Contingency analysis based on a hybrid machine learning approach

Dimitrios Semitekos, Nikolaos Avouris
Electrical and Computer Engineering Dep. HCI group
University of Patras,
26 500 Rio Patras, Greece.
+30610997349, [dsem, N.Avouris] @ee.upatras.gr

ABSTRACT

In this paper we present an innovative approach for power systems - contingency risk assessment. A number of stand-alone and hybrid machine learning tools are proposed for off-line steady state network operation, making the use of classical load flow studies not necessary. This paper presents an outline of the proposed methodology, the network indices used as well as an interpretation of the experimental results. A hybrid solution that combines a number of machine learning approaches, is demonstrated. It produces contingency predictions with a higher accuracy than the other stand-alone machine learning tools.

Keywords: Contingency Analysis, Decision Trees, Neural Networks, k-Nearest Neighbors and Hybrid Machine Learning Tools.

INTRODUCTION

Contingencies are defined as potentially harmful disturbances for the steady state operation of an electrical network. A contingency may comprise the loss of any combination of network elements. The best way to assess the risk that a contingency may pose to a network operating point (OP) is by performing a load flow analysis after applying the contingency to the network OP. The price for such an assessment is a long computing time. An alternative solution adopted in this paper is through the implementation, through a process that involves training and testing, of machine learning tools such as decision trees, neural networks, *k*-nearest neighbors and hybrid systems. All experimentation carried out was based on a maximum load - full connectivity snapshot of the electrical network of the Greek island of Crete having 61 buses and 78 lines, named hence "base case" OP.
The contingency study software environment built is an innovative toolkit, also discussed in [9], with modular characteristics facilitating the process. The user can build a warehouse of validated OPs, compute a series of global indices and metrics on them, define a set of contingencies and apply them on a set of validated OPs. Once this is done, the set of the simulated indices and metrics is stored in an OP data warehouse with per contingency results. The indices and metrics refer to pre-contingency OPs.
Any subset of indices and metrics with relevant contingency results can be extracted on a per contingency basis. Then, this data set can be split at random into training and testing sets at any given quota for machine learning training and testing purposes. Once the machine learning tools are trained, the testing set can be used for evaluation. A series of experiments that repeatedly split the data to training and testing sets at random are performed. They are called *n*-fold cross validations and their results are stored in relevant files on a per machine learning tool and per contingency basis. In such a way, through cross validations, the statistical bias of the experiments is eliminated. The mean average (*AVG*) and the sampling standard deviation (*STD*) of the *n*-fold cross validations are good performance and reliability measures.
In what follows a more detailed presentation of the functionality of the developed toolkit is undertaken as well as a presentation of the experimental results.

THE TOOLKIT

A. The Operating Points (OP) Generation Module.

The OP generation module simulates a number of OPs from the base case OP, through variations of the network connectivity (elimination of lines and buses), uniform reduction of active and reactive load and variations of generation schemes (generators powered at any user defined sequence). This method of OP simulation is proposed in [1]. "*A power system situation is mainly defined by three parameters: the consumption level, the unit commitment and the network topology.*" Consequently, the number of the simulated OPs is equal to the Cartesian product of C_i x L_j x G_k,, where C_i are network connectivity variations, L_j load level variations and G_k generation variations respectively. (If line limit

or voltage violations are reported, then the valid simulated OPs are fewer). Valid OPs and relevant network wide indices and metrics are stored in a data warehouse.
The generation scheme is simulated in such a way that the generators are operated close to their economic operational level. An initial estimation of the network wide generation level is calculated on the load level basis, while slack bus surpluses or deficits are considered at a re-scheduling load flow cycle.

B. The contingencies definition and processing module

The contingencies are defined manually in two separate line and bus contingency data files. The line numbers of these files are in correspondence. Every line of these files contains the number of elements that will be removed, followed by a bus, or line detailed description respectively.
Contingencies are further processed being applied to all valid OP produced in the OP generation module. This is achieved through a network load flow solution. The final outcome of a contingency applied on an OP is classified into three major categories codified by (1) for innocent-non violating contingencies, (2) for violating contingencies and (3) for seriously violating - non load flow solution converging contingencies.

