Machine learning based binding contingency pre-selection for AC-PSCOPF calculations

Nipun Popli* Elnaz Davoodi† Florin Capitanescu†(*Senior Member IEEE*) Louis Wehenkel*

***Abstract*—We propose to use off-line machine learning to train an oracle predicting the set of binding contingencies for an Alternating Current Preventive Security-Constrained Optimal Power Flow (AC-PSCOPF) solver. On-line, the oracle's predictions are used instead of the full set of all postulated contingencies, as an input to the PSCOPF solver. A Steady-State Security Assessment (SSSA) is applied to the resulting PSCOPF solution to check the absence of false negatives. Our oracle is a deep neural network multi-label classifier that uses as inputs active and reactive loads, generations, and power flows, computed by an OPF using the same cost function and base-case constraints as the PSCOPF. The proposal is show-cased on the Nordic32 benchmark.**

***Index Terms*—machine learning, multi-label classification, security-constrained optimal power flow, security assessment.**

I. Introduction

The AC-PSCOPF is a nonlinear optimization problem aiming at finding an optimal operating point of a power system that is also secure with respect to a set of contingencies [1]. For large power system models with tens of thousands of buses, lines, and contingencies, this leads to nonlinear optimization problems with billions of variables and constraints, which are currently not solvable accurately in a tractable way.

A way to make PSCOPF calculations more tractable would be to use an oracle to guess the subset of *binding contingencies*; the latter are contingencies that at the PSCOPF solution have at least one non-zero Lagrange multiplier associated to their branch loading or bus voltage limits. Solving the PSCOPF with only these contingencies explicitly modelled may be much more efficient and would yield a solution that would also be secure for the non-binding contingencies.

In order to identify an as small as possible subset of contingencies containing all the binding ones, heuristic iterative approaches have been proposed in the literature [2]. These methods progressively grow the subset of covered contingencies, until the resulting PSCOPF solution is found to be secure against all other (not explicitly modelled) contingencies. They typically need several iterations and converge to a superset of the subset of binding contingencies. In practice however, the number of potentially binding contingencies they find remains less than a few tens, even for very large systems [3].

While Machine Learning (ML) has been proposed recently to enhance AC-PSCOPF in various ways [4], [5], we propose in this letter, for the first time, to use ML, namely supervised learning of deep neural network based multi-label classifiers, to pre-select *binding* contingencies for the AC-PSCOPF.

Our method works in the following way:

1) in off-line mode, we generate a dataset of solved PSCOPF instances and use supervised ML to train an oracle that can predict binding contingencies,
2) in on-line mode,
   a) we run the PSCOPF with the subset of binding contingencies predicted by the learnt ML-oracle,
   b) we run SSSA, to check whether some of the other postulated contingencies lead to security violations; if this is the case, we add them to the contingency subset and return to step a.
3) we continuously enrich the training set with the subsets of binding contingencies found in on-line mode and, if needed, we retrain the ML-oracle in off-line mode.[1]

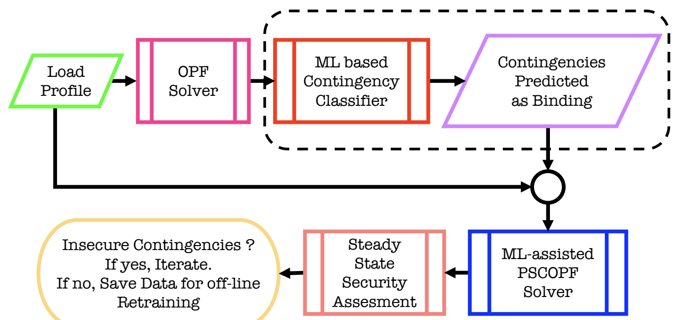

Fig. 1. On-line use of the ML-oracle to speed up PSCOPF computations

II. ML-based binding contingency detection

In Fig. 1, we present the on-line workflow for using our classifier-assisted PSCOPF solver. It begins with computing an OPF solution (i.e. without any contingency) by applying a physics-based alternating-current solver on the given load profile. This OPF solution is then fed into the contingency classifier, which works as an oracle predicting the set of binding contingencies. The predicted set is used as input for a physics-based alternating-current PSCOPF solver, generating the ML-assisted PSCOPF solution. In the final step, the SSSA identifies contingencies for which the classifier-assisted PSCOPF solution is found to be insecure.

A. Proposed multi-label classification formulation

Our primary objective is to predict potentially binding contingencies for a given load profile. We formulate this

The authors acknowledge the funding from the FNRS-Belgium for the project ML4SCOPF under grant T.0258.20, and the funding from FNR-Luxembourg in the framework of the project ML4SCOPF (INTER/FNRS/19/14015062). Computational resources have been provided by the CÉCI-FNRS under Grant No. 2.5020.11 and by the Walloon Region.

