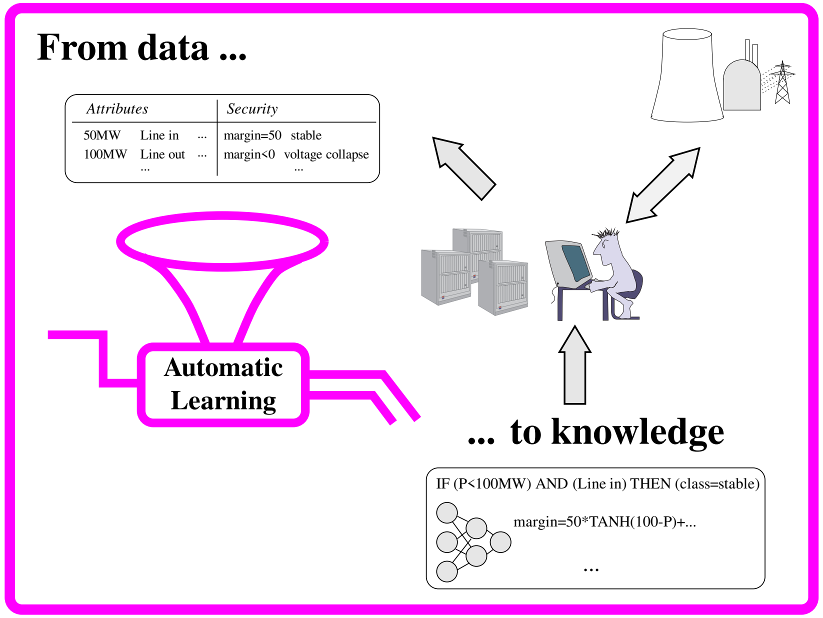
CPSPP'97 - IFAC/CIGRE SYMPOSIUM
ON CONTROL OF POWER SYSTEMS AND POWER PLANTS

TUTORIAL COURSE ON
INTELLIGENT SYSTEMS AND THEIR POWER ENGINEERING APPLICATIONS

AUTOMATIC LEARNING APPLICATIONS TO DSA


From data ...
Attributes
Security
50MW Line in ... margin=50 stable
100MW Line out ... margin<0 voltage collapse
...
Automatic Learning
... to knowledge
IF (P<100MW) AND (Line in) THEN (class=stable)
margin=50*TANH(100-P)+...
...

Louis WEHENKEL
University of Liège - Belgium

Final version of the course notes
July 1997

Contents

Glossary of frequently used acronyms

Below we provide a short list of frequently used terms and acronyms in the context of automatic learning. Note that there is no wide agreement on a standard terminology in this field, since research was carried out in various communities using their own terminology.

Attributes :	input parameters used in the rules extracted by automatic learning
AL :	automatic learning
ANN :	artificial neural network
DB :	data base (comprises all examples, e.g. security scenarios)
DM :	data mining (application of AL algorithms and graphical visualizations to a DB)
DT :	decision tree
KB :	knowledge base (contains expertise extracted by AL)
KNN :	k nearest neighbor rule (a statistical pattern recognition method)
LS :	learning set (subset of examples used as input to AL algorithms)
KDD :	knowledge discovery from data bases
ML :	machine learning (a class of AL methods building symbolic rules)
MLP :	multilayer perceptron (a type of ANN)
1NN :	(one) nearest neighbor rule
PR :	pattern recognition (a class of AL methods, often used as a synonym of AL)
Pe :	test set error rate of a classification rule
RBF :	radial basis functions (a type of ANN)
TS :	test set (subset of examples used to test result of AL methods)
TSE :	total square approximation error of a regression function
TDIDT :	top down induction of decision trees (a generic class of ML methods)

Scope of the tutorial

The material of this tutorial covers two complementary aspects : automatic learning methods (at an intuitive level) and their application to dynamic security assessment (at a more technical level).

In Chapter 3, we will use a simple transient stability assessment example in order to highlight, at an intuitive level, the key concepts while illustrating the use and interest of the different classes of techniques. In particular, in the realm of supervised learning we will discuss three types of methods : decision tree induction, multilayer perceptrons, nearest neighbor. We will also shortly discuss unsupervised learning methods.

In Chapter 4 we review security problems and security assessment environments and how automatic learning can be applied in the different contexts. Then, in Chapter 5, we will discuss how to generate data bases by numerical simulations, which pitfalls to avoid and how to validate automatic learning results. In particular, we will explain how parallel computations may be used to speed up data base generation by exploiting existing computing power. In this part we will call upon examples from various researches on real life large scale power systems.

The bibliography collects some general references on automatic learning for further reading and, without aiming at exhaustiveness, various articles on its application to power system dynamic security assessment relevant to the material presented in the course.

The course does not assume prior knowledge about automatic learning. It assumes some familiarity with dynamic security assessment, in particular transient stability and voltage security.

Finally, let us insist on the fact that is is not possible within a such a short course to cover with the necessary level of detail the very broad topic of this tutorial. We found it useful to provide some of the most relevant details in the notes, and to give hints on good references for further reading. However, in the oral presentation we will skip some of the more technical aspects covered in the notes, for the sake of time and clarity.

1

Historical perspective on automatic learning

The term Automatic Learning (AL) is nowadays used to denote a highly multidisciplinary research field and set of methods to extract high level synthetic information (knowledge) from data bases containing large amounts of low level data. The researchers in the field are statisticians and computer scientists, but also psychologists and neurophysiologists. The latter's aim is mainly to understand human learning abilities, by modeling them through computer algorithms. The former concentrate on the mathematical properties of such algorithms to enable them to solve engineering problems. AL encompasses statistical data analysis and modeling, artificial neural networks, and symbolic machine learning in artificial intelligence. Related work in statistics dates back to Laplace [Lap10] and Gauss [Gau26]. In the field of artificial neural networks, the early attempts were in the 1940's [MP43], while work in symbolic machine learning started only in the mid sixties [HMS66].

In the last two decades, automatic learning has progressed along many lines, in terms of theoretical understanding (see e.g. [Wol94, Vap95]) and actual applications in diverse areas. Probably the main reason for the important breakthrough was the tremendous increase in computing powers. This makes possible the application of the often very compute intensive automatic learning algorithms to practical large scale problems. Conversely, automatic learning algorithms allow one to make better use of existing computing power by exploiting more systematically the information contained in data bases. The availability of very large scale data bases waiting for being properly exploited is thus a very strong, though recent motivation fostering research in automatic learning.

Nowadays, automatic learning methods thus receive routine applications, for example in medical diagnosis, in character recognition and image processing, as well as in financial and marketing problems. Thus, the term Knowledge Discovery from Data bases (KDD) was recently coined to denote the emerging R&D field aiming at developing methodologies and software environments for large scale applications of automatic learning [FPSSU96].

In the context of power system Dynamic Security Assessment (DSA), research on automatic learning started with Pattern Recognition (PR) in the late sixties and seventies [Dy 68, PPEAK74, GEA77].

The idea was, and still is to improve security assessment by combining analytical system theory tools (mainly numerical simulation) with statistical information processing techniques from automatic learning. In this scheme, analytical tools are exploited to screen ranges of security scenarios and build data bases containing detailed security analysis results. Automatic learning methods are then applied to extract relevant synthetic information from these data bases in various forms, in order to figure out under which conditions a particular power system is secure, in a particular context. After careful validation, the extracted knowledge eventually translates into planning and operating decisions.

We will see in the following chapters that this is a very flexible framework, which may be applied to a large diversity of DSA problems. With a proper methodology, the information extracted by automatic learning is indeed complementary to classical system theory methods along three dimensions : computational efficiency (in terms of response time and data requirements); interpretability (in terms of physical understanding); management of uncertainties (in terms of modeling and measurement errors).

In the early attempts, the methodology was essentially limited, on the one hand, by the small size of the security information data bases which could be managed, on the other hand, by the parametric nature of the existing PR methods, which were unable to handle properly the large scale and non linear character of power system security problems. However, since the mid eighties research has accelerated, due to several factors.

First of all, the computing environments became powerful enough to enable the generation of rich enough security information data bases, with acceptable response times.

Second, research in automatic learning has produced new methods able to handle the complexity and non-linearity of power system security problems. In particular, artificial neural networks and machine learning methods have shown their complementary potentials, as reflected by the growing number of publications on their applications to various power system problems (e.g. [PDB85, FKCR89, SP89, ESMA$^+$89, WVRP89, MT91, OH91, Dil91, NG91, HCS94, RKTB94, MK95], to quote only a few ones).

The last - but not least - factor comes from the real interest shown by electric utilities. A few years ago, some of them started ambitious research projects to assess the approach and methods with respect to their own practical needs. This contributed significantly to formalize the application methodology and develop software tools able to handle real, large scale problems. It opened also new research directions and suggested new types of applications.

Today, only a few large utilities in North-America and Europe have actually assessed the approach in the context of their specific power systems. At least one of them - Electricité de France - is presently using the methodology in real field studies. Certainly, some others will start using it in the near future.

Indeed, the fast changes which take place in the organization of power systems imply the use of more systematic approaches to dynamic security assessment in order to maintain reliability at an acceptable level. In the future, power systems will behave more erratically, and at the same time operate closer to their limits. Recent experiences in Western USA and other places around the world, have already demonstrated that present day methodologies reach their limits under such circumstances.

In this tutorial course we aim at showing that the automatic learning framework is indeed a mature and very flexible methodology, which may significantly contribute to improve system reliability, by helping power system engineers to make better use of their computer simulation facilities so as to master the growing complexity of power system dynamic security assessment.

2

Overview of the AL framework for DSA

DSA is a very versatile topic and, although there are many different more or less sophisticated tools, the most widely accepted one is numerical simulation. Planning or operations planning engineers thus use numerical simulation together with their expertise to run some scenarios and extract, the relevant security information to take decisions. Thus, *the purpose of applying automatic learning to DSA is to enhance the existing practice by rendering automatic some of the manual tasks* (selection of scenarios and extraction of relevant synthetic knowledge from the simulation results). The engineers are thereby freed from the most tedious parts and may concentrate on the most interesting ones.

We will see that this approach provides a sound DSA methodology, stating input hypotheses explicitly and extracting reproducible properly validated output results. This enables one to easily repeat a security study with different hypotheses, and adapt the resulting information to changing conditions. It makes also the sharing of information among different people much more straightforward. Thus, in times where human expertise within electric utilities tends to be threatened, the automatic learning framework provides a means to maintain and even enhance it.

Figure 2.1 depicts the general three step framework to apply automatic learning to DSA [Weh97].

Step 1 : Data base generation. The first step of the data base generation consists in specifying the range of scenarios that the particular study will address. While the approach aims at enabling the engineers to carry out more systematic and more global studies, it is generally necessary and even desirable to restrict the focus of a study to an a priori well defined scope.

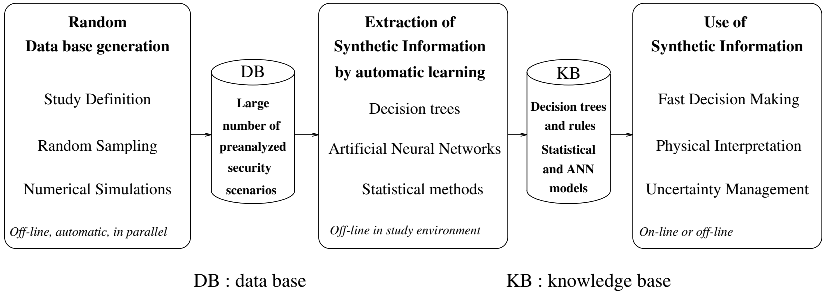


Figure 2.1 *Automatic learning framework for security assessment*

Starting with the existing expertise and problem statement, random sampling specifications are set up, generally through a sequence of discussions among experts in different fields, such as power system dynamics, protections and economic questions. The other part of the data base specification concerns the choice of the parameters which will be extracted from the simulations and stored in the data base.

All relevant security scenarios in a given context are screened and existing numerical simulation tools are exploited - if necessary in parallel - to provide the detailed security information for each one. Note that the data base size and the computation time needed to generate it may vary strongly from one application to another, depending on the type of dynamic model used and the scope of the study. However, to be representative the data bases comprise generally a few thousand simulation scenarios, and in typical large scale applications there may be several hundred parameters.

Step 2 : Application of automatic learning. Application of automatic learning (Data Mining (DM) for short) is the most interesting task. The idea is to exploit various statistical tools in order to "discover" interesting information. We will see that to fully exploit security information data bases it is indeed necessary to combine different automatic learning methods offering complementary features.

At each step of the process the information extracted is validated against an independent test set and against prior expertise and physical knowledge existing about the problem. In particular, the scenarios which lead to dangerous errors may be identified and analyzed in detail. Sometimes, the engineer deems that a new data base should be generated to study some situations not sufficiently well represented in the original one, or to assess the effect of countermeasures suggested by the DM results.

The final step, as in traditional security studies, is report writing and translating the information extracted from the data base into guidelines for operators and planning engineers.

Step 3 : Exploiting extracted information. Anticipating on the following chapters, let us briefly discuss the three dimensions along which this framework may complement classical system theory oriented methods for security assessment.

First of all *computational efficiency*. By using synthetic information extracted by AL, instead of numerical methods, much higher speed may be reached for real-time decision making. Further, in terms of data requirements, whereas analytical and numerical methods require a full description of the system model, the approximate models constructed via automatic learning may be tailored in order to exploit only the significant and/or available input parameters. While computational efficiency was the motivation of the ealry attempts [Dy 68], it remains important even today.

But the synthetic information extracted by AL may itself be complementary to and more powerful than the case by case information provided by analytical or numerical methods. In particular, *interpretability* and the ability to cope with *uncertainties* are important fallouts expected from AL.

Since the first proposal to use AL to provide physical insight into the nonlinear system behavior [PDB85], it has been shown that decision tree induction is indeed an effective way to extract *interpretable* and accurate security rules [WP93b, WVP+94]. We will see such rules express problem specific properties, similarly to human expertise, and hence may be easily appraised, criticized and eventually adopted by engineers in charge of security studies. Moreover, the flexibility of the AL framework allows one to tailor the resulting information to analysis, sensitivity analysis and control applications.

As concerns management of *uncertainties* (dynamic models, external systems, load behavior, measurements. . .) we will provide some illustrations showing how they can be taken into account by AL.

To conclude this introduction, we note that the above motivations for using automatic learning, presented here in the specific context of security assessment, are also relevant in many other applications.

3

An automatic learning tool box

The aim of this chapter is to give a flavor of what automatic learning is all about, and highlight the complementary nature of different methods so as to motivate the tool box approach.

We first define the general *supervised* learning problem and illustrate it by a simple power system transient stability example. Then, we introduce three automatic learning methods, representative of three complementary classes of methods. First, decision trees are presented, able to provide interpretable information. Then multilayer perceptrons are described, able to provide very flexible smooth non linear approximations. We end up with the good old nearest neighbor approach which extracts case by case information from a data base. Although we will stay at an intuitive level, we will introduce key concepts and main technical aspects in supervised automatic learning. We conclude the chapter by briefly discussing hybrid learning methods and unsupervised learning.