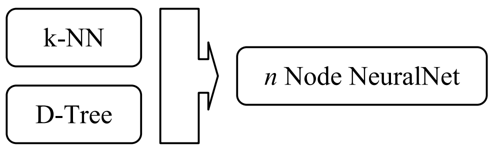
C. The features selection module

In the OP Generation module a number of network wide measures and metrics are calculated on a per simulated OP basis. In this module these measures and metrics are considered as candidate machine learning tool training and testing features. The user can select any number of them. There are simple aggregate features such as the active and reactive power generation and load, the number of lines and buses and the cumulative rating line limit. More elaborate features such as the active and reactive power margins are later discussed in detail. The selection of high quality features is very important for decreasing the machine learning tool overhead and improving the predictability. In order to reveal and rank the most salient features for a series of defined contingencies statistical correlation techniques have been applied. The findings, discussed in detail in this paper, proved that the most salient features were barely the same for all contingencies. Once the features are selected, they are stored in text files on a one contingency per file basis. The last field of every such file contains the classified result of the contingency applied to a specific OP.

D. The Randomization - Machine Learning Tool Training and Testing Module.

Here the files with the selected features are further split at random to training and testing sets at user defined quotas. The work done here is actually a random per line partition of the files produced in the features selection module to corresponding training and testing set files.
Subsequently decision trees, feed-forward one hidden layer networks and *k*-nearest neighbor tools are trained at will. Then the trained machine learning tools are tested with the testing data set. Finally, 3x3 confusion matrices are created. Each row and column covers the “safe”, “violating” and “serious violating - non-load flow converging” classes. Rows refer to predicted values while rows refer to actual values.
The hybrid machine learning tool that is proposed in this article is a composite tool. It yields high score predictions, combining decision trees, k-nearest neighbors and neural networks, as depicted in fig. [1].

Fig. [1] The proposed hybrid machine-learning tool



This hybrid machine-learning tool is based on the following principles. Once the decision trees and the *k*-nearest neighbors are trained and tested a common file is tabulated per contingency with the following normalized field structure: *k-nearest neighbor prediction, decision tree prediction, actual load flow contingency value.* The number of records of this file is equal to the number of the test cases. Normalized predictions are codified as 0.1, 0.2 and 0.3 for the “innocent”, “violating” and “serious violating - non-converging” cases, while the actual load flow outcome is codified as “1 0 0”, “0 1 0” and “0 0 1” respectively. This file is further split to neural network training and testing subset quotas. Separate files are used for every contingency. Once the training of the neural network is complete, its predictability is tested with the testing subset of the initial testing set. The neural networks so implemented actually play the role of an arbitrator for cases where the *k*-nearest neighbors and the decision trees do not manage to make correct predictions. The input node data set of the synergetic neural network is actually comprised of $3^2 = 9$ ordered pairs while all possible input and output neural network ordered pairs are $3^3 = 27$ cases.

NETWORK WIDE INDICES AND METRICS

It this section a discussion on the network features proposed and used in this study is included. There have been various elaborate network wide indices and metrics proposed by other authors that were here implemented as machine learning tool input features.

1. The real power margin index and reactive power margin index are generally given by equation [1] (ref. [2], [3], [4], [5]) and equation [2] (ref. [6]).

Equation 1. The real power margin index

$$PI = \sum_{i=1}^{L} W_i \left(\frac{P_i}{\hat{P}_i} \right)^{2n}$$

P_i is the real power of the branch *i*, $\hat{P}_i$ is the branch real power flow limit, W_i are weighting coefficients and *L* is the number of branches.