*Institut Montefiore, Université de Liège, Liège, Belgium (email: {n.popli, l.wehenkel}@uliege.be)

†Luxembourg Institute of Science and Technology, Esch-sur-Alzette, Luxembourg (email: {elnaz.davoodi, florin.capitanescu}@list.lu)

[1]Retraining would be needed as soon as the accuracy of binding contingency detection shows a significant degradation. This could be caused by important system topology changes and/or by significant drifts in demand patterns or in the cost functions or constraints used by the PSCOPF [5].

problem as a multi-label classification task, each label being dedicated to one of the postulated contingencies. To construct a multi-label binding contingency classifier, we extract features from solutions obtained using alternating-current formulations of OPF and PSCOPF solvers (see Fig. 2).

- **Input Features**: With a load profile represented by $P_D, Q_D \in \mathbb{R}^d$ for d aggregate demands, calculate an OPF solution in the absence of contingencies. Obtain vectors of generation schedules, denoted as $P_G, Q_G \in \mathbb{R}^g$, representing the power output of g power sources, and branch flows denoted as $P_F, Q_F \in \mathbb{R}^f$ for f lines and transformers. Concatenate all vectors to obtain an input feature vector as $[P_D, Q_D, P_G, Q_G, P_F, Q_F]$.
- **Output Features**: For the same load profile $P_D, Q_D \in \mathbb{R}^d$, and given a set of postulated contingencies $\mathcal{M}$, compute a PSCOPF solution (with the same objective function as used in OPF computations). Obtain the set of binding contingencies as a bit-vector $\mathcal{B} \in \{0,1\}^{|\mathcal{M}|}$. $\mathcal{B}$ is a vector of (binary) output features, indicating the binding (1) or non-binding (0) nature of each contingency.

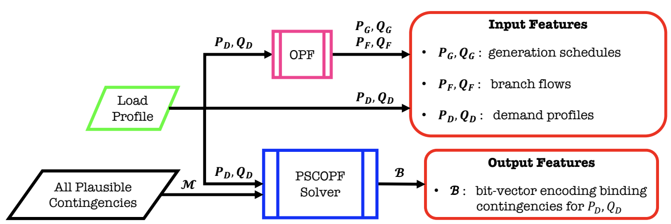

Fig. 2. Dataset construction pipeline for off-line training of the ML-oracle

Our decision to utilise OPF solutions to compute input-features used by our oracle is driven by three reasons:

1) **Optimization problem awareness**: OPF solutions tell us something about how the cost function selects the set of binding constraints of the optimization problem.
2) **Physics awareness**: OPF solutions provide branch flows and generation schedules that comply with system topology and physical characteristics. (In the case of voltage limited systems, we could similarly include voltage profiles computed by the OPF as input-features.)
3) **Availability**: OPF results without considering the postulated contingency set should normaly be available in any on-line context where a PSCOPF is to be used.

The process of constructing datasets for the multi-label classifier training thus involves performing both OPF and PSCOPF computations, for both training and test samples.

B. Machine learning setting

We illustrate our approach on the Nordic32 test system, as shown in Fig. 3. A comprehensive system description, the non-linear PSCOPF formulations, and the precise dataset generation assumptions can be found in [5].

Dataset generation: We generated $14,000$ load profiles ranging between their peak and lowest values, and introduced bus-wise independent Gaussian noises in both real power demands and power factors. Exactly $12,031$ resulted in feasible PSCOPF solutions when considering 33 line-based and 19 generator-based (single-outage) contingencies, i.e., $|\mathcal{M}| = 52$.