3.1 Supervised learning

At an abstract level, the generic problem of supervised learning from examples can be formulated as follows :

> *Given a set of examples (the learning set (LS)) of associated input/output pairs, derive a general rule representing the underlying input/output relationship, which may be used to explain the observed pairs and/or predict output values for any new unseen input.*

In automatic learning we use the term *attribute* to denote the parameters (or variables) used to describe the input information. The output information can be either symbolic (i.e. a *classification*) or numerical.

In the context of security assessment, an *example* would thus correspond to an operating state of a power system, or more generally to a simulated security scenario. The input *attributes* would be relevant parameters describing its electrical state and topology and the output could be information concerning its security, in the form of either a discrete classification (e.g. secure / insecure) or a numerical security margin. This is illustrated by the example below.

3.2 Illustrative transient stability problem

To explain ideas we use a very simple "toy" problem, which is, by no means, meant to be representative of actual large scale applications. Later on we will illustrate some real life application concerns.

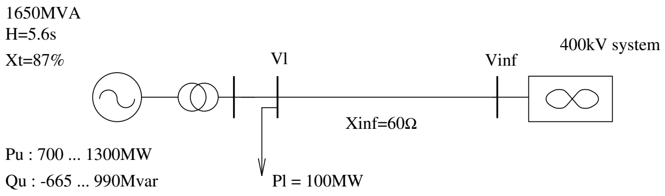


Figure 3.1 *One-machine infinite bus system*

Figure 3.1 depicts a simple One-Machine-Infinite-Bus (OMIB) system, composed of a generator, a transformer, a load connected at the EHV (400kV) side of this transformer, and a reactance (Xinf) representing the equivalent impedance of the EHV network. The generator inertia constant H and the reactance Xt (modeling the transient direct axis reactance, the transformer reactance, and a short line connecting the generator to the EHV substation) are given in p.u. of its nominal MVA rating (1650MVA).

Let us consider that a three-phase short-circuit occurs in the EHV substation close to the generator, normally cleared (without line-tripping) after 155ms, and let us declare the system as insecure if the generator loses synchronism after this disturbance. To determine the degree of security of the system, we will determine the *critical clearing time* (CCT) of this disturbance, which is the maximum time for the short-circuit to be cleared without yielding loss of synchronism of the generator. In other words, we will consider the system to be insecure if the CCT is smaller than 155ms, secure otherwise.

In order to illustrate automatic learning algorithms, we will generate by numerical simulation a data base of examples of diverse pre-fault operating conditions of the OMIB system, determining for each one its CCT and classifying it accordingly into the secure or insecure class. Note that only a part of this data base will be used as a learning set to train the automatic learning methods, the remaining part as a test set to evaluate their reliability.

In the above OMIB system the following parameters may influence the security of the system : the amount of active and reactive power of the generator (denoted by Pu and Qu), the amount of load nearby the generator (Pl), voltage magnitudes at the load bus and at the infinite bus (Vl and Vinf), and also the short-circuit reactance Xinf, representing the effect of variable topology in the large system represented by the infinite bus. However, for the sake of simplicity we assume that only the active and reactive power of the generator are variable, while keeping the voltages Vl and Vinf constant (and equal to 400kV).[1]

We have generated a data base of 5000 such states, by (randomly) sampling Pu and Qu uniformly in the ranges indicated at Fig. 3.1, computing the CCT of each case by a step-by-step dichotomic search, then classifying the states as secure if their CCT is larger than 155ms, insecure otherwise. Figure 3.2 shows the scatter plots, illustrating how Pu and Qu act upon security (security class in the left hand part, CCT in the right hand part).

Learning set (LS). We use a random subsample of 3000 (Pu,Qu) states together with the output security information (class or CCT, as appropriate).

Test set (TS). We use the remaining 2000 states to evaluate the reliability of the different methods.

[1] Admittedly, to be more realistic we could introduce the effect of the other mentioned parameters, Pl, Vl, Xinf and Vinf by making them vary according to appropriate random distributions.

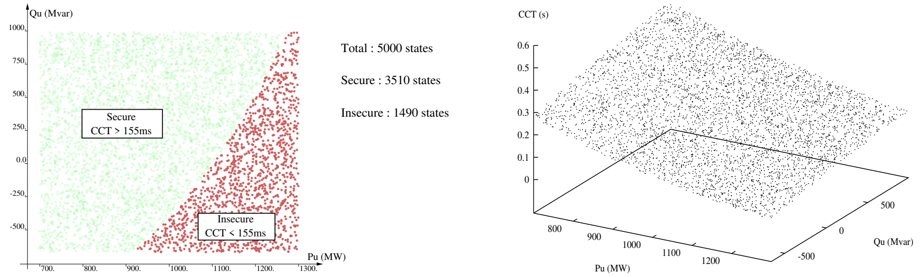
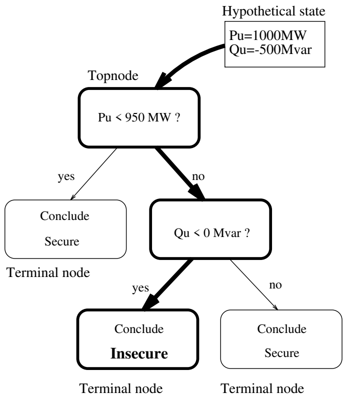


Figure 3.2 *Data base of 5000 random states of the OMIB system*



Equivalent If-Then rules :

Rule 1 : If (Pu < 950MW) then Conclude Secure

Rule 2 : If (Pu > 950MW) and (Qu < 0Mvar) then Conclude Insecure

Rule 3 : If (Pu > 950MW) and (Qu > 0Mvar) then Conclude Secure

Figure 3.3 *Hypothetical decision tree and equivalent if-then rules*

3.3 Symbolic knowledge via machine learning

Machine learning (ML) is a subfield of automatic learning concerned with the automatic design of rules similar to those used by human experts (e.g. if-then rules). We will describe only *Top down induction of decision trees* (TDIDT), which is one of the most successful classes of such methods [BFOS84, Qui93].

Decision trees

Before describing how TDIDT proceeds to build decision trees let us explain what a decision tree is and how it is used to classify a state. Figure 3.3 shows a hypothetical binary decision tree (DT) for our problem using the two attributes Pu and Qu. The bold arrows on the tree suggest how a hypothetical state (Pu = 1000 MW and Qu=-500 Mvar) traverses the tree in a top down fashion to reach a terminal node. One starts at the topnode and applies sequentially the dichotomous tests encountered to select the appropriate successor. When a terminal node is reached, the output information stored there is retrieved. Thus, for our hypothetical state the conclusion is “insecure”. Note that the tree may be translated into an equivalent set of if-then rules, one for each terminal node. E.g. the tree in Fig. 3.3 translates into the rules indicated beneath it.

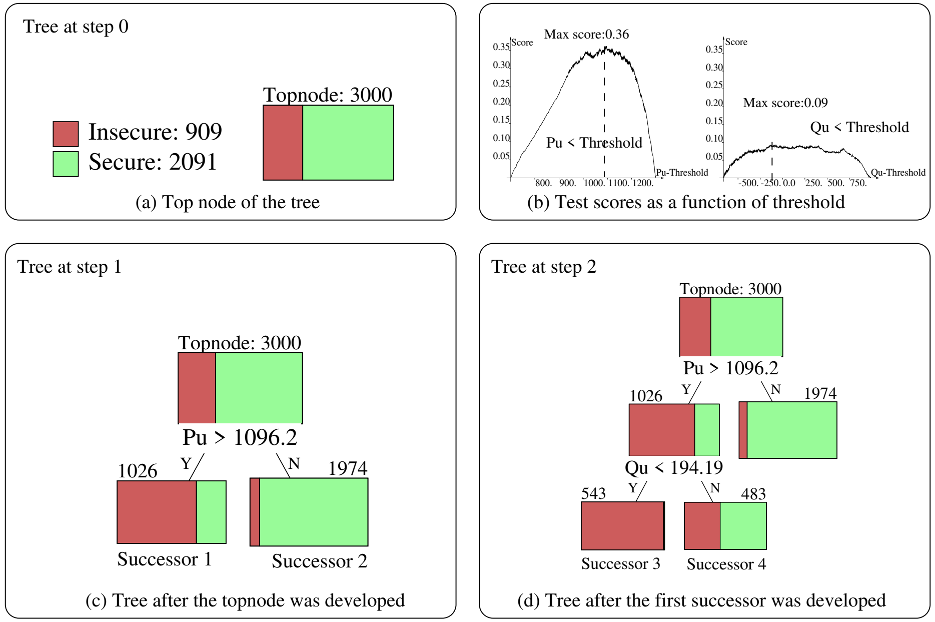


Figure 3.4 *Three first steps of decision tree growing*

Decision tree growing

Now, let us illustrate on our example how the TDIDT method will extract from our learning set a set of classification rules in the form of a decision tree.

Figure 3.4 illustrates the successive node splitting procedure. The procedure is initialized by creating the topnode of the tree, which corresponds to the full LS as shown in Fig. 3.4a. Note that the relative size of the dark and light areas of the box used to represent the topnode corresponds to the proportion of insecure and secure states in the full learning set (909 insecure states vs 2091 secure states).

To develop the topnode each candidate attribute (here Pu and Qu) is considered in turn, in order to determine an appropriate threshold. To this end, the learning set is sorted by increasing order of the considered attribute values, then for each successive attribute value a dichotomic test is formulated and the method determines how well this test separates secure and insecure states, using an information theoretic score measure. The score measure is normalized, between 0 (no separation at all) and 1 (perfect separation). Figure 3.4b shows how the score varies in terms of the threshold both for Pu and Qu at the topnode. Thus, the optimal threshold for Pu is found to be 1096.2 MW (with a score of 0.36) and the optimal threshold for Qu is found to be -125Mvar (with a score of 0.09). Thus the overall best test is identified at the topnode to be Pu>1096.2 MW.

Once the optimal test is found, the next step consists of creating two successor nodes corresponding to the two possible issues of the test; the learning set is then partitioned into corresponding subsets by applying the test to its states. The result of this step is represented at Fig. 3.4c. Note that the number on

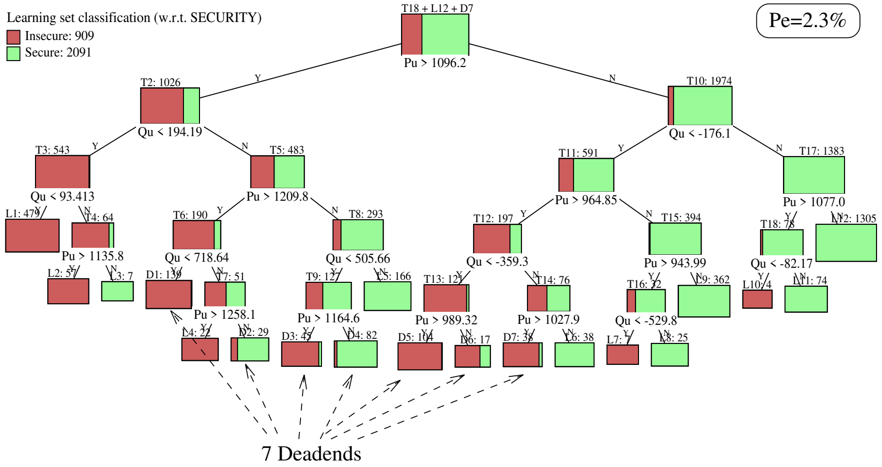


Figure 3.5 *"Orthogonal" decision tree (end result)*

the top of each node represents the number of corresponding learning states : 3000 at the topnode, 1026 at the first successor and 1974 at the second successor. Note also that the first successor contains a strong majority of insecure states, while the second successor contains a very strong majority of secure states.

Stopping to split criterion

As is illustrated on Fig. 3.4d, the procedure continues recursively to split the recently created successors, gradually separating the secure and insecure states until a stop splitting criterion is met. There are two conditions which thus yield two types of terminal nodes : *leaves* and *deadends*. A leaf is a node which corresponds to a sufficiently pure subset (e.g. all states belong to the same class). A deadend is a node where there is not enough statistical support for choosing an appropriate test. Stop splitting at deadend nodes prevents the tree from overfitting the learning set and hence allows the method to reach a good compromise between accuracy and simplicity.

The end result of this procedure is the tree shown at Fig. 3.5 partitioning the learning set into subregions defined by line segments orthogonal to the Pu or Qu axes; this "orthogonal" tree is composed of 18 test nodes, 12 leaves and 7 deadends.

Validation

Since the tree is grown to reach a good compromise between simplicity and separation of secure and insecure *learning* states it provides a kind of summary of the relationship observed in the learning set between Pu and Qu attributes and security class. But, how well does it generalize to unseen states ? To answer this question, we use the test set of 2000 states different from the learning states and compare the security class predicted by the tree with the one derived from the CCT computed by numerical simulation.

Thus each test state is directed towards a terminal node on the basis of its input attribute values (Pu and Qu) and applying sequentially the tests encountered to select appropriate successors. When a terminal node is reached, the output majority class of the corresponding sub-learning-set stored there is retrieved to classify the test state. E.g. states reaching terminal nodes L1, L2, D1, L4, D3, D5, D6, D7, L7 and L10 are predicted to be insecure, while those reaching terminal nodes L3, D2, D4, L5, L6, L8, L9, L11 and L12 are predicted to be secure. Among the 2000 test states, this yields 1954 correct classifications, 15 insecure states declared erroneously secure, and 31 false alarms, i.e. an error rate Pe of 2.3%.

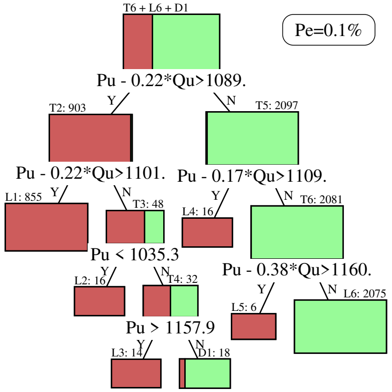


Figure 3.6 *"Oblique" decision tree*

Refinements

There are many refinements of the TDIDT method of interest in the context of security assessment. First of all, decision trees may exploit easily discrete attributes (e.g. to represent topology) together with numerical ones. They may also be generalized to an arbitrary number of (security) classes and to tests with more than two outcomes.

Another interesting extension consists of using linear combinations instead of single attribute (orthogonal) splits, yielding so-called "oblique" decision trees. They are useful when there are strong interactions among different candidate attributes. For example, in our illustrative problem we could use linear combinations among Pu and Qu, which should provide a more efficient separation between secure and insecure states.

