A Machine Learning Toolkit for Power Systems Security Analysis

Dimitrios Semitekos, Nikolaos Avouris
Electrical and Computer Enginnering Dep., HCI Group
University of Patras,
26 500 Rio Patras, Greece.
[**dsem**, **N.Avouris**] **@ee.upatras.gr**

ABSTRACT

Machine Learning techniques have been extensively used in Power Systems Analysis during the last years. A Machine Learning Toolkit, i.e. a versatile software environment incorporating multiple inter-operable tools facilitating experimentation, can be a valuable asset during power systems analysis studies. In this paper our experience with building such a Toolkit, incorporating a repository of data (Data Warehouse), is described. A number of machine learning and analysis tools have been built and applied on collected structured data in order to perform steady state security assessment of a power system. The Toolkit (UMLPSE), the data collection process, the analysis performed and the tools used are the subject of this paper.
In particular application of the developed toolkit in contingency analysis is demonstrated. A series of system indices and metrics are stored in the data warehouse describing the effects each contingency is expected to have on each operating point. The system indices and metrics are subsequently filtered through statistical procedures with qualitative measures and criteria to be used as training data for machine learning tools (neural networks and decision trees). The performance indices and the fine tuning of the parameters of these machine learning tools are then considered for the screening and ranking of the contingencies for any given electrical network operating point. It is argued that the proposed approach and tools can be applied to many similar power systems analysis studies.

Keywords: Power Systems, Contingency Analysis, Machine Learning, Steady State Security Analysis,

I. INTRODUCTION

Steady state security analysis aims at assessing the risk a contingency would entail for an electrical network operating at a certain point.
System operators' expertise and even human intuition in many ways are successful at assessing the risk a contingency would pose to a network.

Paper accepted for presentation at the 3rd Mediterranean Conference and Exhibition on Power Generation, Transmission, Distribution and Energy Conversion MED POWER 2002, jointly organized by National Technical University of Athens, IEE Hellas, Israel and Cyprus, Athens, Greece, November 4-6, 2002

For instance, an outage of a branch of a two-circuit line of a high connectivity network, operating at low load levels, might not be a threat for the network. To the contrary, an outage of a single-circuit, high voltage, main grid line of a heavily loaded network could be fatal.
From this point of view, there are varying consequences associated to possible contingencies, depending on the electrical network operating state.
Contingencies of an electrical network may involve outages of various network elements, including unscheduled line and bus, generator or transformer outages.
An innovative machine learning toolkit, called UMLPSE (Unified Machine Learning Power Systems Environment), has been build to facilitate power systems security analysis and in particular contingencies estimation. UMLPSE has a modular structure, combining independent power flow and machine learning tool packages. It can be applied for the automatic contingency risk assessment. It can also be used as a benchmarking tool for selection of contingency analysis indices and metrics and experimentation with building automatic learning tools. This machine learning toolkit, described in this paper, may be of interest to EMS operators, to be used as a contingency analyser, as well as to power systems–steady state contingency analysis researchers, to be used as a modular user-friendly experimentation environment, as it enables the user to further proceed and experiment with the fine-tuning of various power and machine learning parameters.
The presentation of the toolkit is done through a contingency analysis case study. The power system involved in this study is the network of the Greek island of Crete, made of 61 buses and 78 lines.
In the following sections, a detailed description of the UMLPSE features and an illustrative presentation of its use is made, followed by a discussion on the analysis findings of the discussed case study.

II. THE UMLPSE

A. The operating points creation module

According to Mitchel [1], a computer program is said to learn from experience *E* with respect to some class of tasks *T* and performance measure *P*, if its performance at tasks in *T,* as measured by *P*, improves with experience *E*. In our case, which concerns building a machine learning toolkit for power systems analysis, the task *T* is that of the contingency risk assessment. The performance measure *P* is the percent of the correct contingency classifications, while the training experience *T* refers to the knowledge the machine learning

tool can acquire applying the set of contingencies under study to a diversified large set of operating points.

An important issue in any machine learning problem is that of data collection. The acquisition of a set of operating points can be directly done from the network through the SCADA subsystem. If so, however, the operating points risk to be rather uniform and not diversified, with certain operating points over-represented and some other under-represented, or even missing.