Equation 2. The reactive power margin index

$$QI = \sum_{i=1}^{NB} \frac{W_{V_i}}{2n} \left(\frac{|V_i| - |V_i^{sp}|}{\Delta V_i^{Lim}} \right)^{2n} + \sum_{i=1}^{NG} W_{Q_i} \left(\frac{Q_i}{Q_i^{Max}} \right)^{2n}$$

V_i and ΔV_i^{Lim} are bus voltages and limits, Q_i is the reactive power produced at bus *i*, Q_i^{Max} is the reactive power production limit, *NB* is the number of buses, *NG* is the number of generating (reactive) production and Q_{Qi} is a real non-negative weighting factor.

For our study six similar indices were implemented with weights all set equal to one. The first three indices were based on equation [1]. For the first index the *2n* exponent was set to 1. For the second and third the values of *n* where set to 1 and 2. The names of these indices were *PMarg1, PMarg2* and *PMarg4*. The last three indices, namely *QMarg1, QMarg2, QMarg4*, were based on the second part of equation [2] and their exponent values were calculated as for the previously three real power indices. We should notice that statistical correlation methods applied indicated all six indices as the best input feature predictors. (Experimental results are further provided).

For the calculation of $\hat{P}_i$ and $\hat{Q}_i$ values an assumption has been made, due to calculation difficulties and missing data. ($\hat{S}_i$ apparent power limits were only provided). So, $\hat{P}_i$ and $\hat{Q}_i$ maximal values were calculated through the power triangle equation $\hat{S}_i^2 = \hat{P}_i^2 + \hat{Q}_i^2$ and the assumption that $\frac{P_i}{Q_i} = \frac{\hat{P}_i}{\hat{Q}_i}$. Solving these equations we have that $\hat{P}_i = \sqrt{\frac{\hat{S}_i^2}{1 + \frac{Q_i^2}{P_i^2}}}$ and

$$\hat{Q}_i = \frac{\hat{P}_i \cdot Q_i}{P_i}$$

2. The relative dispatch coefficient index. This index is proposed by L. Wehenkel: " *a voltage stability index is computed (the sensitivities of the total reactive power generation to a reactive power consumption), known as 'reactive power dispatch coefficients'.*" [7]

Equation 3. Relative dispatch also called Voltage Stability Index).

$$VStabIndex = \frac{\sum_{Generators} Q_{GENERATED}}{\sum_{Loads} Q_{LOAD}}$$

3. PV curves relate the voltage at a load bus to the active load delivered. PV curves are used as a method of voltage stability evaluation for contingencies.[8] The empirical index *PVIndex* (equation [4]), provided encouraging results in previous experiments.

Equation 4. PVIndex

$$PVIndex = \sum_{LoadBuses} (VM - 1) * P_{LOAD}$$

where, *VM* is a load bus voltage in per unit terms and P_{LOAD} the active load of each load bus.

THE EXPERIMENT

The contingency study environment we built is network independent. For our experiments we used the network of the Greek island of Crete (peak load 1998 data). For the OP generation twenty line and bus outage scenarios were considered, (each scenario covering one to three lines outages), 5 reduced load levels rating from 100% to 70% of the peak load and six different generation scenarios. The simulated OPs totaled 20x5x6=600, 287 out of which violation free OPs were found, suitable for the study of contingencies.
A total of 7 contingencies were defined, tabulated in table [1]:

Table [1]. The defined Contingencies

Ct#	Line Outages (Number of Outaged Lines, FromLine#, ToLine#, ...)
1	1, 2, 4
2	1, 6, 11
3	1, 6, 8
4	1, 10, 16
5	2, 2, 4, 10, 16
6	2, 2, 4, 6, 8
7	2, 6, 8, 6, 11

Once the contingencies were applied on the validated 287 OPs, the network wide indices and metrics were statistically evaluated through correlation procedures. The six most significant features per contingency were ranked and marked on a 1 to 6 scale according to their importance. The results are depicted in table [2].