Figure 3.6 shows a tree obtained in this fashion. During tree building, we search for splits in the form of "Pu + Weight*Qu<Threshold" instead of searching for single attribute splits (in the form of "Pu<Threshold" and "Qu<Threshold"). The optimal splitting procedure is modified in order to determine automatically both an appropriate weight and the optimal threshold at each test node. The fact that the resulting "oblique" tree is significantly simpler than the "orthogonal" one of Fig. 3.5 (only 6 test nodes, 6 leaves and 1 deadend) confirms our intuition. The tree is also much more reliable (no non-detections and only two false alarms among the 2000 test states, i.e. an error rate of 0.1%). Figure 3.7 further illustrates the difference between the two classification boundaries induced by the two trees : a rather rough staircase approximation for the orthogonal tree; a much smoother boundary for the "oblique" tree.

The only price to pay for this improvement is an increase in CPU time at the tree growing stage, since searching for linear combinations is more intricate than searching for optimal thresholds. E.g. in our example it took 120 seconds[2] to grow the "oblique" tree and only 13 seconds to grow the "orthogonal" one.

In addition to "oblique" trees, other interesting extensions are *regression* trees which infer information

[2] CPU times on a SUN Sparc10 workstation.

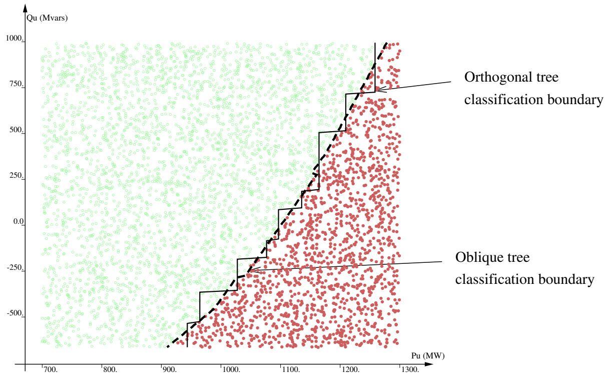


Figure 3.7 *Classification boundaries of the DTs of Figs. 3.5 and 3.6*

about a numerical output variable, and *fuzzy* trees which use fuzzy logic instead of standard logic to represent output information in a smooth fashion [BW95, BW96]. Both approaches allow us to infer information about security margins, similarly to the techniques discussed below in §§3.4 and 3.5.

Salient features of decision trees

The main strength of decision trees is their interpretability. By merely looking at the test nodes of a tree one can easily sort out the most salient attributes (i.e. those which most strongly influence the output) and find out how they influence the output. Furthermore, at the tree growing stage the method provides a great deal of additional information, e.g. about scores of different candidate attributes, their correlations, and the overall information they provide to the tree.

Another very important asset is the ability of the method to identify the most relevant attributes for each problem. Unfortunately our toy problem was too simple to illustrate this feature, but in large-scale applications typically less than twenty percent of the candidate attributes are actually selected while growing a tree.

The last characteristic of decision trees is their computational efficiency. Typically, tree growing computational complexity is linear in the number of candidate attributes and in the number of learning states, allowing one to tackle problems with a few hundred candidate attributes and a few thousand learning states, with response times of only some minutes. The use of a tree to classify an unseen situation is ultrafast since only a few logical tests need to be computed.

Thus computational efficiency together with interpretability enable the method to be used in an interactive trial and error fashion, so as to discover interesting information contained in a data base and gain physical insight into a problem.

In the next three sections we will describe methods which are complementary to decision trees and may be combined with them in various hybrid approaches.

3.4 Smooth non linear approximations via artificial neural networks

The field of artificial neural networks (ANNs) has grown to an important and productive research field. We restrict ourselves to multilayer perceptrons; for further information, a widely recommended theoretical introduction to neural networks is given in [Hay94]. We mention here that in addition to neural networks there exists a large diversity of modern statistical regression techniques which have similar, sometimes better abilities.

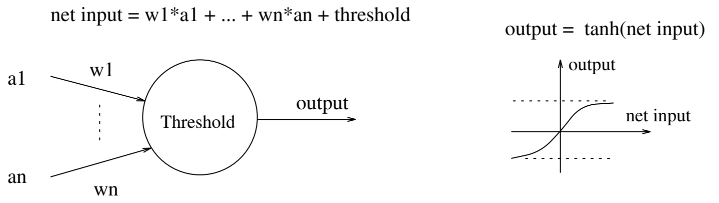


Figure 3.8 *Perceptron (neuron)*

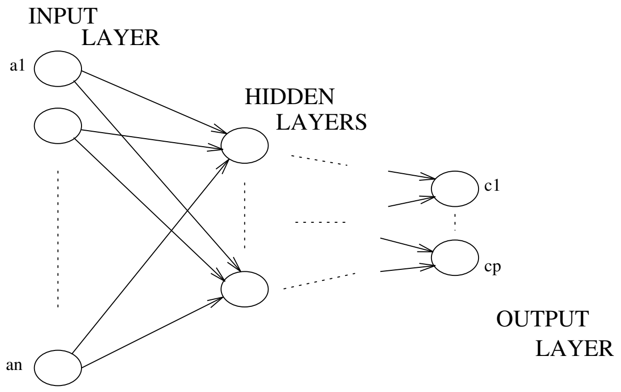


Figure 3.9 *Multilayer perceptron*

Multilayer perceptrons

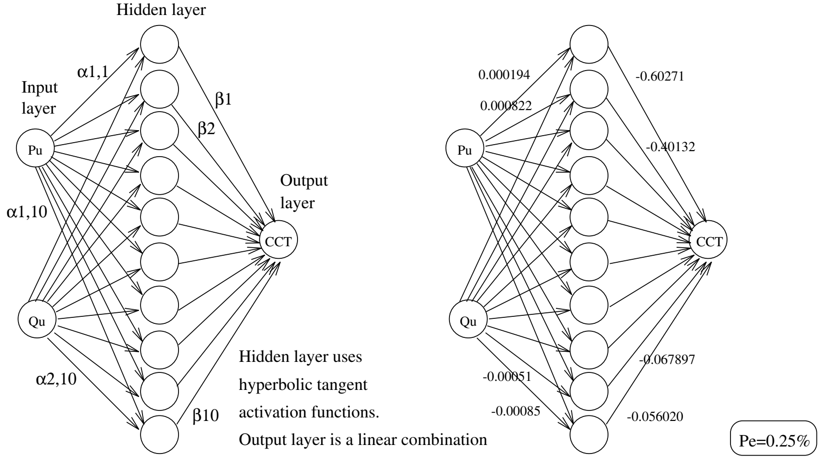
Work on artificial neural networks started several decades ago with the work on perceptrons. Figure 3.8 illustrates the perceptron, which is basically a simple linear threshold unit, thus able to represent only linear boundaries in the attribute space. Its limited representation capabilities have motivated the consideration of more complex ANNs, composed of multiple interconnected layers of perceptrons, multilayer perceptrons (MLPs) for short. These latter are able to represent non linear input/output functions in a very flexible way.

Figure 3.9 illustrates a typical multilayer perceptron. Each neuron is a perceptron : input layer neurons are fed with linear combinations of the input attributes; hidden and output layer neurons receive linear combinations of outputs from neurons in the preceding layers.

Learning

In the context of multilayer perceptrons, the learning stage consists of determining an appropriate structure of the MLP and of identifying appropriate values of the different parameters (weights and thresholds).

The structure is defined by the number of neurons, the topology of their interconnection, and the type of activation functions they use. Usually, it is determined by a trial and error procedure. However,



(a) Initial values of the weights are random

(b) After training the weights are tuned to fit the problem

Figure 3.10 *Single hidden layer perceptron*

nowadays there exist various algorithms to determine the structure automatically.

The parameter identification task amounts to a complex non linear numerical optimization problem, which may be solved by various techniques. Historically, the first method which was proposed was the so called "back-propagation" algorithm, which is equivalent to a fixed step gradient descent technique. It is interesting from a biological point of view, but rather inefficient from a computational point of view. Nowadays, one uses generally second order quasi-Newton methods.

Parameter identification

In our illustrative problem MLPs can be exploited interestingly to approximate the CCT as a closed form function of Pu and Qu. Generally, for the approximation of a continuous function MLPs with a single hidden layer may provide good approximators [Bar93]. Thus, let us try to approximate the CCT with such a structure. The learning set used will be the same 3000 input states that we used for building the decision trees, but we associate as output information to each state its CCT, rather than the security class.

Figure 3.10 graphically sketches the MLP that we have used, containing 10 hidden neurons. Each hidden neuron i has an input/output relationship of the form

$$\text{Output}_i(\text{state}) = \tanh(\alpha_{i1}Pu(\text{state}) + \alpha_{i2}Qu(\text{state}) + \theta_i), \tag{3.1}$$

where α_{i1} (resp. α_{i2}) is the connection weight between the neuron and the Pu (resp. Qu) input, and θ_i its threshold.

The output of the MLP is obtained as a linear combination of the preceding functions, i.e.

$$\text{CCT}_{\text{MLP}}(\text{state}) = \sum_{i=1\ldots10} \beta_i \tanh(\alpha_{i1}Pu(\text{state}) + \alpha_{i2}Qu(\text{state}) + \theta_i), \tag{3.2}$$

where β_i represents the contribution of neuron i in the overall output.

The parameter identification thus aims at choosing appropriate values of the 40 parameters ($\alpha_{ij}, \theta_i, \beta_i$), in order to fit for each learning state the MLP output to the CCT value determined by numerical simulation. The fitting criterion we use is the total sum of square errors (TSE)

$$\text{TSE} = \sum_{\text{state}\in LS} |\text{CCT}(\text{state}) - \text{CCT}_{\text{MLP}}(\text{state})|^2, \tag{3.3}$$

which is a smooth, complex non linear function of the parameter values, which needs to be minimized.

Before starting the learning procedure the parameters are all initialized at random, then they are progressively adapted in order to minimize the TSE. In our example we used a Broyden-Fletcher-Goldfarb-Shanno (BFGS) method, which is a second order method iteratively building up an approximation of the inverse Hessian matrix.

At initialization the value of TSE was equal to 206.870605, corresponding to the random initial parameter values. After 46 iterations the algorithm stops at a local minimum, having reduced the TSE to 0.000941. Given the very small value of the TSE we deem that we are close to the global minimum. All in all, the parameter adaptation process took 730 CPU seconds on a Sparc 10 SUN workstation.

The resulting closed form approximation of the MLP input/output function corresponding to the final parameter values is as follows

$$\begin{aligned}\text{CCT}_{\text{MLP}} \quad = \quad & -0.602710\tanh(0.000194Pu - 0.00034Qu - 0.93219)\\ & -0.401320\tanh(0.000822Pu - 0.00020Qu - 0.76681)\\ & +0.318249\tanh(0.000239Pu - 0.00050Qu - 0.29351)\\ & -0.287230\tanh(0.002004Pu - 0.00034Qu - 1.20080)\\ & +0.184522\tanh(0.000131Pu - 0.00057Qu - 0.03152)\\ & +0.177701\tanh(0.001799Pu - 0.00011Qu - 2.08190)\\ & -0.150720\tanh(0.001530Pu - 0.00056Qu - 1.68040)\\ & +0.142678\tanh(0.002152Pu - 0.00046Qu - 1.72280)\\ & -0.067897\tanh(0.001910Pu - 0.00051Qu - 1.71343)\\ & -0.056020\tanh(0.000202Pu - 0.00085Qu - 0.39876)\end{aligned} \tag{3.4}$$

Validation

In order to evaluate the reliability of this approximation, we have used the MLP to predict the CCTs of the 2000 test states. Figure 3.11 shows the distribution of errors; it is clear that in this simple example the MLP approximates the CCT with very high accuracy. Thus, the MLP can be used in order to classify states with respect to a threshold. For example, with respect to the threshold of 155ms used in the decision trees, it classifies 5 insecure states as secure (their CCT is however very close to the threshold of 155ms) and makes no false alarms, i.e. its error rate is of 0.25%.

Note that since the MLP provides a very accurate closed form approximation of the CCT its derivatives may be computed analytically, and used to determine sensitivities of the CCT with respect to Pu and Qu. These derivatives could in turn be used to find out preventive control actions to increase the value of the CCT whenever it is found to be too small.

Refinements

Although in many problems a single hidden layer is sufficient, it is straightforward to generalize the MLP by adding any number of further layers. It is also possible to use other activation functions than the hyperbolic tangent, e.g. Gaussian or trigonometric functions.

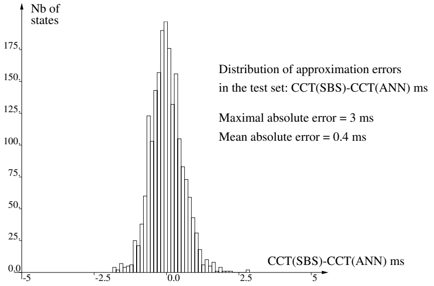


Figure 3.11 *Distribution of MLP approximation errors*

Another extension consists of growing the neural network by progressively adding neurons and/or layers. Reciprocally, pruning techniques were designed so as to remove the useless connections and hence reduce overfitting problems. There are even techniques which are able to adapt, automatically during the learning procedure, the shape of the activation functions to the problem features, like the projection pursuit regression method [FS81, FSS84, Fri87].

Finally, let us mention that the MLP learning algorithm may be used in an adaptive on-line scheme, so as to adapt parameters whenever new learning states become available.

Salient features of MLPs

The main characteristic of MLPs is flexibility in approximating non-linear functions in multidimensional spaces.

This flexibility is obtained at the expense of high computational burden. In real-life problems, when the number of inputs and hidden neurons is large, training times are typically of several hours to several days. At the same time, it becomes rather difficult to appraise and interpret the type of input/output relationship represented by such an MLP, which behaves like a black box.

3.5 Memory based reasoning via statistical pattern recognition

Decision trees and multilayer perceptrons essentially compress detailed information about individual simulation results contained in their learning set into general, more or less global security characterizations.

Additional information may however be provided in a case by case fashion, by matching an unseen situation with similar situations found in the data base. This may be achieved by defining generalized distances so as to evaluate similarities among power system situations, together with appropriate fast data base search algorithms.

We refer the reader interested by pattern recognition (PR) methods to [DH73]. Here we will only describe the so-called “K nearest neighbors” (KNN) method.

KNN consists of classifying a state into the majority class among its K nearest neighbors in the learning set. In its most simple version the learning stage of the KNN method thus merely consists of storing the learning states in a table; the actual work (computing the distances and sorting out the K nearest neighbors) is done when the method is used to predict output for an unseen state.