So a more effective technique for studying contingencies over a more extensive operating points data set is to define operating points through a simulation (load flow studies). The simulation of the operating points set itself, is an attractive idea as the operating points acquisition seems to have been a tedious task in many cases, as also discussed in [2].

Further more, a machine learning toolkit for power system security analysis is more functional when it can produce itself the required operating points. An Operating point can be defined by the load level, the unit commitment and the network topology [3].

In UMLPSE a similar strategy is adopted. The operating points are simulated from a single maximum load - full network connectivity base case operating point, on which connectivity C_i, load L_j and generation sequence - level G_k scenarios are applied. For reasons of simplicity, connectivity scenarios include the removal of a set of buses and lines at a time, load scenarios cover the uniform power load reduction, while the generation sequence scenarios aim at a simulated sequential commitment of generators. Generators are committed and operated according to economic dispatch criteria, powered close to their operational optimal limits. For this reason load flows are executed also defining the reactive power production[1]. The number of the operating points so produced is less or equal to the Cartesian product of C_i *x* L_j *x* G_k. This is because the operating points simulated are subsequently screened for overload and voltage violations. Island producing operating points, as well as operating points for which the power flow algorithm exhibits poor converging behaviour are also discarded.

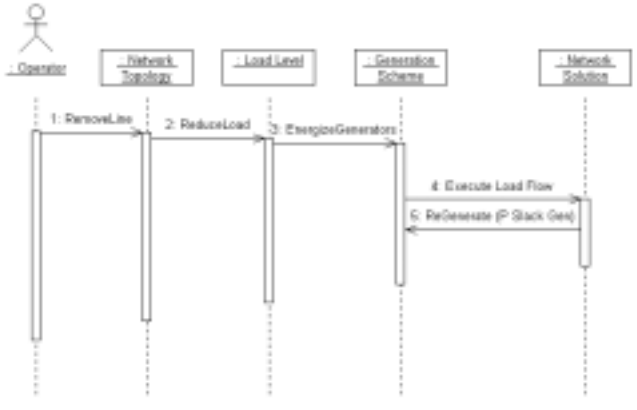


Figure1. UMLPSE OP creation module UML sequence diagram

[1] For all load flows calculations in UMLPSE the *PCFLO* package[4] is used.

The UMLPSE operating points creation module can be schematically described in a UML (Uniform Modelling Language) sequence diagram [5] as illustrated in figure1 and figure2.

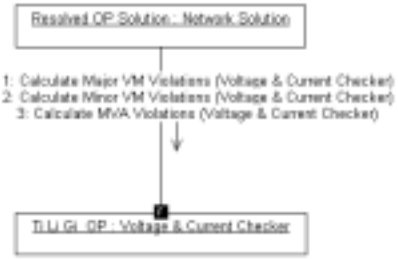


Figure 2. UMLPSE OP Violations Screening Collaboration Diagram

B. The contingencies definition and processing module

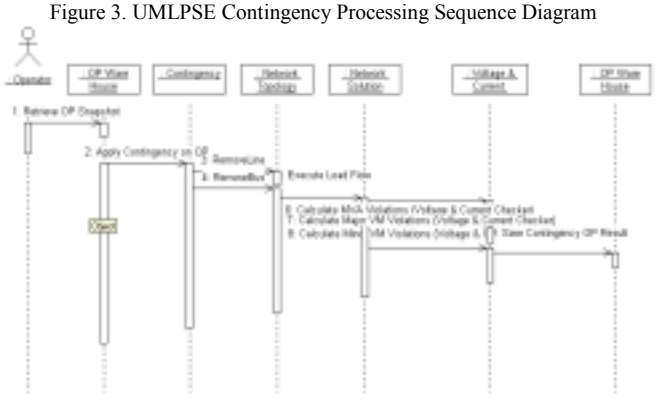
The contingencies examined in MLPSE cover branch and bus outages only. They are stored in relevant *contigency files*, one for the bus and one for the line outages respectively, in a contingency per line format.

The contingencies processing module applies all studied contingencies to all valid operating points stored in the operating points data warehouse.

