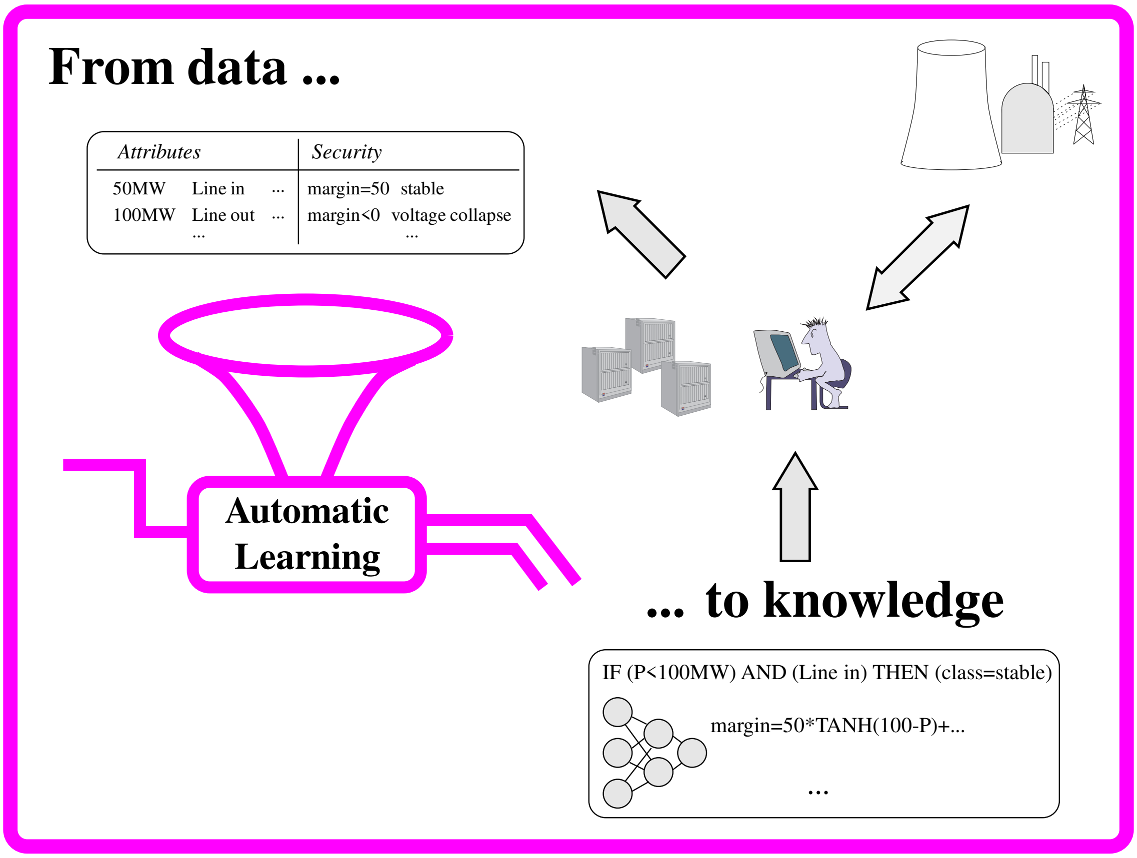
PICA'97 - IEEE POWER INDUSTRY
COMPUTER APPLICATIONS CONFERENCE

TUTORIAL COURSE ON

AUTOMATIC LEARNING METHODS
APPLICATION TO DYNAMIC SECURITY ASSESSMENT



Louis WEHENKEL
University of Liège - Belgium

Yannick JACQUEMART
Electricité de France - France

Final version of the course notes
April 1997

Contents

Conclusions

IV Copies of transparencies

Introduction

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]) 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 [DL68, 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 is generally approached in a pragmatic, power system and problem-specific way. Although there are many different more or less sophisticated tools, the most widely accepted one is numerical simulation. In planning or operations planning departments, engineers thus use numerical simulation and their own expertise to run some scenarios and extract, by hand, the relevant security information, and finally produce guidelines.

Thus, *the purpose of applying automatic learning to DSA is not to replace, but rather 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.

Further, 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 were 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.

Random sampling techniques are considered to screen all relevant situations in a given context, while existing numerical simulation tools are exploited - if necessary in parallel - to derive detailed security information.

The heart of the framework is provided by automatic learning methods used to extract and synthesize relevant information and to reformulate it in a suitable way for decision making. This consists of transforming the data base (DB) of case by case numerical simulations into a power system security *knowledge base* (KB). We will see that, due to their interpretability, decision trees are the cornerstone of the automatic learning tool box, the other methods being used for their complementary features according to the type of information they may exploit and/or produce.