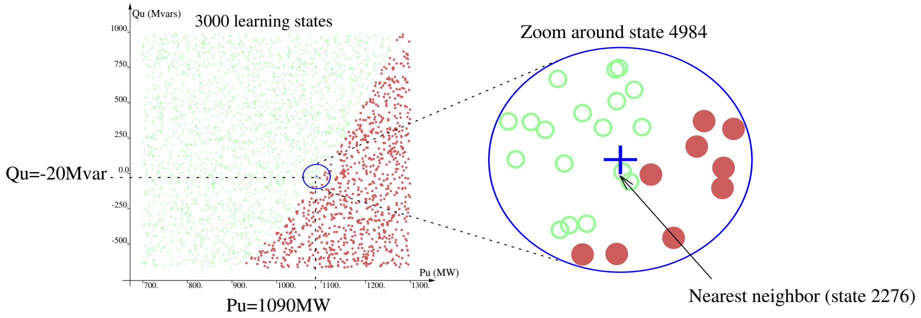


Figure 3.12 *Learning set of 3000 random states and nearest neighbors of state 4984*

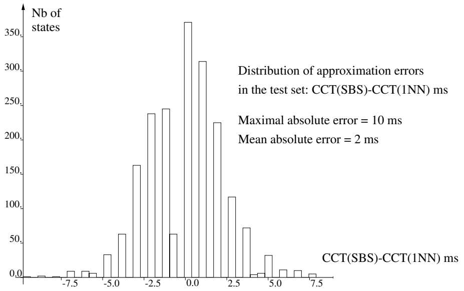


Figure 3.13 *Distribution of 1NN approximation errors*

For example, in our example let us consider the state no 4984 of our data base (a test state). Its values of Pu and Qu are respectively of 1090 MW and -20 Mvar. Figure 3.12 shows in its left hand part the location of this state in the attribute space together with the learning states. In the right hand part we have zoomed on the nearest neighbors of the state. Note that the points on the borderline of the zoom region are equidistant (Euclidean distance) to the test state.[3] One may identify on Fig. 3.12 the nearest neighbor, i.e. the learning state closest to the test state (state no 2276 : Pu=1090 MW, Qu=-31 Mvar, and CCT=0.157s). Thus, according to the 1 nearest neighbor (1NN) rule, the CCT of the test state will be approximated to 0.157s and it would be classified into the secure class. Note that its actual CCT is equal to 0.158s; hence the state is correctly classified, in spite of being very close to the security boundary.

Validation

Repeating this procedure for all 2000 test states yields an error rate of 0.9%. Figure 3.13 shows the distribution of CCT approximation errors. Comparing with Fig. 3.11, we notice that the 1NN approximation is slightly less accurate than the MLP approximation. On the other hand, the 1NN provides additional information to that of the MLP and the DT : the distance to the nearest neighbors, attribute values of the nearest neighbors, and more generally any type of information attached to the nearest neighbors, like, for example, optimal preventive or emergency control strategies.

[3] The equidistant region is slightly oval due to the fact that we have normalized Pu and Qu by their standard deviation before computing the distance.

Refinements

The basic refinement consists of using K neighbors instead of a single one, in order to extrapolate information in a more reliable fashion. Then, since the nearest neighbor rule is quite sensitive to the distance chosen, in many practical problems it is necessary to down weight less relevant attributes and enhance more relevant ones. Thus, distance learning algorithms have been devised so as to choose automatically the weights (and also the value of K) on the basis of a learning set. A further refinement consists of using different distance definitions in different regions of the attribute space.

In §5.6.1, we will illustrate some of these features on a real large-scale problem.

Salient features of KNN

The main characteristics of this method are high simplicity but also sensitivity to the type of distances used. In particular, to be practical, ad hoc algorithms must be used to choose the distances on the basis of the learning set [HWP95, WHP95a, HWP97].

The fact that the KNN approach is quite similar to human reasoning (recalling similar situations seen in the past) makes it also interpretable by human operators.

3.6 Hybrid automatic learning methods

The preceding presentation has shown the complementary nature of the different automatic learning methods. Of course, in the simplistic illustrative problem used above all three types of methods appear to work very effectively. In real life DSA problems the number of input parameters is generally two to three orders of magnitudes larger. Thus, dimensionality problems appear which have to be tackled.

As we have seen, among the automatic learning methods discussed, only decision tree induction is presently able to handle large-scale problems efficiently. Multilayer perceptrons become excessively slow while nearest neighbor lacks of accuracy. Nevertheless, it is possible to use these methods, provided they are appropriately modified. In particular, they can be coupled with decision trees in hybrid approaches in order to reduce their weaknesses to some extent.

In these hybrid approaches decision trees are first used in order to have a first look at the data and in particular identify the most important parameters for a given problem, generally less than 20% of the initial candidate attributes used to build the tree. Then these parameters are used as input variables for the other two types of techniques, thus improving their performances in terms of computing times and accuracy.

Table 3.1 taken from [WP96b] summarizes the main characteristics of various pure and hybrid techniques, in particular in terms of the type of information they can exploit/provide, their expected level of accuracy, and their flexibility.

In terms of accuracy, there exists no universal panacea. However, while in general each method has its own field of competence, in security assessment problems those methods which exploit margins rather than classes (especially with smooth models) generally provide increased accuracy, and also more refined security assessment.

In particular, the proper way to exploit margins consists of saturating them outside a small window around the relevant classification threshold and building an approximate regression model, using only those attributes which influence the margin value within this window. If the end result searched is a discrete classification it can be derived straightforwardly by discretizing the output of this model. By doing so one succeeds in taking advantage simultaneously of continuous margins and problem simplicity.

Table 3.1 *Salient features of AL methods applied to security assessment*

Method		Functionalities	Computational	
			Off-line	On-line
Pure	Crisp DTs	**Good interpretability (global)**. Discrete. Good accuracy for simple "localized" problems. Low accuracy for complex, diffuse problems.	Very fast	Very fast
	MLPs	**Good accuracy**. Low interpretability. Possibility for margins and sensitivities.	Very slow	Fast
	kNN	**Good interpretability (local)**. Conceptual simplicity.	Very slow	Very slow
Hybrid	Fuzzy DTs	**Good interpretability (global)**. Symbolic and continuous. More accurate than crisp trees. Possibility for margins and sensitivities.	Slow	Fast
	DT-ANN	Combine features of DTs and MLPs	Slow	Fast
	DT-kNN	Combine features of DTs and kNNs	Slow	Slow

In terms of CPU time, the variations are much larger. For example, growing a decision tree can be up to 1000 times faster than optimizing the weights of multilayer perceptron for the same problem. Thus, while the former method may be used in an interactive trial and error fashion, the latter is hardly practical for large data sets, typically encountered in power system security problems.

3.7 Unsupervised learning

In contrast to supervised learning, where the objective is clearly defined in terms of modeling the underlying correlations between some input variables and some particular output variables, unsupervised learning methods are not oriented towards a particular prediction task. Rather, they try to find out by themselves the existing relationships among states characterized by a set of attributes.

Thus, one of the purposes of clustering is to identify homogeneous groups of similar states, in order to represent a large data base by a small number of representative *prototypes*. Graphically, two-dimensional scatter plots may be used as a tool in order to analyze the data and identify clusters.

Another application of the same techniques is to identify correlations (and redundancies) among the different attributes used to characterize states. In the context of power system security both applications may be useful as complementary data analysis and preprocessing tools.

Unsupervised learning algorithms have been proposed under the three umbrellas given above to classify classification methods, termed *cluster analysis* in the statistics literature, *conceptual clustering* in the machine learning community, and *self-organizing maps or vector quantization* in the neural net community [Koh90].

Unsupervised learning methods become really useful only in the context of large scale data bases, containing several thousand states described by many attributes. A more detailed discussion of these methods falls out of the sciope of this tutorial.

4

Overview of security problems

In this section we provide a brief overview of power system security and possible applications of automatic learning. We start by reviewing the different types of physical problems, restricting our focus on DSA. Then we will consider the different working environments where security assessment tools are needed, and comment on the applicability of automatic learning.

4.1 Operating modes

Security assessment consists of evaluating the ability of the power system to face various disturbances and of proposing appropriate remedial actions able to counter its main weaknesses, whenever deemed necessary. Disturbances may be due to external or internal events (e.g. faults subsequent to lightning vs operator initiated switching sequences) and may be small (slow) or large (fast) (e.g. random behavior of the demand pattern vs generator or line tripping).

The different operating modes of a power system were defined by Dy Liacco [Dy 68]. Figure 4.1 shows a more detailed description of the “Dy Liacco state diagram”.

Preventive security assessment is concerned with the question whether a system in its normal state is able to withstand every plausible disturbance, and if not, preventive control would consist of moving this system state into a secure operating region. Since predicting future disturbances is difficult, preventive security assessment will essentially aim at balancing the reduction of the *probability* of losing integrity with the economic cost of operation.

Emergency state detection aims at assessing whether the system is in the process of losing integrity, following an actual disturbance inception. This is a more deterministic evolution, where response time is critical while economic considerations become temporarily secondary. Emergency control aims at taking fast last resort actions, to avoid partial or complete service interruption.

When both preventive and emergency controls have failed to bring system parameters back within their inequality constraints, automatic local protective devices will act so as to preserve power system components operating under unacceptable conditions from undergoing irrevocable damages. This leads to further disturbances, which may result in system splitting and partial or complete blackouts.

Consequently, the system enters the *restorative* mode, where the task of the operator is to minimize the amount of un-delivered energy by re-synchronizing lost generation as soon as possible and picking up the disconnected load, in order of priority. While automatic learning may be useful in the context of restoration [KAG96], we restrict our discussion to preventive and emergency modes.

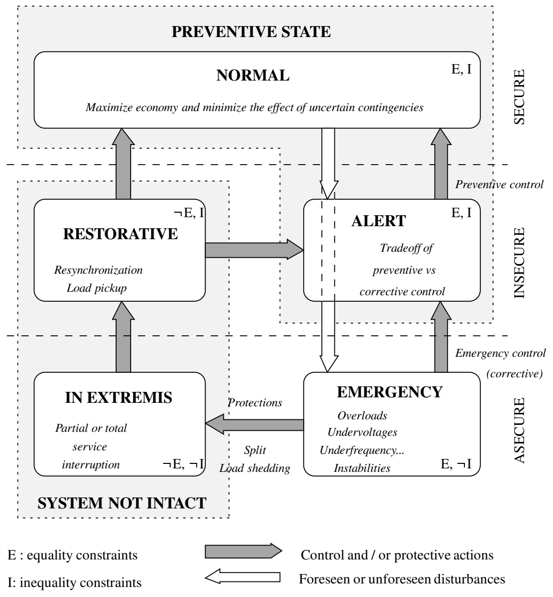


Figure 4.1 *Operating states and transitions. Adapted from [FC78]*

4.2 Physical classification of DSA problems

Figure 4.2 taken from [KM97] provides an overview of various types of stability problems which have to be tackled in DSA. Beneath each type of stability problem is indicated the type of phenomena which are characteristic of this type of instability and the type of physical causes which drive the problem.

Various security problems are distinguished according to the characteristic symptoms (low voltage, large angular deviations. . .), and the control means (reactive power, switching. . .) to alleviate problems, the time scales of the dynamics, and further the amplitude of disturbances.

For example, in transient stability, the dynamic performance is a matter of seconds and is mainly affected by switching operations and fast power controls (e.g. fast valving, high voltage direct current converters, FACTS) and voltage support by the automatic voltage regulators of synchronous generators and static var compensators (SVCs).

In the context of voltage stability, the fastest phenomena are characterized by sudden voltage collapses developing at even higher speeds than loss of synchronism. More classical is the *mid-term* voltage instability, which corresponds to a typical time frame of one to five minutes. In this case voltage collapse is mainly driven by automatic transformer on-load tap changers trying to restore voltage nearby the loads. There is a third, even slower time frame, corresponding to the so-called *long-term* voltage instability, which involves the gradual buildup in load demand. This interacts with classical static security and is well within the scope of operator intervention.

Note that in some particular contexts these phenomena may interact strongly and their distinction becomes useless.

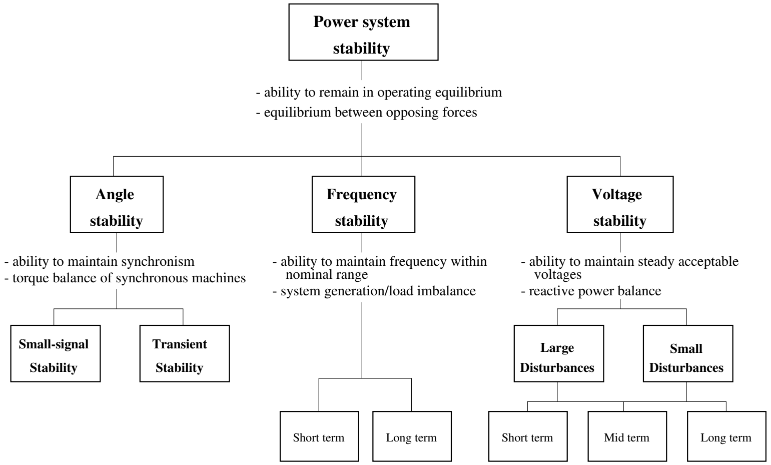

Power system stability
- ability to remain in operating equilibrium
- equilibrium between opposing forces
Angle stability
- ability to maintain synchronism
- torque balance of synchronous machines
Frequency stability
- ability to maintain frequency within nominal range
- system generation/load imbalance
Voltage stability
- ability to maintain steady acceptable voltages
- reactive power balance
Small-signal Stability
Transient Stability
Large Disturbances
Small Disturbances
Short term
Long term
Short term
Mid term
Long term

Figure 4.2 *Types of power system stability phenomena. Taken from [KM97]*

Static security

Static security, which concerns essentially thermal overload problems of generation transmission system components, is by definition out of the scope of DSA. Nevertheless, it is very often the initiating factor leading subsequently to loss of synchronism or voltage collapse phenomena.

The so-called static phenomena span over significantly longer periods of time. For example, line overloads may be tolerated during 30 to 60 minutes under favorable weather conditions. Due to the longer time frames, operator based emergency control may be possible within the context of static security, provided that corrective actions have been prepared in the preventive mode. In all other cases, emergency control must be carried out automatically.

4.3 Practical application environments and possible uses of automatic learning

Table 4.1 shows the practical study contexts or environments which may be distinguished in security assessment applications. The second column specifies how long in advance (with respect to real-time) studies may be carried out; the last two columns indicate respectively if an operator is involved in the decision making procedure and if an expert in the field of power system security is available.

Generation-transmission system planning

In system planning, multitudinous configurations must be screened for several load patterns, and for each one a large number of contingencies. An order of magnitude of 100,000 different scenarios per study would be realistic for a medium sized system. While enough time may be available to carry out so many security simulations, there is still room for improved data analysis methods to exploit their results more effectively for the identification of structural system weaknesses and to provide guidelines to improve reliability.

Note that the probabilistic Monte-Carlo simulation based planning tools could also be adapted so as to take advantage of automatic learning methods. For example the scenarios generated by random sampling could be stored in a security information data base and further analyzed by automatic learning. The results of automatic learning could then be reused so as to define better sampling schemes in order to reduce variance in subsequent studies [Weh95].