The contingencies, once applied to the operating points data warehouse, are screened for violations as in the operating points creation module through load flow studies. The outcome of each contingency is classified in three discreet states: "1 0 0" denoting contingencies causing serious violations, forming either islands or leading to non converging power flow study, "0 1 0" denoting contingencies with limited violations and "0 0 1" denoting innocent - violations free contingencies.

The contingencies definition and processing module in UML terms is depicted schematically in the sequence diagram of fig. 3

Figure 3. UMLPSE Contingency Processing Sequence Diagram



C. The data warehouse module

The *Data warehouse module* is accessed by to the *Operating points creation module*, the *Contingencies definition* and

processing module and the *Features selection module* to be described in the following subsection.

The Data warehouse module consists of the operating points detailed data as computed in the Operating point creation module and the Operating point abstraction sub-module containing macroscopic information about every simulated operating point. The abstraction sub-module contains data related to the simulated OP connectivity C_i, L_j, G_k, the overall load level, the number of lines of the network, the total rating of the lines, the active and reactive generation and load and other indices, metrics of qualitative measures, that are later described.

The abstraction sub-module is the actual data warehouse of the various attributes used for the subsequent training of the machine learning tools. Though contingency-related information and power-flow results are also hosted in the abstraction sub-module, all attributes stored refer to pre-contingency operating point data.

All data of the abstraction sub-module refer to pre-contingency measurements; this does not constitute a problem as the UMLPSE machine learning tools are trained on a per-contingency basis.

Other researchers have also used pre-contingency figures in machine learning applications:

The total real and reactive demand, pre-contingency real and reactive line flows, and pre-contingency terminal voltage of the contingent element are selected as input features for training of the neural network in [6], while in the problem examined by [2] the objects are pre-fault operating states and points.

The interface of the user or other modules to the data warehouse module is through SQL statements, a reliable and time optimized approach.

D. *The features selection module*

The features selection module aims at the selection of a number of features stored in the data warehouse module for the subsequent training of the machine learning tools. Features determination is not of premium importance for machine learning tools like decision trees, as the selection of the most important features can be part of the Decision tree training process. However this is not the case for other machine learning approaches, like neural networks. So while decision trees, using information entropy can discover the most salient features, neural networks will demonstrate a degrading performance for every redundant feature selected, while training times increase exponentially. It is essential to reduce the number of inputs to a neural network and to select the optimum number of inputs which are able to clearly define the input-output mapping[7]. One of the features that is always forcibly selected is the result of the contingency applied on every valid operating point in the data warehouse. This is because this feature is necessary for training and testing purposes of the UMLPSE machine learning tools.

E. *The features randomisation - splitting module*

In this module the previously selected features are split at any ratio selected by the user to the training data set and the testing data set, being stored to respective data files. The data are also further processed being normalised. Normalisation of data is necessary when neural networks are to be trained. Features are split at random. Once this is done, the user can proceed to the following UMLPSE modules. This module is very important as it enables the user to plan multiple cross validations, by splitting the same data to training and testing sets repetitively.

F. *The machine learning tool training module*

The Machine Learning Tool training module is split in two discreet modules: the decision tree and the fully connected feed-forward neural network training modules. If the neural network module is selected, the user has first to select the number of the hidden layer neurons. Within this module all necessary parameterisation and initialisation of the machine learning tools is executed automatically according to the selected features, as well as the training of the tools themselves.

The machine learning toolkit used in UMLPSE is the DMSK (Data Miner Software Kit) toolkit.[7] Its high modularity enables the addition of more automatic learning tools with practically no programming cost. In figure 4 the functions of the last three modules described are depicted in UML terms.

Figure 4. UMLPSE Automatic Tools Training Sequence Diagram

G. *The machine learning tools testing module*

The UMLPSE testing module mimics a run-time predictor that could operate on real data as received on line in a power station control room, once the selected features are computed. In UMLPSE, the test features data (produced in the UMLPSE features randomisation – splitting module) play the role of the control room.

Despite the time required for the training of machine learning tools (and especially neural networks) even with a rather constraint number of features, the contingency analysis classifications computed in the UMLPSE module are performed in no-noticeable for the user time even using a low performance personal computer for our test case.

Once the predictions of contingency effects are computed for every selected feature row of the testing data set, which represents a power system operating point and for all contingencies, they are compared with the actual power flow simulated results and a performance confusion matrix is tabulated.