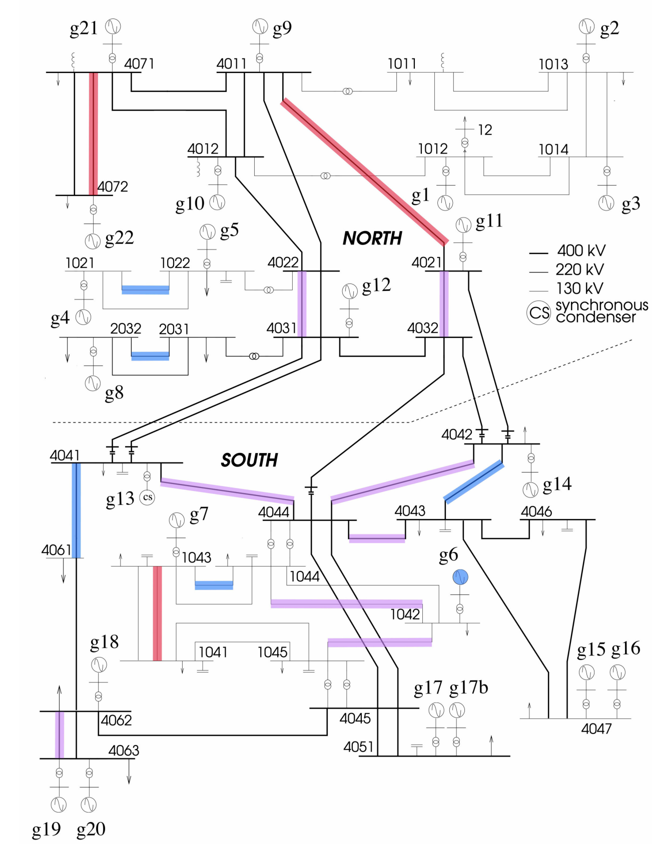

Fig. 3. Binding contingencies observed for the Nordic32 system: *Red: binding for $> 99\%$ of the $12,031$ samples, Lavendar: binding for 22%–96% of the samples, Blue: binding for 2.2% – 10.9% of the samples*

OPF solutions with the same cost function were computed for these latter. In our study all SCOPF/OPF runs are based on the IPOPT solver via the JuMP framework, as in [5].

Preprocessing: Consider Fig. 3. Out of $12,031$ PSCOPF solutions, 3 line-based contingencies (marked in red) were consistently observed as binding in over 99% of $12,031$ samples. Then, 8 line contingencies (highlighted in lavender) exhibited a binding occurrence in 22% to 96% of $12,031$ samples. Finally, 5 line and 1 generator failures (accentuated in blue) were found to be binding in 2.2% to 10.9% of the total samples. For classifier training, we consider contingencies with a minimum 2% minority class labels. Specifically, we exclude contingencies that were either binding or non-binding in less than 2% of $12,031$ PSCOPF solutions. For instance, we assume 3 red-highlighted line contingencies to be always binding and the non-highlighted ones as never binding. Effectively, we reduced in this way the dimension of the output feature vector $\mathcal{B} \in \{0,1\}^{52}$ to $\mathcal{B}_\mathrm{r} \in \{0,1\}^{14}$.

Class Imbalances: A major challenge in multi-label classification is the substantial diversity in the underlying distributions of binary labels (binding or non-binding) for each label (plausible contingency). For example, line $4011 - 4021$ is identified as a binding contingency in $11,910$ out of $12,031$ samples. In contrast, tripping of generator g6 is observed to be binding in only 1064 out of $12,031$ samples. The dataset is severely imbalanced, in different ways for different contingencies. Traditional methods such as over-sampling of minority class [6], under-sampling of majority class [7], or a combination of both [8] are invoked for class balancing,

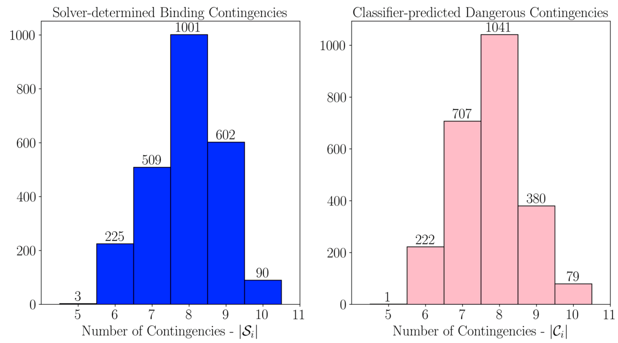

Fig. 4. Histograms showing the number of test operating conditions (y-axis) for different binding contingency set cardinalities (x-axis): PSCOPF solver-determined ($|\mathcal{S}_i|$, in Blue); ML based classifier-predicted ($|\mathcal{C}_i|$, in Pink)

often altering the underlying distributions. Instead, we found it more suitable in our multi-label context to utilise weighted binary cross entropy as a loss function to address class imbalance. Herein, incorrect predictions for minority labels are naturally penalised higher compared to imperfect predictions for majority labels. For each output label, we assigned different weights to its binary labels based on their frequencies in the dataset. This ensured that the trained classifier accorded equal emphasis on correct prediction of less-frequent labels, while preserving the underlying joint distribution of binary labels.

III. ML-ASSISTED AC-PSCOPF SOLVER RESULTS

In this section we provide, in the context of the Nordic32 benchmark, a concise overview of the multi-label classifier training setting and we outline the main results in terms of accuracy of the trained classifier.