Table [2]. The six most significant features per contingency as found statistically. (The higher mark is, the more significant a feature is).

Network Wide Index	Contingency #							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**Σ**
1. Number of Lines			5					**5**
2. Σ (Line Rating)			6					**6**
3. Σ (Active Load)			1					**1**
4. Σ (Act. Generation)			2					**2**
5. Σ (React. Load)								
6. Σ(React. Gen/tion.)	2				2	2		**6**
7. VoltStabIdx (eq.[2])								
8. Σ (Line S Flow)								
9. (8) / (2)					3			**3**
10. Σ (Active Losses)								
11. PVIndex (eq. [3])								
12. PMarg1	5	4	4	6	5	6	4	**34**
13. PMarg2	1	3		3	1	1	3	**12**
14. PMarg4	6	6		4	6	5	6	**33**
15. QMarg1	3	2	3	5		4	2	**19**
16. QMarg2		1		1			1	**3**
17. QMarg4	4	5		2	4	3	5	**23**

Trying to interpret the findings we conclude that the real and reactive margins are proved to be the best network estimators. The results seem to be diverging for contingency 3. Having a closer look at the definition of contingency 3 we see that it is a rather harmless contingency. Indeed, the confusion tables of all machine-learning tools show that for the majority of the test cases the predicted and actual result is "innocent". For this reason, statistical findings are rather confusing. (As we will see later on, contingency 3 is the only contingency for which the hybrid approach yields worse predictions than any classic one).

After ranking the candidate training features, a wider set of features was selected for a broader experiment, covering a total of eleven elaborate indices, namely indices number 2, 7, 8, 9, 11, 12, 13, 14, 15, 16 and 17 of table [2].
The 287 operating point features with their load flow outcomes were split on a per contingency basis to training / testing quotas (70% / 30%) yielding a total of 201 training OP features and 86 testing OP features. Then they were split at random to training / testing cases twenty times over, yielding 20fold cross validations. For every cross validation all decision trees, 12 nodes 1 hidden layer feed forward neural networks, 1-nearest neighbor, and 10 nodes 1 hidden layer synergetic hybrid (1-NN - d-tree) neural networks were trained and tested with the same training / testing input feature data set. For the synergetic hybrid 1-NN - d-tree neural network solution the testing set was further split to training / testing quotas (70% / 30%) at random, yielding 60 training and 26 testing cases (accounting the total of the initial 86 test cases).
The results of the experiments are illustrated at table [3].

Table [3]. Decision Tree, 12 Nodes 1 Hidden Layer Feed Forward Neural Network, 1-Nearest Neighbor, 1-NN - D-Tree to 10 Nodes 1 Hidden Layer Feed Forward Neural Network Synergetic Hybrid *successful prediction percentages*.

Cont	D-Tree	12N Neural	1-NN	Hybrid
1	82.21%	78.60%	83.78%	93.46%
2	78.72%	68.61%	80.64%	87.31%
3	93.66%	94.48%	95.00%	91.92%
4	78.20%	66.80%	83.02%	100.00%
5	81.40%	73.84%	81.40%	95.38%
6	83.78%	81.05%	85.35%	92.69%
7	77.67	69.19	78.49	93.27
AVG	**82.23**	**76.08**	**83.95**	**93.43**

Examining closer table [3], we observe that the hybrid solution exhibits the strongest predicting power. Then, the one nearest neighbor (*1-NN*) tool provides slightly better results than the decision trees (*d-trees*), while the neural network tool, (despite the extremely long training times required), exhibits the lowest predicting power. For contingency number 3, for the reasons mentioned in the paragraph discussing the results of table [2], the successful predictions rate is reversed: the 12 node neural network tool achieves better predictions than the rest, while the hybrid tool is doing worse than all others.

RELIABILITY ASSESSMENT

Accurate predictions is the prime performance measure that is expected from a machine-learning tool. Reliability, however, is equally important. As a measure of reliability for a set of 20fold cross validations we propose the calculation of the sampling standard deviation as depicted in table [4].