For a certain testing set of features both decision tree and neural network confusion matrices can be computed and saved / appended to files, including a "simple" and a "quality" performance index computed for every confusion matrix.

The confusion matrices have three rows and three columns. A CM_{ij} confusion matrix element represents the number of test cases for which the prediction has been *i,* while the power flow - actual result has been *j*. (*i* and *j* may take the values of "1 0 0", "0 1 0" and "0 0 1", classifying a contingency as explained in the previous section). The prediction success rate (the simple index), of the confusion matrix, is the percentage of the sum of the diagonal elements of the confusion matrix over its total population. The "quality" index is an improved index that distinguishes between severe and non-severe misclassifications. It is computed as follows:

Equation 1. The "quality" index

$$QA = \frac{\sum_{i=1}^{3} CM_{i,i} + 0.5*\left(CM_{1,2} + CM_{2,1} + CM_{2,3} + CM_{3,2}\right)}{\sum_{i=1}^{3}\sum_{j=1}^{3} CM_{i,j}}$$

III. INDICES, METRICS, QUALITATIVE MEASURES

In UMLPSE a variety of indices, metrics and qualitative measures is computed. Some of them are used to identify the violations occurring during the operating point simulation, while others are computed and used as machine learning training features.

A. Violation Indices

During operating point simulation as well as contingency evaluation, real power generation within permitted limits for the slack bus and other system generation buses is monitored. Measures are also taken for MVA line and Voltage bus violations. Voltage violations over 0.1 p.u. (in absolute values) in any bus are considered major violations, while voltage violations over 0.05 p.u. are summed and divided by the number of violating buses. When the calculated index exceeds 1.5, voltage violations are considered. MVA violations are monitored for any line close to the MVA rating limit. In UMLPSE the following MVA line overload index is proposed

Equation 2. The line overload index

$$LOI = \sum_{i=1}^{NL} \left(MVALine - MVALineLimit\right) \cdot MVALineLimit$$

where: *MVALine* is the apparent power, *MVALineLimit* the power Limit and *NL* the number of lines.

B. Machine learning training feature indices and metrics

There is a very rich literature concerning global network machine indices. In [6] the following Voltage Performance Index is suggested:

Equation 3. The voltage performance index

$$PI = \sum_{i \in LV} (w_i / M)(f_i)^M$$

where f_i is a function of limit violated buses equal to over-limit violations, w_i weights and M an exponent against the masking effect.

Similar performance indices are also found measuring the real power flow:

Equation 4. The power margin index

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

where P_i is the real power of the branch i, k is the outage branch, $\hat{P}_i$ is the branch real power flow limit, W_i are weighting coefficients and L is the number of lines.[8] Similar indices are also suggested in [2], [9], [10], [11], [12].

In UMLPSE a variety of qualitative measures used provided encouraging results, (including low exponent variations of the above mentioned indices). Here are the most important:

1. The relative dispatch coefficient index. This index, proposed by L. Wehenkel, as a voltage stability index measuring the sensitivities of the total reactive power generation to a reactive power consumption, known as '*reactive power dispatch coefficients* [13] .

Equation 5. Relative dispatch coefficient (mentioned in UMLPSE as Voltage Stability Index).

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

2. 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[14]. An empirical index applied in UMLPSE, described in equation 6, gave encouraging results.

Equation 6. PVIndex [13]

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

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

IV CROSS VALIDATIONS AND RESULTS

UMLPSE can serve as a testing toolkit for contingencies as well as for testing alternative network indices and automatic learning tools. In what follows an illustrative example is given describing experiment results of a typical UMLPSE use case.

In this case 5-fold cross validations have been applied. By the term *N*-fold cross validations, we denote *N* - different repetitions of the randomisation - test / training data split, automatic tool training and testing UMLPSE operations, based on the same data set of features. *N*-fold cross validations aim at eliminating bias in data selection.

A. The experiment

The experiment refers to the Greek island of Crete network and as "base case" is taken the peak load of year 1996 - 1997 operating point. Operating point simulation includes 20 connectivity C_i scenarios removing one to three lines and zero to one buses, four L_j load scenarios, uniformly decreasing load from 100% (base case is a maximum load base case scenario) to 70% base case load and six generation G_k scenarios. The number of the simulated operating points equals to $C_i \times L_j \times G_k = 480$. Once screened for violations, 287 operating points (out of 480) suitable for the contingency study are stored in the operating point data warehouse. A total of seven line contingencies are defined as in table 1.