Features: **Non constant Input Features** - $P_D, Q_D \in \mathbb{R}^{22}$ for 22 loads, $P_G \in \mathbb{R}^{22}$, $Q_G \in \mathbb{R}^{23}$ for 21 synchronous generators, a tie-line flow, and one synchronous condenser, $P_F, Q_F \in \mathbb{R}^{72}$ for 36 branches (at both ends); **Reduced Output Features** - a bit-vector $\mathcal{B}_\mathrm{r} \in \{0,1\}^{14}$ corresponding to contingencies marked in lavender and blue in Fig. 3 (red-marked contingencies are assumed to be always binding).

Neural Network Architecture: A fully-connected feed forward neural network architecture was selected with the following hyper parameters: an input layer with 233 neurons and two hidden layers with 700 neurons each, all with ReLU activation functions, the output layer with 14 neurons uses sigmoid activation functions. For training, the Adam optimiser was harnessed with a weighted binary cross-entropy loss function. The input features were standardised towards zero mean and unit standard deviation. PyTorch-1.13.1 was used to train the multi-label classifier $\hat{C}_\mathrm{r} : \mathbb{R}^{233} \to \{0,1\}^{14}$ with 1400 epochs, a mini-batch size of 25, and a learning rate of $3 \cdot 10^{-4}$.

Training, Validation and Testing: The dataset was split into $80:20$ ratio, with 80% allocated for training and 20% unseen ones for testing. During training-validation steps, the training set was further split into $80:20$ ratio for hyper-parameter tuning. Our parameter tuning objective was to minimise incorrect predictions of observed binding contingencies. To achieve this, we tuned 14 positive weights within the loss function to reduce the number of false negatives (FN). It is important to note that false positives (FP) are undesirable from a computational perspective. But, unlike FN, presence of FP do not adversely affect the security of classifier-assisted PSCOPF solutions. Of course a perfect predictor would result in only true positives (TP) and true negatives (TN). Other hyper-parameters were tuned to ensure a smoother convergence of validation losses within each epoch. The final multi-label classifier was trained using the complete training dataset, i.e., with 9600 samples.

TABLE I
ACCURACY STATISTICS OF THE BINDING CONTINGENCY CLASSIFIER OVER THE TEST SET OF 2431 UNSEEN OPERATING CONDITIONS

Contingency	Accuracy	Recall	Specificity	Precision	MCC
1021-1022	99.5%	94.3%	99.7%	91.1%	0.924
1042-1044	95.8%	97.3%	92.6%	96.5%	0.903
1042-1045	95.6%	95.5%	95.7%	92.1%	0.905
1043-1044	99.2%	89.0%	99.5%	85.5%	0.869
2031-2032	99.3%	95.2%	99.5%	91.5%	0.930
4021-4032	98.3%	96.7%	98.8%	95.9%	0.952
4022-4031	97.9%	98.1%	97.6%	98.6%	0.955
4041-4044	99.5%	99.7%	94.5%	99.8%	0.932
4041-4061	98.6%	69.6%	99.2%	61.5%	0.647
4042-4043	93.2%	78.9%	94.9%	64.5%	0.676
4042-4044	97.6%	99.5%	95.1%	96.3%	0.951
4043-4044	95.2%	97.0%	86.9%	97.3%	0.834
4062-4063	99.0%	98.4%	99.3%	98.8%	0.978
G6	99.2%	97.6%	99.3%	93.1%	0.949
All	97.7%	97.4%	97.9%	96.1%	0.950

MCC: Matthews correlation coefficient.

Accuracy assessment of the learnt oracle

For an operating condition i amongst the 2431 unseen test operating conditions, we consider two sets of contingencies: set $\mathcal{S}_i$, which contains the ground-truth of binding contingencies determined via conventionally-computed all-plausible-contingency PSCOPF solutions, and set $\mathcal{C}_i$, which contains classifier-predicted binding contingencies. Fig. 4 contrasts the distributions of cardinalities for sets $\mathcal{S}_i$ and $\mathcal{C}_i$ over the 2431 test samples. We observe that the distribution for $|\mathcal{C}_i|$ (in pink) is slightly more skewed towards the y-axis.

Table I provides various standard accuracy indicators over the set of 14 contingencies covered by the trained multi-label binding contingency classifier.[2] While the overall performances are very good, the table shows that for the outage of line 4041-4061 or of line 4042-4043 the multi-label classifier is less precise than for the other 12 contingencies.