The final step consists of using the extracted synthetic information (decision trees, rules, statistical or neural network approximators) either in real-time, for fast and effective decision making, or in the off-

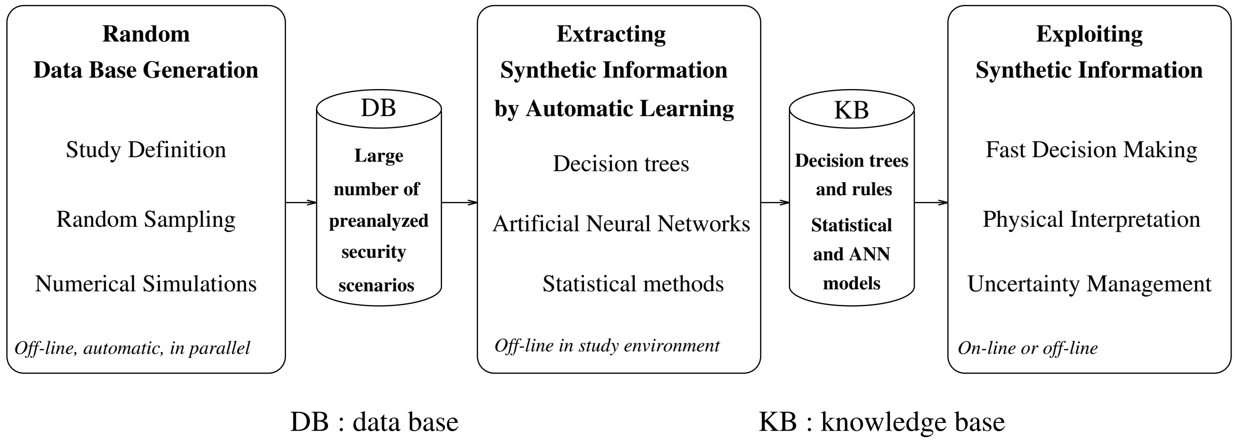


Figure 2.1 *Automatic learning framework for security assessment*

line study environment, so as to gain new physical insight and to derive better system and/or operation planning strategies.

2.1 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.

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.

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.

2.2 Application of automatic learning (Data mining)

Application of automatic learning (Data Mining (DM) for short) is the most interesting task. The idea is to exploit the capabilities of 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 in order to study some situations which were not sufficiently well represented in the original one, or in order to assess the effect of countermeasures suggested by the data mining 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.

2.3 Exploiting extracted information

Anticipating on the following chapters, let us briefly discuss how this framework may complement classical system theory oriented methods for security assessment.

In practice, there are three dimensions along which we may expect important fallouts.

First of all *computational efficiency*. By using synthetic information extracted by automatic learning, 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. Computational efficiency was actually the motivation of Dr Dy Liacco, when he first envisioned in the late sixties the use of pattern recognition for on-line DSA [DL68]. Even today, this remains a strong motivation.

But the synthetic information extracted by automatic learning methods may itself be complementary to and generally more powerful than that provided in a case by case fashion by existing analytical or numerical methods. In particular, much more attention is paid nowadays to *interpretability* and management of *uncertainties*, the two other important fallouts expected from automatic learning methods.

As concerns *interpretability*, the use of automatic learning to provide physical insight into the nonlinear system behavior was first proposed by Professor Pao and Dr Dy Liacco in the mid-eighties [PDB85]. In the meantime, it has been shown that machine learning (in particular, decision tree induction) is indeed an effective way to extract interpretable security rules from very large bodies of simulated examples [WP93b, WVP$^+$94]. We will see that extracted rules express explicitly 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 automatic learning framework allows one to tailor the resulting information to analysis, sensitivity analysis and control applications.

As concerns management of *uncertainties*, we will see that the above framework is indeed able to take into account and manage uncertainties on dynamic models, external systems, load behavior, measurements...

To conclude this introduction, we note that the above motivations for using automatic learning, although presented in the specific context of security assessment, are also applicable to many other application fields.

3

Scope of the tutorial

The material of this tutorial covers three complementary aspects.

In Part 1 we will describe a tool box of automatic learning algorithms. While there is a huge number of such methods, we have selected those which have shown to be useful and complementary in the context of DSA.

In Chapter 4, 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. Chapter 5 provides complementary material, in particular mathematical and algorithmic details of different automatic learning methods. A short glossary of terms and acronyms frequently used in the context of automatic learning is provided page 13 so as to facilitate reading.

Part 2 of the course concentrates on the application to large scale DSA problems.

In Chapter 6 we review security problems and security assessment environments and how automatic learning can be applied in the different contexts. Then, in Chapter 7, 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.

In the third part, we consider in some more detail the viewpoint of Electricité de France (EDF), an electric utility who has been active in this field since several years.

Chapter 8 presents the history of applications of automatic learning to DSA at EDF. Chapter 9 will consider an actual case study on voltage stability assessment in the EHV system of EDF, showing how practical difficulties may be solved, and highlighting interesting outcomes. Chapter 10 will present the present perspectives for various applications at EDF.