Table 1. The contingencies of the experiment

Contg#	Buses	Lines	From line#	To line#	From line#	To line#
1	0	1	2	4		
2	0	1	6	11		
3	0	1	6	8		
4	0	1	10	16		
5	0	2	2	4	10	16
6	0	2	2	4	6	8
7	0	2	6	8	6	11

In the UMLPSE data warehouse a variety of indices and simpler network metrics are stored such as the network wide active or reactive power generation. The user may select any number of indices and metrics.

For the experiment, the following features are selected for every pre-contingency operating point:

1. The total reactive power generation
2. The voltage stability index (as described in equation 5)
3. The total power flowing in all lines divided by the total power rating limit (in MVA)
4. The PI index (as described in equation 6)

For the total of four selected training features, it is interesting to point out the average per contingency training times, as in table 2. The neural network training time increases exponentially with number of hidden layer nodes used.

Table 2. Four features average per contingency training times

Decision Trees	Neural Networks (One Hidden Layer Fully Connected Feed-Forward NN)		
	3 Nodes	8 nodes	30 Nodes
< 1second	9.29 seconds	58.57 seconds	6 min. and 45 sec.

B. The results

There is variety of factors that influence the quality of predictions in a machine learning environment used for contingency analysis. The ability of the selected features to well describe an operating point of the network is one factor. The qualities of the machine learning tools used is another. A third, even more important factor is the character of the contingency itself. There are "most of the times" innocent and "most of the times" fatal contingencies in reference to the operating points they are applied on. For any such category of contingencies, predictions tend to reach a high score. Contingencies lying in between these two extremities tend to exhibit a behaviour that is more difficult to predict correctly. We believe that through *N*-fold cross validations the UMLPSE researcher has the ability to automatically identify, the most "predictable" and the most "unpredictable" sets of contingencies. So a contingency may be "predictable and innocent", while another one maybe "unpredictable and fatal". The second type of contingencies is of more interest to the researcher. The "predictability" of a contingency can be measured through high score *N*-fold cross validations. How "innocent" a contingency is, can be measured as the percentage of the data warehouse operating points that prove to be "innocent" (when a certain contingency is applied on them), over the whole population of data warehouse operating points.

In what follows, the results of the experiment are presented on an overall averaged per automatic learning tool basis and on a per contingency basis for all averaged *5*-fold cross validations. Simple index and quality index (as described in equation 1) results are mentioned.

1) Overall averaged results per automatic learning tool.

Graph 1. Simple and Quality index results per automatic learning tool

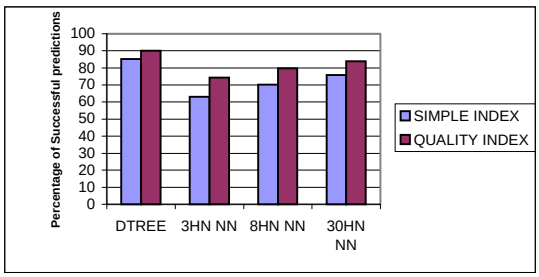


Table 3. Per automatic learning tool Simple and Quality index results

	Decision Trees	One Hidden Layer Fully Connected Feed-Forward Neural Networks		
		3 Hidden Nodes	8 Hidden nodes	30 Hidden Nodes
Simple Index	85.18	63.02	70.13	75.85
Quality Index	90.10	74.24	79.87	83.89

2) Averaged results per contingency.

Graph 2. Simple Index (SI) and Quality index (QI) results per automatic learning tool (DTREE = Decision Tree, HN = Hidden Node, NN = Neural Network).