Table 4.1 *Security assessment environments. Adapted from [WP93a]*

Environm.	Time scales	Problems	Operator	Expert
System planning and design	1 - 10 years	Generation-Transmission system Protection systems Control systems	No	Yes
Operation planning	1 week - 1 year	Maintenance Unit commitment Protection settings	No	Yes
On-line operation	1 hour - 1 day	Preventive mode Security assessment	Yes	Partly
Real-time*	sec. - min. - hour	Emergency control Protective actions	No**	No
Training	months - days	Improve operator skill	Yes	No

* Here we distinguish between *real-time*, which considers dynamic situations following a disturbance inception, from merely *on-line* which considers static pre-disturbance situations.

** except for static security corrective control

Design of protection and control systems

The other type of task which is carried out in the off-line study environments concerns the design and validation of all kinds of protection and control systems. We will elaborate a little more on this topic, since automatic learning could be particularly useful in this context. For the sake of clarity we will not distinguish between the types of protection and control systems considered (local, centralized, special stability controls, . . .).

As concerns existing protection and control systems, the automatic learning framework would be useful in terms of analysis, in order to evaluate performances in the context of diversified simulation scenarios. Furthermore, the automatic learning methods could be used so as to tune various parameters (gains, temporizations, thresholds. . .) in the most effective way and find out strategies to decide how to adapt these latter to changing operating conditions (e.g. winter and summer settings of low voltage thresholds for automatic tap changer blocking schemes. . .).

As concerns the design of new systems, the automatic learning methods may be used in order to identify the most appropriate real-time measurements and/or signals and to determine appropriate control laws or protection logics. The resulting systems may then be validated against a diversified test set of simulation scenarios before going towards field tests.

Note that, while the distinction of different types of phenomena provided in Fig. 4.2 is interesting from a conceptual point of view, it may become irrelevant in the context of the most extreme operating modes when the power system is undergoing a breakdown scenario.

Thus, in the context of special stability control systems' design it may be wiser to look at the power system dynamics as a single global phenomenon, in order be able to study interactions of phenomena and related protection systems. In Chapter 5, we consider an example data base generation within this context, drawn from a research collaboration between Electricité de France and the University of Liège [WLTB97a].

Operation planning

Operation planning, as suggested in Table 4.1, concerns a broad range of problems, including maintenance scheduling (one year to one month ahead), design of operating strategies for usual and abnormal situations, and setting of protection delays and thresholds. The number of combinations of situations which must be considered for maintenance scheduling is also generally very large, and automatic learning approaches will be useful to make better use of the available information and to exploit the system more economically.

In the context of operation planning, it may be possible to re-tune protection and control law parameters in order to adapt them to unforeseen conditions, yielding similar applications of the automatic learning framework than those discussed above in the context of the design environment.

Similarly, for the closer to real-time determination of operating security criteria it would allow engineers to screen more systematically representative samples of situations, in order to identify critical operating parameters and determine their security limit tables needed for on-line operation.

These types of applications will be further illustrated later on.

On-line operation

In on-line operation, it is presently not feasible to generate data bases automatically nor to extract information from them by applying automatic learning.

However, the data bases generated off-line and decision rules extracted from them may be exploited for on-line decision making. On-line operation will also provide the required feedback to the engineers in charge of defining strategies, when major changes happen in the system.

In the future, with faster computers and more efficient security assessment tools it is also conceivable that data bases and security criteria might be refreshed automatically on-line [DL96]. We mention also the very ambitious proposal made by Dr. Rovnyak and Prof. Thorp from Cornell University which aims at building on-line decision trees for real-time stability control [RKTB94].

Real-time monitoring and control

A fortiori, in the context of real-time monitoring and control it not feasible today to build data bases and apply automatic learning.

On the other hand, as we mentioned above, automatic learning may be used off-line to design criteria to trigger automatically emergency control actions, so as to prevent a disturbed system state to evolve towards blackout.

Even more than in the preventive mode studies, it is important to use appropriate models to reflect the *disturbed* power system behavior, when designing these security criteria, and in particular, to take into account modeling uncertainties and measurement errors while generating the data bases.

Furthermore, the use of readily available system *measurements* as inputs to the derived emergency control rules is often an operational constraint in addition to minimal data requirements and ultra high speed.

From the automatic learning point of view, one of the main difficulties is to handle the dynamic time varying nature of attributes.

Operator training

During operator training, the security criteria derived in either of the preceding contexts might be usefully exploited as guidelines, provided that they are presented in an intelligible way. In addition, these models might be used internally in a training simulator software, in order to set up particular scenarios presenting particular insecurity modes.

4.4 Analytical tools

In addition to standard time domain numerical simulation, a rather large set of numerical methods are available for security assessment in the different time frames mentioned [Cigré97a]. We call them *analytical* tools since they exploit analytical power system models in contrast to the *synthetic* ones extracted by automatic learning techniques.

All these tools, provided that they are accepted by the concerned utility, may be exploited during the data base generation.

Furthermore, the automatic learning framework may be used in order to assess simplified tools and/or simplified dynamic models, by comparing systematically and on a large number of simulation scenarios their security assessment with the one provided by reference methods and/or reference dynamic models. This would allow to calibrate simplified models and security indicators provided by fast screening tools, and determine their validity and error bounds with respect to a representative set of security scenarios.

4.5 Summary

The effect of a contingency on a power system in a given state can be assessed by numerical (e.g. time-domain) simulation of the corresponding scenario or by other analytical tools. However, the nonlinear nature of the physical phenomena and the growing complexity of real-life power systems make security assessment a difficult task. For example, the everyday monitoring of a power system calls for fast analysis, sensitivity analysis (which are the salient parameters driving the phenomena, and to which extent?), suggestions to control.

Thus, there is a very large diversity of security problems and the way they are tackled in practice is generally power system specific. Very often also, the methodologies, models and criteria used in planning environments are different from those used for operation, which leads to further difficulties.

The automatic learning framework, due to its flexibility, offers promising capabilities in all these problems, and proposes a unified methodology which can be used in all environments. Thereby, information could be shared more easily and more systematically between planners, operation planners and operators. As we pointed out, it enables one also to take into account modeling uncertainties and measurement errors in the security assessment task.

The sceptic reader might wonder whether all this is really feasible, or how well it could work in practice for complex large scale power systems. The last chapters of this course will provide some answers.

In particular, in the next chapter we will describe a sound methodology and technical means to generate high quality security information data bases, in order to make these capabilities become reality. We will also illustrate it on some real-life case studies. However, in order to appraise actual interest in practice we will have to wait until we come to the third part of the course, where a great deal of the theory will be illustrated on a real case study.

5

The data base generation problem

In this chapter we describe in detail the methodology and tools needed to generate sound data bases. The quality of the security information data base is paramount. If the data base is biased, unrepresentative, or too small, then the information extracted by automatic learning will probably be useless.

In the literature, some researchers have proposed clever techniques to a priori reduce the size of data bases, in order to reduce the computational burden. Indeed, by choosing the simulation scenarios in a sequential fashion, similar to the trial and error procedures used by human experts one can try to localize the scenarios at the vicinity of security boundaries. Unfortunately, the scenarios contained in such data bases are correlated, and strongly biased by prior information. Our experience has shown that this may dangerously affect the quality of the information extracted by automatic learning.

Our point of view is that today computing power and storage are not a bottleneck anymore, and they become cheaper and cheaper everyday. What is really needed, is a sound methodology and appropriate software to take advantage of it. Thus, in order to avoid bias, in our methodology a data base is first specified (study scope, random sampling, extracted information), then the scenarios are generated automatically, and finally they are simulated (if necessary, by exploiting parallel computations). In particular, during random sampling, simulation scenarios are chosen independently from each other, and before automatic learning is applied, the data base goes through a validation stage.

Below, we first start by describing what we mean exactly by a security scenario. Then, we go through the overall process of data base generation, and discuss in detail the different steps. The methodology has crystallized during research collaborations with industry (Electricité de France and Hydro-Québec) in the context of large scale DSA problems (transient stability, voltage stability, preventive and emergency modes). At the end of this section we will briefly describe two examples of these.

5.1 Security scenarios

In the context of a particular power system and DSA problem, a security scenario is defined by the three following components : initial operating point, external disturbances, dynamic modeling hypothesis. In some security studies all three components may vary randomly from one scenario to another. In other cases some are kept constant. For example, in the simple illustrative example of Chapter 3, only the operating point was variable.

Below we will further comment on each scenario component, in order to highlight different sampling strategies corresponding to different types of studies.

Initial operating point (OP)

The initial operating point is a static equilibrium at which the system is supposed to sit, before any external events start initiating dynamics. It is defined by available equipments in operation such as generators, lines, transformers, SVCs, capacitors, reactors, . . . , substation topologies, load level and pattern, generation schedules, interconnection tie line flows, and voltage/var dispatch.

Depending on the kind of study, it may be a normal secure or insecure state, optimally dispatched or not, viable or not. For example, in preventive security assessment studies we can consider all kinds of random viable states, in order to find out differences between secure and insecure ones, independently of any a priori operating philosophy, since the purpose is precisely to find out such philosophies. In other studies, for example considering the design of emergency controls, it may be interesting to consider only a small number of normal, N-1 secure operating states (see below).

External disturbances (ED)

The external disturbances are the events which will initiate the dynamics and drive the system away from its equilibrium. Depending on the type of study, they may be simple outages, load disturbances, faults, or any kind of combination of these.

For example, presently in most utilities the policy for preventive security assessment is deterministic. It consists in assuming a list of contingencies which the system must survive. Thus, in the data base these latter should be simulated for each operating point.

In other studies, for example for the design of special stability emergency control systems it would be wiser to consider a much larger diversity of randomized disturbances (e.g. fault duration and location, multiple faults, . . .). Similarly, future operating strategies might switch to probabilistic preventive security assessment, leading to the consideration of multiple disturbances, with different probabilities.

Dynamic modeling hypothesis (MH)

The dynamic modeling hypothesis concerns the parameters of the system (generators, lines, loads. . .) and the assumptions made concerning the behavior of various automatic actions which will take place in the system in order to respond to its dynamics (control loops, protections, special stability controls, reaction of external systems. . .), as well as manual actions (dispatchers, plant operators) which may interfere in the case of slow dynamics.

Note that, in most classical security assessment studies, the MH is considered to be fixed, just as if it were perfectly known. In the automatic learning approach, it is possible to randomize those aspects which are uncertain, according to the information at hand in the particular study context. For example, in planning studies it would be wise to randomize the characteristics of the not yet installed equipments. In operation planning studies, it might be wise to randomize load models and external system models. In emergency control studies, it might be wise to randomize also relay settings, fault impedances. . . and take into account possible malfunctionings.

5.2 Overall process of data base generation

Figure 5.1 summarizes the three successive steps of the data base generation process.

In practice, the specification is the most important and time consuming stage. As we will see below, it is at this stage that the existing expertise is injected in the automatic learning framework.

The second step is automatic, carried out by appropriate software tools. However, presently there exists

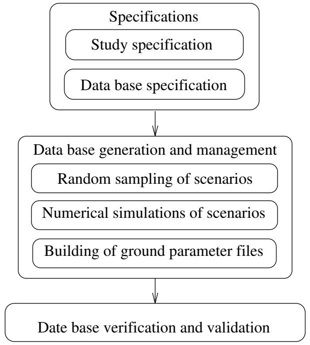

Specifications
Study specification
Data base specification
Data base generation and management
Random sampling of scenarios
Numerical simulations of scenarios
Building of ground parameter files
Date base verification and validation

Figure 5.1 *Overall data base generation process*

no single software package able to encounter the various needs of all types of security studies. Thus, in our research we have developed a series of specific tools to encounter the needs of specific types of applications. Below we will describe the outlook of a general tool which hase been developed recently.

The third and last step calls for engineering judgment together with data mining tools. Before extracting any security criteria from a data base, it is indeed necessary to verify its consistency with the initial specifications. Below we will provide some indications on the types of problems which might indeed lead to unrepresentative data bases.

5.3 Specifications

5.3.1 Study scope specification

The first step of the data base generation consists in specifying the range of scenarios that the particular study will address. While the approach aims at enabling the engineers to carry out more systematic and more global studies, it is generally necessary and even desirable to restrict the focus of a study to an a priori well defined scope (see [JWVP95], for a detailed discussion).

Starting with the existing expertise and problem statement, and depending on the kind of information expected from the study (e.g. preventive vs emergency), it is decided which parts of the security scenarios will be variable, which parameters to change for each component, what kind of simulations will be carried out (time scales, analytical security assessment tool, level of modeling), what information should be extracted from them.

Some base cases are selected, and a catalog of variable parameters which are important for the study under consideration is set up, as well as contingency lists and uncertain modeling components.

Constraints among the various parameters may also be defined in order to filter out unrealistic scenarios.

5.3.2 Data base specification

Random sampling specifications

In order to finalize specifications, it is first necessary to choose probability distributions for the random sampling procedure. In practice, one starts with existing statistical information about the variability in real life of the considered parameters. However, using this information directly is not possible in general, because it would not lead to rich enough data bases. In particular, in most cases it would lead to a very small number - if any - of interesting scenarios, among the few thousand which can typically be simulated. Thus, in practice it is necessary to bias probability distributions, for example in order to increase the proportion of stressed operating points, or dangerous faults. This is where the engineering judgment comes into play.

Thus the random sampling specifications are set up, generally through a sequence of discussions among experts in different fields, such as power system dynamics, protections and economic questions.

In the random sampling specification, it is useful to distinguish among *primary* and *secondary* parameters. The former are those upon which the security study is focusing, and in terms of which it is desired to characterize security. The latter are those parameters which are either uncontrollable, or unobservable or uncertain : they are made variable in order to yield robust security information.

Concerning operating point parameters, flexibility will depend on the type of tool used to build consistent operating points. For example, if a simple power flow calculation is used to build operating points, then the parameters must be consistent with the load flow equations : thus the random sampling specifications are formulated in terms of independent power flow input variables.

Extracted ground parameters

The other part of the data base specification concerns the choice of the *ground parameters* which will be extracted from the simulations and stored in the data base. These, and combinations of these will be used later as input and output variables for automatic learning.

Again, the type of parameters and how they are extracted from the simulation results will strongly depend on the type of study. For example, in preventive security assessment the input variables will typically be static operating point parameters (power flows, active and reactive generations, topology information) and the output information will be security margins or classes, defined with respect to a contingency list. In emergency control, the input parameters would be dynamic system measurements available in real time (voltages, rotor velocities . . .) and outputs would measure severity, e.g. load and generation loss.

Notice that in large scale DSA problems, there may be thousands of state variables and it is generally not necessary to extract them all, but it is important not to miss interesting ones. Our experience shows that in general a few hundred ground parameters are selected, at the data base specification time. Later on, some of them are found to be useless; others, which may be computed as functions of the ground parameters, can be easily added if required.

Acceptability criteria and filtering

During the specification of the data base it is generally not possible to guarantee that all scenarios will be realistic, reasonable, acceptable or even simulatable.

Thus, one should expect that some of the operating point specifications will lead to non converging power flows, or yield an unrealistic state. Similarly, the dynamic simulation tool may fail to simulate some of the very severe scenarios. Therefore, the last step of the data base specification consists of defining acceptability criteria which will be checked during the data base generation in order to filter only those scenarios which are deemed acceptable.