Further analysis shows that solver-determined binding contingency sets $\mathcal{S}_i$ and classifier-predicted binding contingencies sets $\mathcal{C}_i$ are identical for 1496 out of 2431 test samples. More importantly, when solving the PSCOPF with only the classifier-predicted binding contingencies and then applying SSSA to check security, we find that for about 84% of the test samples (i.e. 2032) the resulting PSCOPF solution is actually secure with respect to all 52 postulated contingencies; among the remaining 399 cases (about 16%), 280 are found to be insecure only with respect to a single postulated contingency, while 119 (less than 5%) are found to be insecure for 2 (mostly) or more plausible contingencies.

Let us also notice that in our study the thermal line constraint violations were the most common cause of insecure classifier-assisted PSCOPF solutions. For instance, the thermal constraint for transmission line between buses $\{2031, 2032\}$ was violated in 100 samples, and 96 violations for the line

[2]The precise definition of these accuracy indicators can be found at https://en.wikipedia.org/wiki/Phi_coefficient.

connecting buses $\{4022, 4031\}$ were detected by the SSSA module. Besides, voltage violations were detected in classifier-assisted PSCOPF solver for 23 samples, with 20 corresponding to those at bus 4062.

IV. Conclusions

This letter proposes for the first time to use supervised machine learning in order to construct an oracle to predict binding contingencies for the AC-PSCOPF problem.

Our preliminary case study on the classical Nordic-32 benchmark shows that this may be addressed by training a single multi-label neural-network classifier with two hidden ReLU layers and sigmoid activation functions in the output layer. Our results also showed a very good level of accuracy in terms of binding contingency prediction capability. While we considered in the empirical study of this work only single-outage contingencies, our method could also be applied to multiple outages to the same extent as classical PSCOPF solvers can do this. Yet, novel research directions need to be pursued to investigate computationally-demanding scenarios resulting from numerically larger values of N and k, wherein scalability emerges as a core challenge.

In our investigation, we however covered in our training and test samples only variations in terms of active and reactive demand patterns. It remains a challenge to leverage machine learning so that it can take into account all relevant variations in the system topology (grid, generation, substations), and ideally even the expected variabilities in the cost-function terms, the line ampacities, and other constraints of the PSCOPF [5], that can arise seasonally and within an operating hour. Further work is clearly needed to address all these issues.

Perhaps, a potential solution is to leverage graph neural network (GNN) techniques to account for topological variations in the grid [9]. By employing GNN techniques, classifiers could harness their inherent ability to model and reason about complex spatial and physical relationships in data, which is particularly useful for addressing power system optimization and control.

References

[1] O. Alsac and B. Stott, “Optimal Load Flow with Steady-State Security,” *IEEE Trans. PAS*, vol. PAS-93, no. 3, pp. 745–751, 1974.

[2] F. Capitanescu, M. Glavic, D. Ernst, and L. Wehenkel, “Contingency Filtering Techniques for Preventive Security-Constrained Optimal Power Flow,” *IEEE Trans. Pow. Syst.*, vol. 22, no. 4, pp. 1690–1697, 2007.

[3] L. Platbrood, F. Capitanescu, C. Merckx, H. Crisciu, and L. Wehenkel, “A Generic Approach for Solving Nonlinear-Discrete Security-Constrained Optimal Power Flow Problems in Large-Scale Systems,” *IEEE Trans. Pow. Syst.*, vol. 29, no. 3, pp. 1194–1203, 2014.

[4] Z. Yan and Y. Xu, “A Hybrid Data-Driven Method for Fast Solution of Security-Constrained Optimal Power Flow,” *IEEE Trans. Pow. Syst.*, vol. 37, no. 6, pp. 4365–4374, 2022.

[5] N. Popli, E. Davoodi, F. Capitanescu, and L. Wehenkel, “On the robustness of machine-learnt proxies for security constrained optimal power flow solvers,” *Submitted for publication, https://hdl.handle.net/2268/304498*, 2023.

[6] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: Synthetic Minority over-Sampling Technique,” *Journal of Artificial Intelligence Research*, vol. 16, no. 1, p. 321–357, 2002.

[7] I. Tomek, “Two Modifications of CNN,” *IEEE Trans. on Syst. Man. Cybern.*, vol. SMC-6, no. 11, pp. 769–772, 1976.

[8] G. E. Batista, R. C. Prati, and M. C. Monard, “A study of the behavior of several methods for balancing machine learning training data,” *ACM SIGKDD explorations newsletter*, vol. 6, no. 1, pp. 20–29, 2004.

[9] B. Donon, “Deep statistical solvers & power systems applications,” PhD Dissertation, Université Paris-Saclay, 2022.