Equation 5. Sampling Standard Deviation

$$STD = \sqrt{\frac{n\sum_{i=1}^{n} x^2 - \left(\sum_{i=1}^{n} x\right)^2}{n \cdot (n-1)}}$$

Table [4]. Sampling Standard Deviation (equation [5]) results for the 11 feature - 20fold cross validation experiment. Figures refer to deviation observed over successful prediction percentages (n=20), (x is a percentage).

Cont	D-Tree	12N Neural	1-NN	Hybrid
1	6.12	2.65	5.08	5.45
2	5.32	5.56	5.26	5.87
3	3.01	2.58	3.09	5.28
4	4.08	3.98	3.53	0
5	4.51	5.33	3.88	4.25
6	3.23	4.51	4.12	5.42
7	4.09	4.11	4.85	4.48
AVG	**4.34**	**4.10**	**4.26**	**4.39**
STD	**1.10**	**1.17**	**0.83**	**2.02**

The lower the standard deviation is for a set of 20fold cross validations, the most stable the prediction rate is considered to be. Table [4] reveals relatively low standard deviation for all contingencies and the different machine learning tools.
If we try to rank both the predictability and the reliability simultaneously for the machine learning tools implemented using the set of the seven contingencies of our experiment, we could rank table [3] from 1 to 7 weighting more the contingency with the highest prediction rate. For table [4] we weight more the contingency with the lowest standard deviation. Doing so, table [5] is created.

Table [5]. (**P**) Predictability and (**D**) Standard Deviation Ranking

	D-Tree		**12N Neural**		**1-NN**		**Hybrid**	
Cont	**P**	**D**	**P**	**D**	**P**	**D**	**P**	**D**
1	6	1	5	6	5	2	5	2
2	5	2	2	1	2	1	1	1
3	7	7	7	7	7	7	2	4
4	2	5	1	5	4	6	7	7
5	3	3	4	2	3	5	6	6
6	4	6	6	3	6	4	3	3
7	1	4	3	4	1	3	4	5

A high score prediction (**P**) with a minimal sampling standard deviation (**D**) is the desired performance while a low predictability with a maximal sampling standard deviation is the worst case. All cases found in the middle can be considered just average cases where prediction and standard deviation are in accordance. Keeping this in mind we can rank both P (Predictability) and D (Deviation) columns of table [5] by calculating and summing the absolute values of the P and D differences. Results are illustrated in table [6].

Table [6]. Absolute value differences of predictability and standard deviation contingency rankings of table [5]

Cont	D-Tree	12N Neural	1-NN	Hybrid
1	5	1	3	3
2	3	1	1	0
3	0	0	0	2
4	3	4	2	0
5	0	2	2	0
6	2	3	2	0
7	3	1	2	1
SUM	**16**	**12**	**12**	**6**

From table [6] we see that the hybrid machine learning tool reaches the lowest sum of differences, so exhibiting the best behavior in what concerns an increasing predictability in as well as a gradually decreasing standard deviation.
Decision trees account a higher sum of differences, meaning that a high degree of predictability is not always in accordance with a reliable lower standard deviation level. 1-NN and neural networks seem to achieve similar scores. Trying to carry on our further the discussion, from table [4] we can see that though the average deviation for neural

networks is in some degree less than the average deviation for 1-NN. Bearing this in mind in combination with the considerably higher prediction rate of 1-NNs we shall have better rank them higher than the neural networks even from a reliability point of view.

CONCLUSIONS

In this paper a new hybrid approach in contingency studies has been proposed. It involves the use of network wide indices and metrics, as well as the development and use of a synergetic - hybrid system that achieves high performance scores of predictability and reliability.

The environment built enjoys a modular architecture enabling further possible enhancements in future, while at its current state it provides an attractive experimentation tool for the assessment of the risk that a set of line outages could entail to a given network.