Number of simulation scenarios

The number of scenarios is a compromise between two contradicting requirements : the larger the data

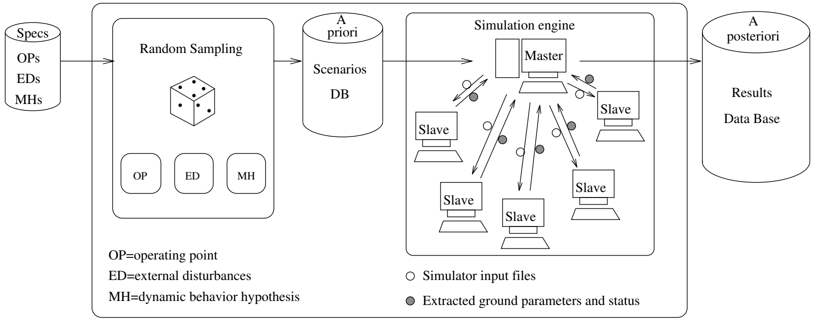


Figure 5.2 *Data base generation tool. Adapted from [WLTB97a].*

base, the better for automatic learning, but the larger also the required computational resources.

The minimal number of simulation scenarios required to obtain useful automatic learning results mainly depends on the problem complexity, which is generally not known in advance. Thus, a rule of thumb is a few thousand accepted scenarios after filtering. In order to reach this number it may be necessary to generate many more, depending on random sampling specifications, power system specifics and, of course, acceptability criteria.

How many scenarios can be simulated with acceptable response times will depend on the type of dynamic phenomena that are simulated (e.g. mid-term voltage security scenarios can be simulated more efficiently than transient stability ones), and on the type of information that is computed (e.g. margin calculations will take longer than mere security classifications), and of course on the computing power made available.

With present day workstations, the data base generation typically will take between a few hours and some weeks of CPU time. Data base sizes can range from a few MBytes (typical) to some GBytes (exceptional).

Summary

The quality of the information extracted by automatic learning is conditioned by the quality of the security information data base. Thus, the first time a new problem is considered, data base specification needs to be done very carefully, with as much as possible input from utility experts. The time required to finalize them may take a few weeks. Sometimes, a couple of iterations are necessary, involving the generation of small pilot data bases in order to tune various parameters. However, once the required information has been formalized and validated, when subsequently similar problems are considered the adaptation of the random sampling specifications is much more straightforward, and the software developed for the data base generation may be reused easily.

5.4 Data base generation and management

We now turn to the computational problems. How to carry out automatically the data base generation, and how to manage the resulting data base ?

Data base generation

Let us consider the advanced parallel scheme depicted on Fig. 5.2, very close to the tool used in the study reported in §5.6.2. It is composed of the following modules

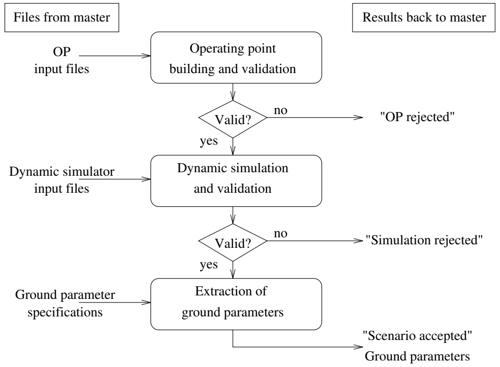


Figure 5.3 *Scenario simulation.*

1. Random sampling.
   Input : specification of the study scope, in terms of probability distributions, base case data files, dynamic modeling data, number of scenarios to generate, random number seeds.
   Output : a priori data base describing the sampled scenarios.
2. Simulation engine (master/slave organization).
   (a) Simulation input file builder (master).
   Input : a description of a scenario, reference input data files.
   Output : a set of modified input data files.

   (b) Task dispatch (master). (Files distributed via NFS)

   (c) Task simulation and extraction of relevant information (slave, see Fig. 5.3).
   Input : Input data files, specification of ground parameters to extract.
   Output: Extracted attribute files, simulation diagnostics.

   (d) Data base builder (master).
   Input : simulation diagnostics, results files for each accepted scenario.
   Output : results data base.

Master and slaves are standard Unix workstations, exchanging data through files. As soon as a slave becomes idle, it receives from the master a scenario to simulate. Simulation involves three successive steps (see Fig. 5.3) : (i) OP building; (ii) dynamic simulation; (iii) extraction of ground parameters.

A scenario may be rejected at the first stage if the operating point specification is not realistic, or at the second stage if the dynamic simulation fails. Thus each of the dispatched scenarios receives a label : accepted, rejected operating point, rejected simulation. These labels are collected by the master and included in the scenario "a priori" data base, for later validation. For the accepted scenarios, the slave sends also the extracted ground parameters back to the master, who collects this information and puts it into the "a posteriori" security information data base.

Data base management

At the present stage, we found that there is no need to use a sophisticated data base management system. The security studies are merely organized into directories, and the a priori and a posteriori data bases are organized into a series of files. The a priori data base collects random sampling specifications and statuses (accept or reject and reason) of the generated scenarios. The a posteriori data base collects the values taken for the ground parameters of the accepted scenarios. To ease access with standard tools, parameters of the same type are grouped together in flat ASCII files, which are compressed with a standard UNIX compression facility to save space. They may be easily exploited by various automatic learning algorithms, possibly after converting them to the appropriate format.

As we will see below, it is important to keep trace of the scenarios which were rejected so as to be able to analyze the validity of the data base. It is also useful to be able to pick a scenario from the a priori data base and re-simulate it, if required, e.g. for detailed analysis of some particular ones.

5.5 Data base validation

The validation of the data base consists of two steps.

The first step consists of analyzing the scenarios which have been rejected, in particular in order to find out whether the filtering doesn't alter too strongly the probability distribution of the independent parameters used in the random sampling. This analysis may be carried out by applying the data mining tools, mainly low level statistical visualizations, on the a priori scenario data base.

The next step of the data base validation concentrates on the analysis of the a posteriori security information data base. Again, the same types of low level data mining tools are applied in order to check that the information is sufficiently rich, i.e. how the ground parameters are distributed and correlated.

As mentioned above, during validation it is the responsibility of the engineer to decide to accept the data base, and proceed to the next step, or to reject it and suggest modifications to the random sampling specifications in order to generate a new data base.

5.6 Examples

5.6.1 Hydro-Québec (transient stability)

This system is characterized by very long UHV transmission lines carrying large amounts of power (735 kV lines carrying over 1500 MW, on distances over 1000 km); its transmission capacity is strongly related to transient stability limits. The objective of the research project (1992-93) was to assess whether the automatic learning framework could outperform the present manual approach used to build up stability limit tables used by the operators.

Study system and data base specification

Within this research, a data base was generated for the Hydro-Québec system corresponding to the situation of summer 1992. The first goal was to screen systematically all relevant "four-link" configurations of the James' Bay corridor, yielding a highly complex set of topologies. The reasons for choosing this situation were the high level of complexity, and the availability of optimized stability limits in LIMSEL (Hydro-Québec's on-line stability limit tables).

In order to generate the data base, the following variables were chosen as parameters of the random sampling procedure.

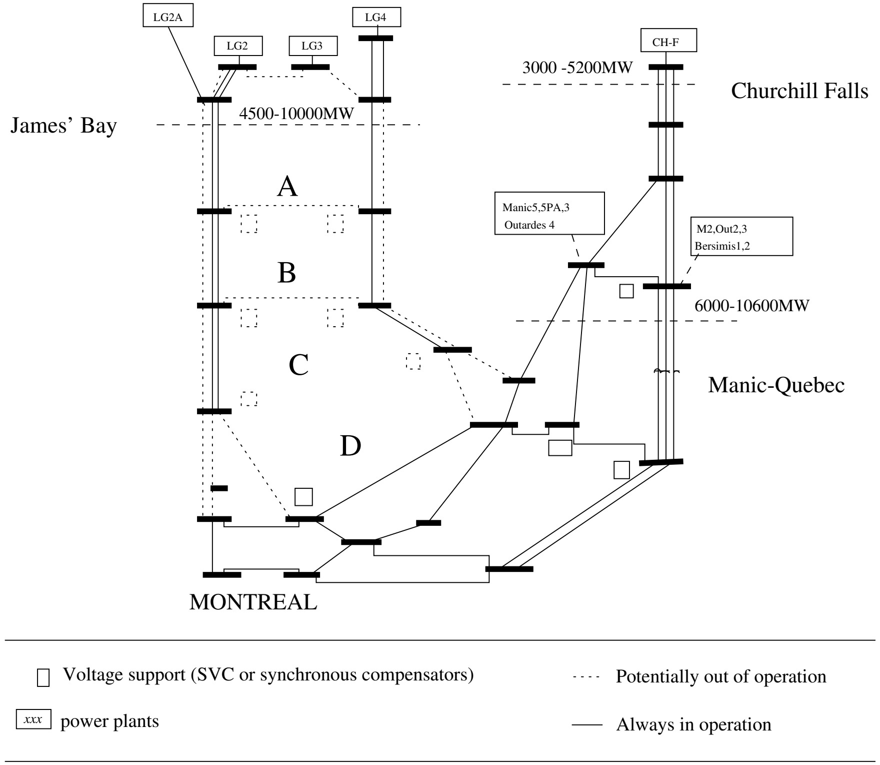


Figure 5.4 *Main transmission corridors of the Hydro-Québec system*

The power flows in the three important corridors of the Hydro-Québec system are drawn independently in the intervals indicated in Fig. 5.4. The James' Bay corridor corresponds to the study region whereas the Manic-Québec and Churchill Falls corridors are outside the study region but may influence the value of its stability limits.

The generation of the main complexes of hydro-electric power plants are adjusted so as to obtain the chosen power flows, while the distribution among the individual Lagrande and Manic/Outardes plants are randomized to yield a wide diversity among the power flows of the individual lines.

The topology is chosen independently according to a pre-defined list of possible combinations of line outages with respect to the complete five-link topology. Only the James' Bay corridor is modified and only so-called four link topologies are generated. This yields a total of more than 300 possible topologies.

The voltage support devices (SVCs and synchronous condensers) available in the six substations of the James' Bay corridor, indicated in Fig. 5.4, are widely variable during the random sampling since their influence on the stability limits is very strong. Their total number is drawn between 0 and 12 according to predefined probabilities, and their distribution in the substations is randomized.

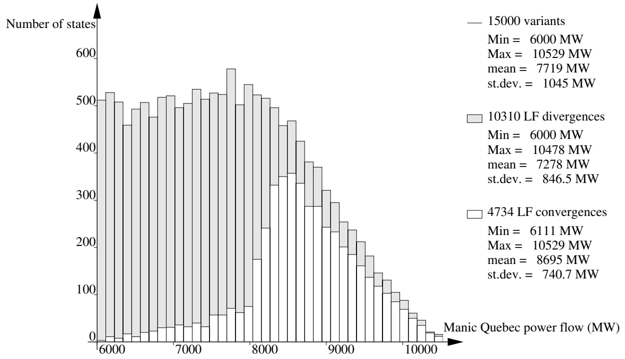


Figure 5.5 *Convergence diagram of Manic-Québec power flow (6 base case files)*

Data base generation

The above specifications led to the development of a specific software to generate a data base of random operating points. We expected difficulties with load flow convergence. Indeed, the very long distances between remote generation sites and load centers and the longitudinal grid lead to voltage control problems. In particular, the important variation of the power flows in the random sampling induces highly variable reactive losses and hence voltage drops, which may prevent the load flow computation from converging properly, thus leading to a low rate of accepted operating points.

Further, to represent normal operating conditions the reactive compensation needs to be adapted automatically to the power flows. This means switching shunt reactors in the UHV system and capacitor banks on lower voltage levels. Thus, an *automatic reactive compensation* loop was developed and included into the RP600 load flow program used for this study.

In spite of this improvement, the first random samplings yielded a very high percentage (up to 70%) of diverging load flow computations. To be able to analyze the physical or algorithmic reasons for such high divergence ratios, various frequency diagrams were drawn for the a priori data bases, corresponding to the specifications of the randomly selected variants, classified as *diverging vs converging*.

For example, Fig. 5.5 shows a typical frequency diagram, similar to those obtained in the first a priori data base obtained. The proportion of converging and diverging load flow computations is represented in terms of the specified values of the power flow in the Manic-Québec corridor. One can see that only a small proportion of states did actually converge, and it appears clearly from the diagram that the cases of divergence predominate mainly for power flows below 8,000MW. The reason is that the base case solutions used to initialize the load flow computation were too far away from the solution.

Several iterations were required in order to obtain a satisfactory data base. To improve the convergence we have used a larger number of base cases and a heuristic strategy to choose the appropriate one for each operating point specification. Figure 5.6 reproduces the final distribution of the cases of load flow divergence in terms of the Manic-Québec power flow. With respect to the diagram of Fig. 5.5, one can observe that the proportion of divergences is strongly reduced and they are more or less uniformly distributed. Thus the filtering does not affect the quality of the a posteriori data base.

The generation of the of the 12,500 operating points of the final data base, took about one week using 30% of the CPU of a Sun Sparc 10 workstation.

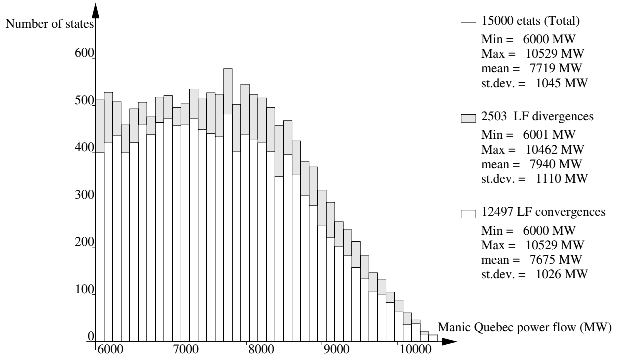


Figure 5.6 *Convergence diagram of Manic-Québec power flow (12 base case files)*

The operating points were then sent back to Montréal and fed into the LIMSEL data base in order to extract stability limits and classify them as stable or unstable; this information was put together with the ground parameters extracted by the load-flow to yield the final data base. The total amount of data (including the a priori data base) was about 20Mbytes, once compressed. Note that, at the time of the project and with the computing power available, using the time-domain simulation program to analyze the stability would have taken several months. We deemed that in order to assess the methodology, the information contained in the LIMSEL data base was sufficiently accurate. As an anecdote, we mention that during data base validation we detected 30 states for which LIMSEL provided erroneous limit values. It was found out to be due to a transcription error in the LIMSEL limit tables.

Below we will comment briefly on some automatic learning results extracted from this large data base. Further results are provided in [WHP95a, WHP$^+$95b].

