET		RF		RR		Gaussian KRR		poly KRR		linear SVR		Gaussian SVR		poly SVR		NN	
	TS	LS	TS	LS	TS	LS	TS	LS	TS	LS	TS	LS	TS	LS	TS	LS	TS	LS
Total Cost	0.879	1	0.842	0.979	0.821	0.818	0.781	0.797	0.926	0.990	0.788	0.734	0.921	0.982	0.824	0.981	0.964	0.998
Redispatch Cost	0.944	0.998	0.941	0.992	0.898	0.908	0.891	0.908	0.936	0.983	0.820	0.824	0.900	0.999	0.887	0.972	0.943	0.994
Load Shedding Cost	0.884	1	0.832	0.952	0.819	0.816	0.778	0.795	0.925	0.990	0.783	0.770	0.919	0.981	0.823	0.981	0.962	0.997
Wind Curtailment Cost	0.883	0.996	0.863	0.979	0.619	0.656	0.703	0.842	0.858	0.990	0.598	0.613	0.800	0.910	0.678	0.957	0.861	0.996
Risk	0.754	0.925	0.730	0.958	0.640	0.684	0.723	0.778	0.749	0.854	0.552	0.559	0.706	0.930	0.694	0.863	0.707	0.870

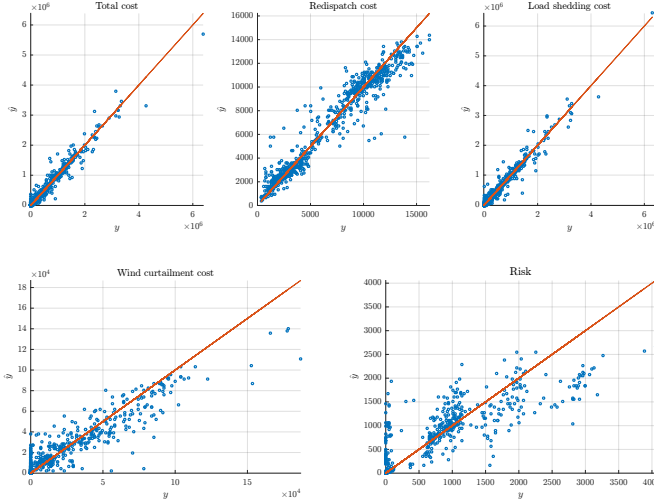


Fig. 3. Scatter plots showing the true values (y) of the test set against the values predicted by the best proxies (\hat{y}) for each output.

The considered input variables describe the realized states of the system. They include the realizations of nodal load and wind generations as well as the availability of each component of the system (generating units, lines and transformers), the total load, the total wind generation and the load net of wind (net load). Furthermore, to model that the outage probabilities of transmission lines depend on the weather conditions, we defined two weather states, namely normal and adverse [14]. Therefore the weather status is part of the input variables. Concerning the TSO day-ahead decisions, the input variable per hour is the market dispatch of each generator, pre-computed in day-ahead mode. Finally, when we build the proxy predicting the system’s risk, we consider in addition probabilities of contingencies. Note that for the RR, KRR, SVR and NN methods the inputs and outputs have been standardized with the means and standard deviations of the learning set.

B. Evaluation of the machine learnt proxies

Table I presents the test and learning scores of the different algorithms for each output. To determine them, the model is learnt with the learning set and then used to predict the test set (TS) or the learning set (LS). Recall that the score chosen is an R^2 -score, for which the best possible value is 1.

One can see in Table I that for the Total Cost and the Load Shedding Cost the best method is ‘NN’ (neural networks). For the other outputs, extremely randomized trees (‘ET’) are better. We also observe that the Risk is the most difficult variable to predict. Finally, we observe that the two linear models (‘RR’ and ‘linear SVR’) are clearly outperformed by the non-linear models, especially for the prediction of the Wind Curtailment Cost and the Risk.

To give an idea about the quality of the prediction, Fig. 3 shows the scatter plots representing the true values of the test set against the values predicted with the best proxies (in bold in Table I). It can be seen that most data points seem to follow the line $y = \hat{y}$, showing that they are well predicted. It is clear in this figure that the predictions are not as good for the risk as for the other outputs.

The time needed to predict the Risk with our proxies is 0.035ms per state in average, which is a great gain with respect to the 0.89s needed by the model used to generate the dataset. Furthermore, it takes 0.04ms in average to predict each control output, while the SCOPF implementation used to generate the dataset required in average 0.04s per state. Therefore the gain in time achieved by our proxies is at least in the order of 10^3 .

C. Study of the relevance of input features

Supervised machine learning can also be used to find the most important input variables to predict an output, in order to have a better understanding of the problem. Input variable “importances” can be computed as a by-product of training models in the form of random forests or extremely randomized trees (see [16] for a study of their theoretical properties).

To illustrate this kind of analysis, we show in Fig. 4 the 15 most important variables and their relative importance for two of the five studied outputs.