The experiment described in this article is the last in a list of experiments previously conducted, see [9,10,11]. The reported here experiment achieved the best results.

Earlier neural network experiments (described in [11]) provided improved predictions as the number of the hidden nodes increased. Training times however were becoming exponentially prohibitive.

Training times for a neural network are proportionate to the input layer nodes, hidden nodes and output layer nodes. For instance for a neural network of k input layers, l one hidden layer nodes and m output nodes, $k \cdot l + l \cdot m + l + m$ weights are to be calculated for every training cycle.

To the contrary, the training of decision trees and k-nearest neighbors is faster. For all machine learning tools built during our study, the testing times can be considered of real time performance.

Earlier experiments conducted with decision trees and neural networks either with fewer features or with not salient ones (according to correlative statistical analysis), yielded results considerably worse than those reported here, see [11]. In those studies, the average predictability rating achieved was in the range of 55% to 70%.

It should be also noticed that for all experiments conducted the neural networks achieved the lowest predictability. The neural networks require normalized input features, while for decision trees normalization is of no use.

Keeping in mind that the initial purpose of our research has been fast contingency analysis evaluations, we can conclude that the results yielded using our hybrid approach are promising ones. The proposed here approach and hybrid machine toolkit could be a useful tools facilitating contingency evaluation studies.

ACKNOWLEDGEMENTS

The research reported here was partially funded by GSRT and the Public Power Corporation of Greece under the YPER program of research - project “Contingency Analysis of Large Electrical Networks”.

REFERENCES

[1] Cholley P., Lebrevelec C., Vitet S., de Pasquale M., “A Statistical Approach to Assess Voltage Stability Limits”, Bulk Power System Dynamics, and Control IV – Restructuring, August 24-28, Santorini, Greece, 2^{nd} page.

[2] Hatziargyriou N.D., Contaxis G.C., Sideris N.C., “A decision tree method for on-line steady state security assessment”, IEEE Trans. on Power Systems, Vol. 9, Nr. 2, May 1994, p. 1052

[3] Albuyeh F., Bose A., Heath B., “Reactive power considerations in automatic contingency selection”, IEEE Transactions on Power Apparatus and Systems, Vol.PAS-101, No.1 January 1982, p.p. 107-

[4] Mikolinnas T.A., Wollenberg B.F., “An advanced contingency selection algorithm”, IEEE Transactions on Power Apparatus and Systems, Vol.PAS-100, No.2. February 1981, p.p. 608-611.

[5] Lauby M.G., Mikolinnas T.A., Reppen N.D., “Contingency selection of branch outages causing voltage problems”, IEEE Transactions on Power Apparatus and Systems, Vol.PAS-102, No.12. December 1983, p.p. 3899-3904.

[6] Ejebe G.C., Wollenberg B.F., “Automatic contingency selection”, IEEE Transactions on Power Apparatus and Systems, Vol.PAS-98, No.1. Jan/Feb 1979, p.p. 97-109.

[7] Wehenkel L.A., Automatic Learning Techniques in Power Systems, Kluwer Academic Publ., 1998

[8] Yorino N., Harada S., Hayashi K., Sasaki H., “A method of voltage stability evaluation for contingencies”, Proc. Bulk Power System Dynamics, and Control IV , 1998.

[9] Semitekos D., Avouris N., Machine Learning Toolkit for Power Systems Security Analysis”, Proc. Med Power Athens. November 2002

[10] Semitekos D, Avouris N., Power Systems Contingency Analysis using Artificial Neural Networks, Proc. CSIT 2002, Patras, September 2002

[11] Semitekos D. D., Avouris N. M., Giannakopoulos G. B., A Flexible Machine Learning Environment for Steady State Security Assessment of Power Systems, Proc. 6th IASTED Int. Conference on Power and Energy Systems, pp. 190-194, July 2001 Rhodes, Greece, (ed.) P.B. Bourkas, ACTA Press, Anaheim, CA.