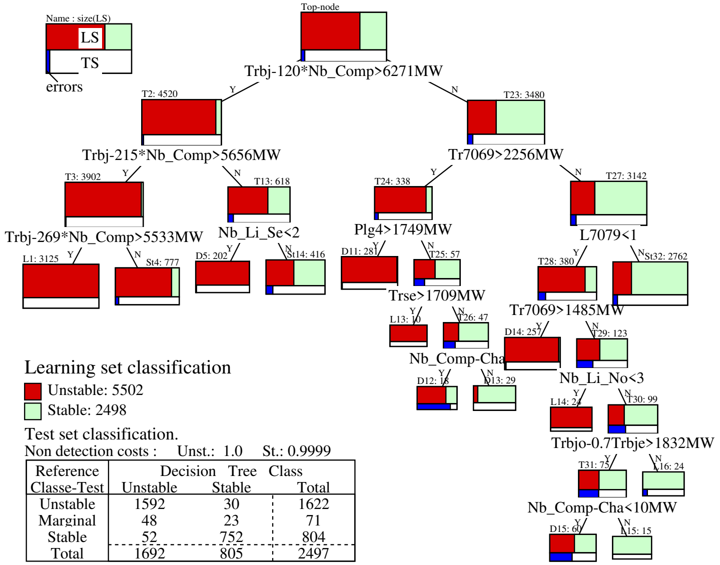
Automatic learning results

The tree partially represented in the right hand part of Fig. 5.7 was built on the basis of the first 10,000 states of the data base and 87 candidate attributes (power flows and generations, topology indicators, var support), including four linear combination attributes. All in all, it comprises 57 test nodes and 58 terminal ones. It has identified among the candidate attributes the 24 most relevant ones. Among others, at several test nodes (including the topnode) it has selected a linear combination of the total power flow "Trbj" in the James' Bay corridor and the number of SVCs in operation "Nb_Comp" which thus confirms prior knowledge. Thus, the threshold values of "Trbj" are functions of "Nb_Comp". For example, if "Nb_Comp"=12, the leftmost terminal node "L1" in Fig. 5.7 corresponds to a limit value of

$$\max\{6271 + 12 * 120; 5656 + 12 * 215; 5533 + 12 * 269\} = 8{,}761\text{MW},$$

above which a state with 12 SVCs in operation is unconditionally declared unstable, meaning that there is at least one line-fault in the James'-Bay corridor which would lead to loss of synchronism.

To evaluate its generalization capability, the tree was tested on the basis of the 2,500 states of the data base not used for its building, yielding an overall error rate of 4.3%. Out of the 1,622 fairly unstable states, only 30 are classified as stable yielding 1.85% "dangerous" errors. On the other hand, 23 marginally unstable states are classified stable, leading to small non-detection errors. There are also 52 false alarms, i.e. stable test states classified unstable by the tree.



Learning set classification

🟥 Unstable: 5502

🟩 Stable: 2498

Test set classification.

Non detection costs : Unst.: 1.0 St.: 0.9999

Reference Classe-Test	Decision Tree Class Unstable	 Stable	 Total
Unstable	1592	30	1622
Marginal	48	23	71
Stable	52	752	804
Total	1692	805	2497

Figure 5.7 *Global decision tree for the Hydro-Québec system (partial view)*

Table 5.1 *KNN results for the Hydro-Québec system*

K	1	3	5	7	9
67 candidate attributes					
P_e% (TS)	12.58	11.33	10.53	10.21	10.25
24 attributes of DT of Fig. 5.7					
P_e% (TS)	6.93	6.73	6.13	6.13	6.61

To improve accuracy, the same data base was further exploited by building a multilayer perceptron (with a single hidden layer of 20 neurons) on the basis of the same 10,000 learning states. Note that in this case we don't use a security margin as output, no such information being available. Thus the output information of the MLP is in the form of a 0/1 encoding of the security class. At convergence, the MLP yields a reduced test set error rate of 2.4%.

In terms of computational requirements we mention the following CPU times determined on a SUN Sparc10 workstation : 1 hour for the decision tree building and 1 second for testing the 2,500 test states; 60 hours for the learning of the MLP weights and 10 seconds for testing it.

Table 5.1 shows the accuracy results obtained with the KNN classifier for two different cases. The first line of results corresponds to the use of all 67 attributes in the distance computation[1]. The results are quite disappointing with respect to the decision tree and the multilayer perceptron. We note that the value of $K = 7$ provides the best results. The second line of results corresponds to using only the attributes identified by the decision tree of Fig. 5.7 : the reliabilities are significantly improved with respect to the preceding ones but the level of performance of the best DTs or MLPs is not reached; here again the value of $K = 7$ yields the best results. The well-known high sensitivity of the nearest neighbor to the attributes used in the distance computation (and more generally to the weights used in the distance) is observed here very clearly.

[1] The attribute values are however normalized.

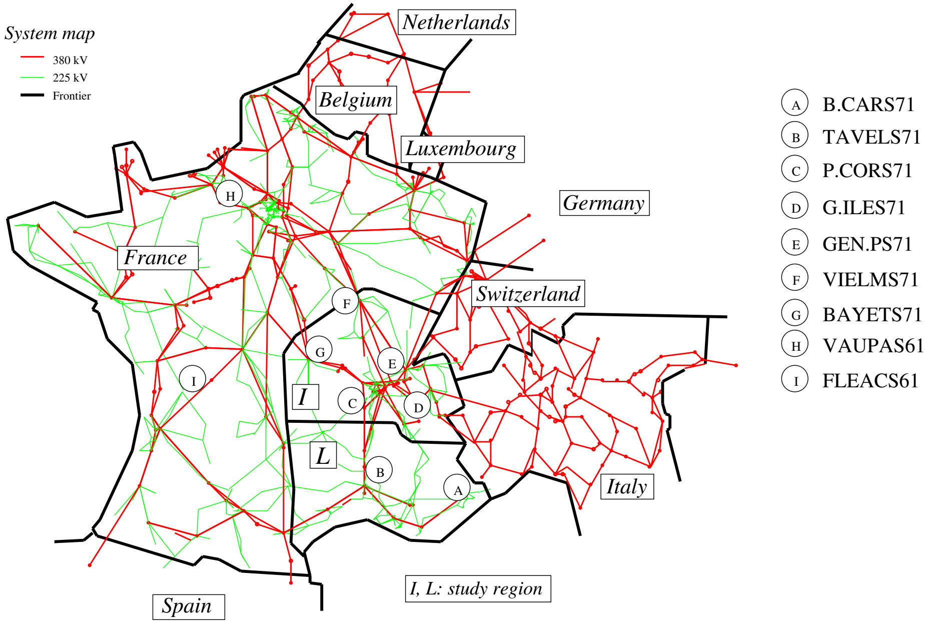


Figure 5.8 *One-line diagram of the study region and surroundings*

5.6.2 Electricité de France (breakdown scenarios)

In order to further illustrate the diversity of problems to which the automatic learning framework may be applied, let us briefly describe the data base generated within a recent research project [WLTB97a]. The long term goal of this research is to develop a global probabilistic approach to analyze and improve the dynamic performance of power systems in extremely disturbed modes, i.e. under circumstances where various fast and slow dynamic phenomena and their corresponding special protection schemes tend to interact yielding cascades of very intricate behaviors.

Specifications

In July 1995 a research collaboration was started between the University of Liège and Electricité de France, to apply the automatic learning framework to a case study on the EDF system. EDF experts defined a study region in the Provence/Alpes/Riviera subsystem, which was already known to present rather diverse failure modes : cascading line trippings, plant and area mode loss of synchronism, voltage collapse.

This South-Eastern part of the EDF system (see Fig. 5.8) is generally exporting large amounts of power to the rest of France and towards foreign countries (Italy, Switzerland and indirectly Spain). In the very extreme South-Eastern part it is weakly meshed and deficient in generation, thus liable to voltage collapse phenomena. This subsystem is already equipped with various automatic emergency control systems, in order to mitigate various types of failure modes.

It was decided to focus the study on the effect of multiple disturbances and abnormal operation of protection systems, which are often the causes of power system failures. On the other hand, in order to reduce the amount of software developments for the data base generation, it was decided to choose three operating points in a manual fashion. Then the random sampling specifications were set up for the EDs and MHs.

To enable the simulation of both fast and slow phenomena, while taking into account the operation of the relevant protections and special stability control systems, a rather detailed dynamic model was first set up. This model, comprising all in all more than 11,000 state variables is described in [WLTB97a].

In order to be able to analyze in detail different modes of failure of the power system, it was decided to build up a data base containing temporal attributes, i.e. curves representing the variation in time of various quantities deemed relevant (see below). A specific curve interpolation routine was developed in order to extract these curves from the dynamic simulator output files, and the data mining software developed in previous researches was adapted so as to handle the resulting very bulky temporal data bases efficiently.

Dynamic modeling hypotheses

In order to take into account the effect of uncertainty and/or errors in protection settings (delays, thresholds...) their DMs were systematically randomized in the data base generation. Similarly, the load model was also randomized in order to represent uncertainty and variability of load behavior. Moreover, in each scenario there is a random selection of some protections which are supposed to mis-operate : untimely generator tripping for over/under frequency or voltage protections, untimely line tripping for overload protections, partial non-operation of under-frequency load shedding protections, in order to represent what is happening in real life.

External disturbances

They are composed of two consecutive contingencies in the study region. They are chosen randomly (probabilities reflecting more or less their relative likelihood in real life) among the following ones :

- **Generator loss:** loss of one unit (thermal or nuclear); loss of some units of the same plant (thermal and hydro units); loss of a plant.
- **Faults on lines:** temporary fault on a line, permanent fault on a line; permanent fault on parallel circuits; two temporary or permanent faults separated by about 100ms on geographically close lines (lightning storm).
- **Faults in substations:** (bus bar fault) fault inside a substation leading to the loss of part of the substation; fault outside (but near) the substation leading to the loss of a part of the substation; loss of the whole substation after a major fault inside.

The two external disturbances are applied in sequence within a rather short time slot (less than 10 minutes) and it is supposed that there is no operator action in between. Then, the scenarios are simulated during 40 minutes after the second external disturbance in order to evaluate consequences.

Ground parameters

About 800 temporal attributes are used to describe the scenarios in the results data base. They are :

- voltage (magnitude and phase angle) of defined buses (130 attributes);
- rotor angle, velocity, and acceleration as well as excitation current and mechanical power of defined units (315 attributes);
- total active and reactive load and mean voltage in defined load areas (54 attributes);
- mean transformation ratio of “on load tap changers” in defined areas (13 attributes);
- equivalent voltage angle and frequency for the regions of the defense plan (6 attributes);
- reactive generation of the units participating in secondary voltage control (19 attributes);
- active and reactive power flows of all 380 kV lines of the EDF system, and some important 225kV lines in the study region (234 attributes);
- a listing of the discrete events happening in the system during the simulation.

Some of these attributes are to be used in order to define the severity of scenarios, i.e. measure the *consequences* in terms of loss of load and generation. The others are to be used as input parameters to criteria for characterizing the dynamic behavior of the scenarios and predict their severity as accurately and as early as possible. Those among these latter which are found to be the most informative (upon applying automatic learning to the data base) to predict the future behavior of a scenario in terms of its severity could then be used in order to monitor the system in real-time, together with appropriate decision rules extracted from the data base.

Data base generation and validation

A first preliminary data base of a few hundred scenarios was generated in early 1996, and, in August 1996 the generation of the final data base was started, using the tool described in §5.4. End of October 1996 the total number of scenarios simulated was about 1500, out of which about 100 were rejected.

The scenarios were simulated by Eurostag [MS92]; this simulation program is used in dynamic security assessment studies at EDF and, with its variable integration time step, is able to simulate slow dynamic phenomena (e.g. voltage collapse) as well as faster ones (e.g. loss of synchronism). The simulations were carried out in parallel on a cluster of 12 (HP 700) workstations available at night and during the week-ends.

In order to fix ideas about the computational involvement, let us mention that in the mean a single scenario simulation required about 11 hours and produced about 200MBytes of raw output. The total amount of data extracted for the 1400 scenarios of the a posteriori data base is of 1.5GBytes, compressed.

Table 5.2 provides a first glance at the diversity of the information in the data base.

Automatic learning

The proper exploitation of the data base is in progress. Investigations are under way in order to take the best advantage of such temporal data bases by automatic learning. In particular, adaptation of decision tree induction and clustering techniques to handle temporal attributes seems very promising.

Table 5.2 *Statistics of the result data base. Taken from [WLTB97b]*

Salient scenario characteristics	Min	Max	Mean	σ
CPU simulation time (s)	0	99000	38000	22000
time steps before interpolation	89	46800	3800	3071
time steps after interpolation	4	1728	145	137.2
size (MB) before interpolation	4	2140	174	140
size (kB) after interpolation and compression	32.4	3720	840	460
Number of lines lost				
380 kV	0	48	5	6.3
225 kV	0	149	9.6	18.53
Thermal units				
Number of units lost	0	15	1.15	2.1
mechanical power lost (MW)	0	13000	617	1213
mechanical power variation (EDF system, MW)	-20800	546	-1265	2445
Hydro units				
Number of units lost	0	32	2.7	5.3
mechanical power lost	0	2952	271	584
mechanical power variation (EDF system, MW)	-3039	60.5	-332	604
Load variation (MW)				
I region	-9046	654	-864	1714
L region	-8944	288	-1194	2368
EDF system	-22000	426.8	-2417	4323
Exportation variation (MW)				
EDF system to Belgium	-1527	1568	22	460
EDF system to Germany	-1832	1607	17	476
EDF system to Switzerland	-3301	830	-150	400
EDF system to Italy	-2974	1609	-48	463
EDF system to Spain	-1644	1253	-148	435
EDF system to all foreign systems	-8470	4946	-305	1626
Voltage at some buses at end of simulation (see Fig. 5.8)				
B.CARS71 (380 kV)	0	424	327	137
TAVELS71 (380 kV)	0	467	366	98
P.CORS71 (380 kV)	0	463	394	60
G.ILES71 (380 kV)	0	487	395	71
GEN.PS71 (380 kV)	0	469	395	57
VIELMS71 (380 kV)	0	449	397	45
BAYETS71 (380 kV)	0	416	387	55
VAUPAS61 (225 kV)	0	288	211	59
FLEACS61 (225 kV)	0	255	235	36

6

Conclusions

6.1 Summary

We sum up by first restating our intentions in giving this tutorial, on the emerging technology of automatic learning and its application to power system dynamic security assessment.

We have first considered a *tool box of automatic learning approaches*, starting with an intuitive description of complementary methods in the context of dynamic security assessment, and further discussing the main technical questions which underly their proper use. Some of the more mathematically involved subjects in automatic learning have not been covered, for the sake of time and clarity. We hope that the bibliographical references given hereafter will help the interested reader to find his way in the very rich, but sometimes confusing literature on the subject.

Given the topic of the tutorial, our choice of automatic learning methods was on purpose biased towards those which we found to be useful in DSA applications. Nevertheless, we deem that they present some broader interest, in particular in the many other potential applications in power systems [Weh97]. To be brief, let us only mention as other possible applications load prediction, equipment and plant monitoring, and design of equivalent models [WP96a].