Let’s first analyze Fig. 4(a), concerning the Total Cost (the sum of the preventive redispatch, preventive load shedding, and preventive wind curtailment costs). We see that the most important input is the market dispatch of generator 14, which is on bus 13. If one looks at the total amount of up and down ramping per generator over the database, it can be noticed that generator 14 is one of the generating units for which the amount of redispatch is the largest. Furthermore, it is an oil/steam unit and one of the most expensive ones. This can explain the importance of this generator for the total cost. Note that similar remarks can be made for the market

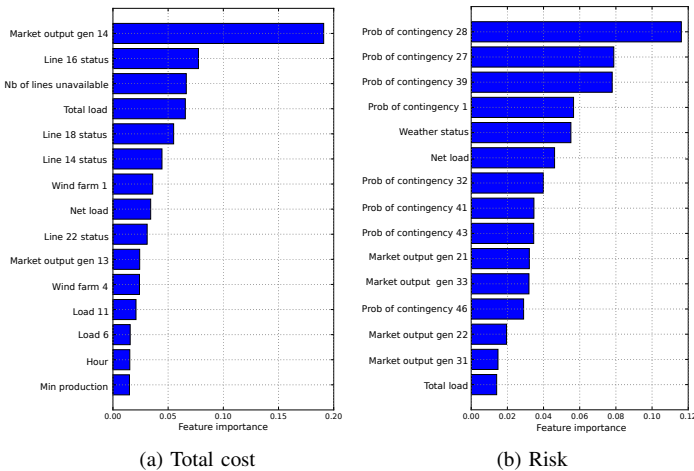


Fig. 4. Input variable importance ranking for (a) total cost and (b) risk level. Only the 15 most important features are shown in each plot. The horizontal axis measures the contribution (as a fraction $\in [0; 1]$) of the information provided by each input variable, normalized so that the total over all inputs is equal to 1.

dispatch of generator 13, which also appears in the feature ranking. Indeed the total amount of redispatch of generator 13 is slightly smaller than the one of generator 14 and the unit is as expensive.

The second most important variable is the availability status of line 16. Its presence in this ranking indicates that this line is important for the preventive control. Indeed line 16 connects the area of the network where most generators are concentrated to the area where most loads can be found and thus the outage of this line may significantly increase the stress on the network and require more preventive actions. The same observations can be made concerning lines 14, 18 and 22, the availability statuses of which also appear in the 15 most important features.

The other important features are the number of lines in outage, the total and net load, some loads and wind farm generations and the minimum stable generation limit of the dispatchable units that are online. Load 11 is the load for which the total amount of load shed is the largest while wind farm 1 is the wind farm that is most often curtailed. These elements can explain why these variables are important. As could be expected, we can find in these 15 features elements impacting generation rescheduling, load shedding and wind curtailment.

If we now look at the most important features for the Risk (Fig. 4b), we see that there are many probabilities of contingencies. One could have expected that only the double outages would appear (contingency numbers greater than 39), given that single outages are covered by the N-1 criterion. However, it is not the case. One can even notice that the probability of having no outage (defined as probability of contingency 1) is the fourth most important feature to predict the risk. In fact all contingency probabilities carry information on the weather state and simultaneous component outages.

Indeed all contingency probabilities change with the weather state and with the concurrent occurrence of outages. Therefore, one can deduce that the weather clearly impacts the risk, as well as the real-time topology. Note that the weather status is also among the 15 most important features.

Regarding the definition of risk (1), it is also not surprising to see that the total load is an important feature. The net load is an indication of the level of stress on the network and can therefore directly impact risk. Concerning the market dispatch of generators 21, 31 and 33, these generators are located on buses connected to other buses with double lines. Since the system is not operated to withstand the simultaneous loss of these double lines, when there is a common mode double outage coupled with a non-zero generation on the bus of concern, the risk can increase. The importance of the market dispatch of generator 22 can be explained similarly, given that most buses connected to bus 16 are linked to the rest of the system with a double line.

IV. RELATED WORK

Our work is certainly not the first work on the use of machine learning in the context of power systems reliability management. As a matter of fact, a large body of work has been carried out in this context during the 1980's and 1990's, with the goal of building classifiers for fast assessment of transient stability and voltage stability (see [17] for some early work, and [18] for a more comprehensive bibliography). More recently, the European project iTesla [19] has developed an industrial software platform to apply these techniques for real-time security assessment. In these works, machine learning is used to build security rules in the form of decision trees or neural networks, to predict the dynamic response in real-time of the power system, on a per-contingency basis, so as to speed up real-time dynamic security assessment.

Furthermore, several papers have already studied the possibility to use machine learning in multi-stage decision-making programs to build proxies of shorter-term stages. For instance, we refer the reader to [20] and [21], in which the authors have built proxies of respectively day-ahead unit commitment and real-time AC-OPF for a mid-term to long-term planning purpose. In [20] the nearest neighbor algorithm is used to predict the costs and decisions of a day-ahead unit commitment program while in [21], several supervised learning algorithms are tested to predict the cost and feasibility of an AC-OPF problem.