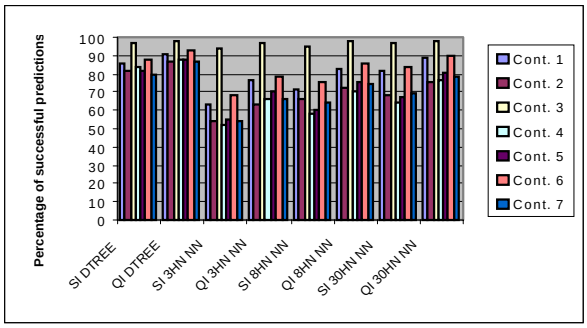
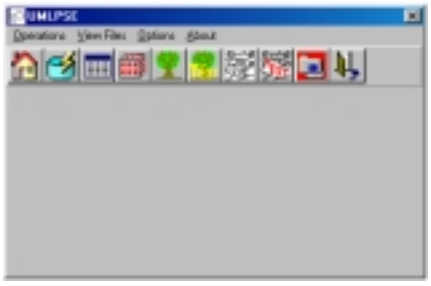


Figure 5. UMLPSE toolkit snapshot



V. CONCLUSIONS

UMLPSE is the result of a long-standing effort to build an automatic, standalone network-independent toolkit for steady state contingency analysis. Additional information on this research can be found in [15].

Though the results achieved are encouraging, there are a number of possible improvements that can still be made. For instance, UMLPSE capabilities can be extended for all network component outages. Also contingency analysis for any given operating point data set, not belonging in the testing data set can be integrated. Towards this direction, a machine learning tool "run - time estimator" that has been implemented serving so far as a stand-alone module is planned to be integrated in the toolkit.

Operating point indices, metrics and quality measures can be further enriched with more versatile, composite, or even user-defined ones.

ACKNOWLEDGEMENTS

The research reported here is partially funded by GSRT and the Public Power Corporation of Greece under the YPER project "Contingency Analysis of Large Electrical Networks".

REFERENCES

[1] Mitchell T.M., Machine Learning, McGraw-Hill Series in Computer Science, 1997, §1.1

[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. 1053, 1056.

[3] 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.

[4] Grady M., PCFLO Version 2.4 User Manual, ECE Dept, The University of Texas at Austin, January 15, 1995. http://www.ece.utexas.edu/~grady/ (*PCFLO Software also downloadable from the site*).

[5] Fowler M., Scott K., UML Distilled: A Brief Guide to the Standard Object Modeling Language (2nd Edition)

[6] Srivastava L., Singh S.N., Sharma J., Knowledge-based neural network for voltage contingency selection and ranking, IEE Proc. – Gener. Trasm. Distrib., Vol. 146, Nr. 6, Nov. 1999, p.650, 651

[7] Weiss, S., Indurkhya N., Predictive data mining, Morgan Kaufmann, 1998.

[8] S. Jadid, M.R. Rokni, "An expert system to improve power system contingency analysis", Electric Power Systems Research 40 (1997), p.p. 37-43.

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

[10] M.G. Lauby, T.A. Mikolinnas, N.D. Reppen, "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.

[11] F. Albuyeh, A.Bose, B.Heath, "Reactive power considerations in automatic contingency selection", IEEE Transactions on Power Apparatus and Systems, Vol.PAS-101, No.1 January 1982, p.p. 107-112.

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

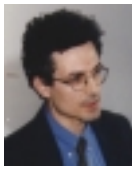
[13] Wehenkel L.A., Automatic Learning Techniques in Power Systems, Kluwer Academic Publ., 1998, p.210

[14] N. Yorino, S. Harada, K. Hayashi, H. Sasaki, "A method of voltage stability evaluation for contingencies", Bulk Power System Dynamics, and Control IV –August 24-28, Santorini, Greece.

[15] Semitekos D. D., Avouris N. M., Giannakopoulos G. B., A Flexible Machine Learning Environment for Steady State Security Assessment of Power Systems, IASTED International Conference on Power and Energy Systems (Euro-PES 2001) July 2001

BIOGRAPHIES

Dimitrios Semitekos Holds a Degree in Statistics and Actuarial science, University of Piraeus, Greece, 1994-1995: MSc in Computing University of Wales, U.K. researcher of the University of Neuchâtel-Switzerland. 1996-, PhD candidate of the University of Patras, Greece. His main research interests include data mining, machine learning, and power systems applications.



Nikolaos M. Avouris, Professor of Software Technology of the Electrical and Computer Engineering Department of the University of Patras, Greece. Research Interests: Human-Computer Interaction, Interactive Systems Design, Distributed Intelligent Systems, application of Knowledge -based techniques in industrial, environmental and educational fields.