The second topic of the tutorial discussed *practical application concerns* to power system dynamic security assessment. Thus we have screened the diversity of such problems and practical DSA environments in order to highlight possible uses. We have also discussed in detail the technical aspects of building security information data bases, further illustrated by practical case studies. These paramount but very time consuming aspects are generally hidden in the literature on the subject.

Maybe it can now be understood why the application of automatic learning to DSA, which was envisioned almost thirty years ago, starts only today being applied in real life. We think that there are several “complementary” reasons to this state of affairs.

Of course, as we mentioned in the first chapter, technology was not mature enough thirty years ago.

Then, ten years ago, when technology became mature enough, there was no methodology, and it took a few additional years of research to come up with what he have presented here.

The last, maybe most difficult obstacle is to convince utilities of the usefulness of this apparently very bulky framework. Indeed, to adopt the methodology needs to change the way of thinking and of carrying out security studies, and this is possible only with a strong enough motivation. Even in the case of those utilities which have already started applying these techniques (e.g. Electricité de France), we believe that it will take some further years before the methodology will be used in a fully systematic way in all the places where it has already shown to be useful, not to say in the other contexts.

Thus, we hope that to those who have attended this course we have been able to show at least some of the practical possibilities of this very powerful framework.

The final decision to take advantage of the automatic learning framework to DSA lies in the hands of the utility engineers. We believe that the rapidly changing economical and technological contexts of today will probably encourage some far seeing ones to start considering this framework as an actual alternative to present day practice.

6.2 Next stage

Taking for granted that the automatic learning framework will see in the future numerous interesting applications to dynamic security assessment, let us review the main present challenges for research and development.

6.2.1 Management of uncertainties

In the last few years of research we became more and more convinced that methodologies able to manage uncertainties properly are becoming a major need in power systems. In order to draw the best out of the automatic learning framework within this context, two further requirements need to be met.

The first one concerns data needed as input to the data base generation. In particular, we can only encourage utilities to collect all kinds of statistical information (e.g. concerning fault occurrences, weather conditions, device failures, operating conditions met in real life. . .) and put these into data bases available in the study environments. While there will always remain some holes to fill in a subjective way, the better the available information the more effective the security strategies that can be designed.

The other one concerns the improvement of methodologies in order to exploit the data so as to determine probabilistic risk levels. While much work has already been carried out in probabilistic security assessment, this is a very broad topic admittedly also needing further research [Cigré97b].

6.2.2 Temporal information

From a more technical point of view, our recent researches suggest that in many of the most interesting applications (e.g. emergency control system design) security information data bases will contain a great deal of temporal attributes.

However, while modern automatic learning methods are good at exploiting non temporal (scalar) attributes they are clumsy with temporal data. In addition, since temporal data bases are typically two orders of magnitude larger than non temporal ones, computational problems may become intractable without parallel computations.

Thus, further research is needed to develop new algorithms for temporal data or to adapt existing ones, and make them work effectively on very large scale data bases.

6.2.3 Software environments

In terms of software development the next stage is to design a comprehensive set of industry grade tools for the application of automatic learning.

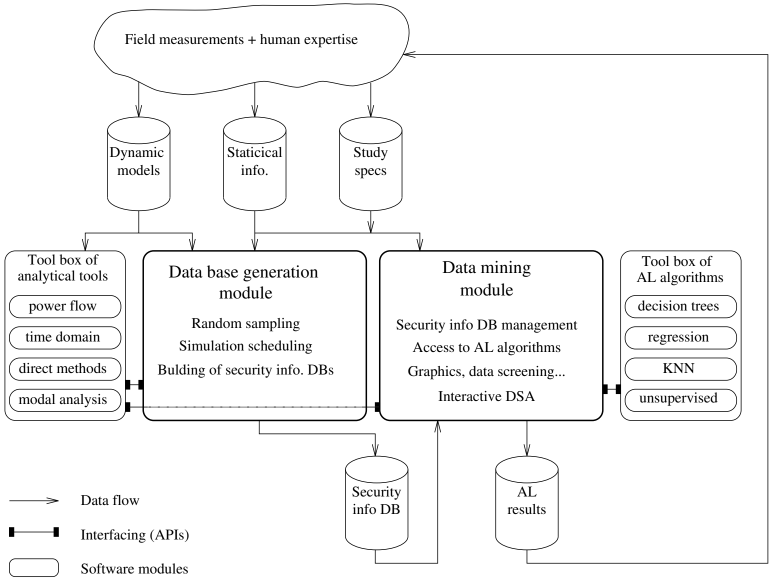


Figure 6.1 *Envisioned software architecture for the application of automatic learning to DSA*

Figure 6.1 depicts the envisioned overall software architecture for DSA studies by automatic learning. In this architecture the data base generation module as well as the automatic learning module will exploit trivial parallelism of their algorithms to take the best advantage of existing and future computing environments. They should be designed in a modular and "open" fashion so as to allow the easy integration of power system security analysis software and new automatic learning algorithms as soon as they become available.

With such a software platform one could apply the automatic learning framework efficiently and systematically in the planning and operating environments where security studies are presently carried out.

Bibliography

Automatic learning

[Bar93] A. R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Trans. on Info. Theory*, 39(3):930–945, May 1993.

[BFOS84] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth International (California), 1984.

[DH73] R. O. Duda and P. E. Hart. *Pattern classification and scene analysis*. John Wiley and Sons, 1973.

[FPSSU96] U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy. *Advances in Knowledge Discovery and Data Mining*. AAAI Press/MIT Press, 1996.

[Fri87] J. H. Friedman. Exploratory projection pursuit. *Jour. of the Am. Stat. Ass.*, 82(397):249–266, March 1987.

[FS81] J. H. Friedman and W. Stuetzle. Projection pursuit regression. *Jour. of the Am. Stat. Ass.*, 76(376):817–823, December 1981.

[FSS84] J. H. Friedman, W. Stuetzle, and A. Schroeder. Projection pursuit density estimation. *Jour. of the Am. Stat. Ass.*, 79(387):599–608, September 1984.

[Gau26] K. F. Gauss. *Theoria combinationis observatorionum errroribus minimis obnoxiae*. Dietrich, Göttingen, 1826.

[Hay94] S. Haykin. *Neural networks. A comprehensive foundation*. IEEE Press, 1994.

[HMS66] E. B. Hunt, J. Marin, and P. J. Stone. *Experiments in Induction*. Wiley, 1966.

[Koh90] T. Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, September 1990.

[Lap10] P. S. Laplace. *Mémoire sur les approximations des formules qui sont des fonctions de très grands nombres et sur leur application aux probablités*. Mémoires de l'Académie des Sciences de Paris, 1810.

[MP43] W. S. McCulloch and W. Pitts. A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133, 1943.

[Qui93] J.R. Quinlan. *C4.5. Programs for Machine Learning*. Morgan Kaufman, 1993.

[Vap95] V. N. Vapnik. *The nature of statistical learning theory*. Springer Verlag, 1995.

[Wol94] D. H. Wolpert, editor. *The Mathematics of Generalization*. Addison Wesley, 1994. Procs. of the SFi/CNLS Workshop on Formal Approaches to Supervised Learning.

Security assessment

[Cigré97a] CIGRE TF 38.02.13 (B. Meyer, convenor) New trends and requirements for dynamic security assessment. *To be submitted to ELECTRA*, 1997.

[Cigré97b] CIGRE TF 38.03.12 (R. Marceau and J. Endrenyi, convenors). Power system security assessment : a position paper. *To be submitted to ELECTRA*, 1997.

[Dy 68] T. E. Dy Liacco. *Control of Power Systems via the Multi-Level Concept*. PhD thesis, Sys. Res. Center, Case Western Reserve Univ., 1968. Rep. SRC-68-19.

[FC78] L. H. Fink and K. Carlsen. Operating under stress and strain. *IEEE Spectrum*, 15:48–53, March 1978.

[KM97] P. Kundur and G. K. Morisson. A review of definitions and classification of stability problems in today's power systems. *Panel session on Stability Terms and Definitions, IEEE PES Winter Meeting*, 1997.

[MS92] B. Meyer and M. Stubbe. EUROSTAG, a single tool for power system simulation. *Transmission and Distribution International*, 3(1):47–52, 1992.

Automatic learning applied to security assessment

[BW95] X. Boyen and L. Wehenkel. Fuzzy decision tree induction for power system security assessment. In *Proc. of SIPOWER'95, 2nd IFAC Symp. on Control of Power Plants and Power Systems*, pages 151–156, Mexico, December 1995.

[BW96] X. Boyen and L. Wehenkel. Automatic induction of continuous decision trees. In *Proc. of IPMU'96, Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 419–424, Granada, July 1996.

[Dil91] T.S Dillon. Artificial neural network applications to power systems and their relationship to symbolic methods. *Int. J. of Elec. Power and Energy Syst.*, 13(2):66–72, April 1991.

[DL96] T. E. Dy-Liacco. On the applicability of automatic learning to power system operation. *IEEE Computer Applications in Power Systems*, May/June 1997.

[ESMA+89] M. A. El-Sharkawi, R. J. Marks II, M. E. Aggoune, D. C. Park, M. J. Damborg, and L. E. Atlas. Dynamic security assessment of power systems using back error propagation artificial neural networks. In *Proc. of the 2nd Symposium on Expert Systems Application to power systems*, pages 366–370, 1989.

[FKCR89] R. Fischl, M. Kam, J.-C. Chow, and S. Ricciardi. Screening power system contingencies using back propagation trained multi-perceptrons. In *Proc. of the IEEE Int. Symposium on Circuits and Systems*, pages 486–494, 1989.

[GEA77] C. L. Gupta and A. H. El-Abiad. Transient security assessment of power systems by pattern recognition - a pragmatic approach. In *Proc. IFAC Symp. on Automatic Control and Protection of power systems*, 1977.

[HCS94] N. D. Hatziargyriou, G. C. Contaxis, and N. C. Sideris. A decision tree method applied to on-line steady-state security assessment. *IEEE Trans. on Power Syst.*, 9(2):1052–1061, 1994.

[HWP95] I. Houben, L. Wehenkel, and M. Pavella. Coupling of K-NN with decision trees for power system transient stability assessment. In *IEEE Conference on Control Applications*, pages 825–832, Albany (NJ), 1995.

[HWP97] I. Houben, L. Wehenkel, and M. Pavella. Genetic algorithm based k nearest neighbors. In *Proc. CIS-97, IFAC Conf. on Contr. of Indust. Syst.*, Belfort, Fr, 1997.

[JWVP95] Y. Jacquemart, L. Wehenkel, T. Van Cutsem, and P. Pruvot. Statistical approaches to dynamic security assessment: The data base generation problem. In *Proc. of SIPOWER'95, 2nd IFAC Symp. on Control of Power Plants and Power Systems*, pages 243–246, December 1995.

[KAG96] T. Kostic, J. J. Alba, and A. J. Germond. Optimization and learning of load restoration strategies. In *Proc. of PSCC'96*, Dresden, 1996.

[MK95] J. D. McCalley and B. A. Krause. Rapid transmission capacity margin determination for dynamic security assessment using artificial neural networks. *Electric Power Systems Research*, 1995.

[MT91] H. Mori and Y. Tamura. An artificial neural-net based approach to power system voltage stability. In *Proc. of the 2nd Int. Workshop on Bulk Power System Voltage Phenomena - Voltage Stability and Security*, pages 347–358, August 1991.

[NG91] D. Niebur and A. Germond. Power system static security assessment using the Kohonen neural network classifier. In *Proc. of the IEEE Power Industry Computer Application Conference*, pages 270–277, May 1991.

[OH91] D. R. Ostojic and G. T. Heydt. Transient stability assessment by pattern recognition in the frequency domain. *IEEE Trans. on Power Syst.*, PWRS-6(1):231–237, 1991.

[PDB85] Y. H. Pao, T. E. Dy Liacco, and I. Bozma. Acquiring a qualitative understanding of system behavior through AI inductive inference. In *Proc. of the IFAC Symp. on Electric Energy Systems*, pages 35–41, 1985.

[PPEAK74] C. K. Pang, F. S. Prabhakara, A. H. El-Abiad, and A. J. Koivo. Security evaluation in power systems using pattern recognition. *IEEE Trans. on Power Apparatus and Systems*, 93(3), 1974.

[RKTB94] S. Rovnyak, S. Kretsinger, J. Thorp, and D. Brown. Decision trees for real-time transient stability prediction. *IEEE Trans. on Power Syst.*, 9(3):1417–1426, August 1994.

[SP89] D.J. Sobajic and Y.H. Pao. Artificial neural-net based dynamic security assessment for electric power systems. *IEEE Trans. on Power Syst.*, PWRS-4(4):220–228, February 1989.

[Weh95] L. Wehenkel. *Machine Learning Approaches to Power System Security Assessment*. Faculté des Sciences Appliquées - Univerité de Liège, No 142, 400 pages, 1995.

[Weh97] L. Wehenkel. *Automatic learning techniques in power systems*. Kluwer Academic, 1997, *to appear*.

[WHP95a] L. Wehenkel, I. Houben, and M. Pavella. Automatic learning approaches for on-line transient stability preventive control of the Hydro-Québec system - Part II. A tool box combining decision trees with neural nets and nearest neighbor classifiers optimized by genetic algorithms. In *Proc. of SIPOWER'95, 2nd IFAC Symp. on Control of Power Plants and Power Systems*, pages 237–242, December 1995.

[WHP+95b] L. Wehenkel, I. Houben, M. Pavella, L. Riverin, and G. Versailles. Automatic learning approaches for on-line transient stability preventive control of the Hydro-Québec system - Part I. Decision tree approaches. In *Proc. of SIPOWER'95, 2nd IFAC Symp. on Control of Power Plants and Power Systems*, pages 231–236, December 1995.

[WLTB97a] L. Wehenkel, C. Lebrevelec, M. Trotignon, and J. Batut. A probabilistic approach to the design of power systems protection schemes against blackouts. In *Submitted for publication*, 1997.

[WLTB97b] L. Wehenkel, C. Lebrevelec, M. Trotignon, and J. Batut. A step towards probabilistic global dynamic security assessment. *Submitted*, 1997.

[WP93a] L. Wehenkel and M. Pavella. Advances in decision trees applied to power system security assessment. In *Proc. of APSCOM-93, IEE Int. conf. on advances in power system Control, Operation and Management (Invited)*, pages 47–53, December 1993.

[WP93b] L. Wehenkel and M. Pavella. Decision tree approach to power system security assessment. *Int. J. of Elec. Power and Energy Syst.*, 15(1):13–36, 1993.

[WP96a] L. Wehenkel and M. Pavella, editors. *Revue E - Special Issue on Automatic learning applied to power systems*. SRBE, Belgium, December 1996.

[WP96b] L. Wehenkel and M. Pavella. Why and which automatic learning approaches to power systems security assessment. In *Proc. of CESA'96, IMACS/IEEE SMC Multiconference on Computational Engineering in Systems Applications*, pages 1072–1077, Lille, Fr, July 1996.