In contrast, in the present work we propose to use machine learning to predict the response of the real-time reliability management process, so as to speed-up its simulation during the day-ahead reliability management process, and thus allow one to more effectively take into account uncertainties in this latter process. Hence, our models predict the outcome of the real-time reliability management process responding to day-ahead forecast errors, sudden changes in weather conditions, and/or forced network component outages in terms of real-time costs and risk. Furthermore, we decompose the total cost between redispatch cost, load shedding cost and wind

curtailment cost in order to be able to analyze each of them separately. It will allow us to know for the predicted samples when load shedding and/or wind curtailment is necessary. This way, in a further step we will be able to select a day-ahead decision minimizing load shedding rather than the total cost for example, depending on the objective of the TSO.

V. CONCLUSIONS

In this paper, we presented a methodology to build machine learnt proxies able to predict the outcome of real-time reliability management response in a look-ahead operation planning context. We tested several supervised learning algorithms and used them to predict in particular the reliability management costs and risk. Furthermore, we used the extremely randomized trees algorithm to rank the features and know which variables have more effect on the studied outputs. The purpose is to have better insight into the problem.

Applying the methodology to the IEEE-RTS 96 network in a day-ahead context, we showed that it is effectively possible to use supervised machine learning to build the proxies. The results are good with test scores close to 90% for most studied outputs. Furthermore the time gain is significant, with an order of magnitude of 10^3 . The results given by the different supervised learning algorithms are close to each other but proxies learnt with extremely randomized trees, neural networks and ridge regression with a polynomial kernel are the most accurate ones.

Concerning the feature importance analysis, we noticed that probabilities of contingencies, weather conditions and net load are important to predict the risk while variables representing market generation of most re-dispatchable units, availability of important lines, loads that are most shed and wind farm generations that are most curtailed are ranked first to predict the total preventive cost.

However, there is still room for improvement. One possibility to improve the prediction is to increase the size of the learning set. Another possibility is to use feature selection to learn with only the most important features and especially remove non-relevant features. Finally, given that non-linear methods give the best results, it would be interesting to investigate the use of deep learning to predict the outcome of the real-time reliability management.

Possible future research directions are the improvement of the prediction for the studied outputs and especially for risk and the development of methods to predict the SCOPF feasibility of a particular sample as well as the real-time control decisions.

Finally, when the proxies will be accurate enough, a future work will be to import them in a suitable way into the look-ahead reliability management problems.

ACKNOWLEDGMENTS

The authors are most grateful to Camille Hamon for his useful editorial suggestions.

REFERENCES

- [1] F. Capitanescu, "Critical review of recent advances and further developments needed in ac optimal power flow," *Electric Power Systems Research*, vol. 136, pp. 57–68, 2016.
- [2] E. Karangelos and L. Wehenkel, "Probabilistic reliability management approach and criteria for power system real-time operation," in *Power Systems Computation Conference (PSCC), Genoa, Italy*, June 2016.
- [3] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [4] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine Learning*, vol. 63, no. 1, pp. 3–42, 2006.
- [5] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. Springer Series in Statistics, 2009.
- [6] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and computing*, vol. 14, no. 3, pp. 199–222, 2004.
- [7] "MATLAB and Statistics Toolbox Release 2015b," The MathWorks, Inc., Natick, Massachusetts, United States.
- [8] "GAMS Development Corporation. General Algebraic Modeling System (GAMS) Release 24.0.2," Washington, DC, USA, 2012.
- [9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [10] 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.
- [11] 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.
- [12] L. Duchesne, "Machine learning of proxies for power systems reliability management," Master's thesis, Université de Liège, Liège, Belgique, 2016.
- [13] J. Yan, Y. Tang, H. He, and Y. Sun, "Cascading failure analysis with dc power flow model and transient stability analysis," *IEEE Transactions on Power Systems*, vol. 30, no. 1, pp. 285–297, 2015.
- [14] R. Billinton and R. N. Allan, *Reliability evaluation of power systems*, 2nd ed. Springer, 1996.
- [15] 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.
- [16] G. Louppe, L. Wehenkel, A. Suter, and P. Geurts, "Understanding variable importances in forests of randomized trees," in *Advances in Neural Information Processing Systems*, 2013, pp. 431–439.
- [17] L. Wehenkel, T. Van Cutsem, and M. Ribbens-Pavella, "An artificial intelligence framework for online transient stability assessment of power systems," *IEEE Transactions on Power Systems*, vol. 4, no. 2, pp. 789–800, 1989.
- [18] L. A. Wehenkel, *Automatic learning techniques in power systems*. Springer Science & Business Media, 2012.
- [19] iTesla: Innovative Tools for Electrical System Security within Large Areas. Collaborative R&D project co-funded by the European Commission (7th Framework Programme), <http://www.itesla-project.eu/>, accessed on 2016-06-06.
- [20] 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.
- [21] R. Canyasse, G. Dalal, and S. Mannor, "Supervised learning for optimal power flow as a real-time proxy," *arXiv preprint arXiv:1612.06623*, 2016.