In the conclusions we scrutinize the main objectives of the tutorial, discuss further research and development needs, and sketch a comprehensive computer software architecture for the application of automatic learning to DSA.

A computer demonstration will be given in order to illustrate further the practical use of automatic learning methods.

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 one day 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.

Part I

Automatic learning methods

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)

4

An automatic learning tool box

Learning theory is the subfield of computer science dealing with inductive inference. Inductive inference derives general rules from sets of specific observations; it is widely used for scientific discovery. One of the concerns of the field is understanding biological - mainly human - learning capabilities. An other consists of designing algorithms to solve engineering problems. In our discussion we will restrict ourselves to the latter aspect, i.e. automatic learning.

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.

Note that we will make a second pass through the automatic learning methods in Chapter 5, in order to fix ideas about the mathematical and algorithmic details.

Note.

In order to explain ideas we use a very simple "toy" DSA problem (transient stability assessment of a one-machine infinite-bus system). This is, by no means, meant to be representative of actual large scale applications. However, it enables us to describe concepts while illustrating them on a simple DSA example, avoiding technical discussions about DSA itself. The detailed discussion of real life application concerns is the topic of the second and third parts of the course. We ask the reader to kindly wait until then.

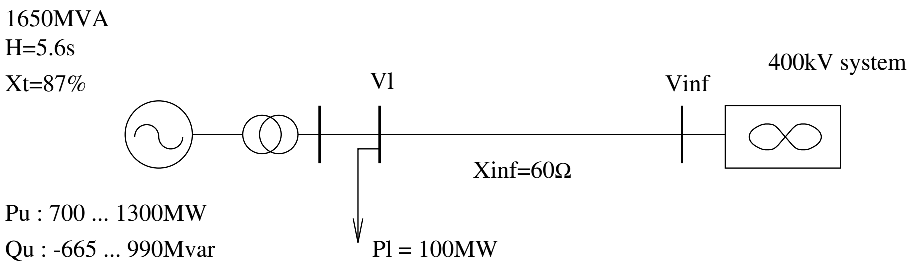


Figure 4.1 *One-machine infinite bus system*

4.1 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.

4.2 Illustrative transient stability problem

Figure 4.1 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

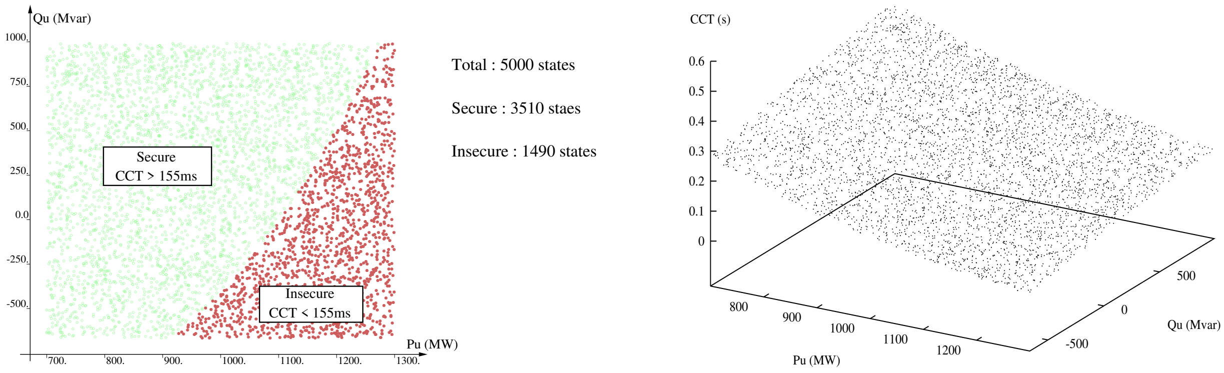


Figure 4.2 *Data base of 5000 random states of the OMIB system*

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. 4.1, 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 4.2 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).

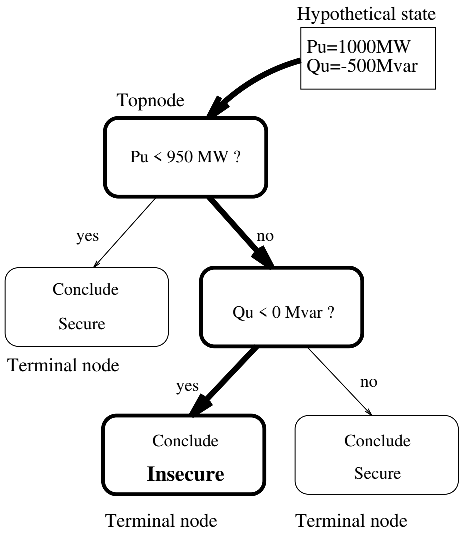
Learning set (LS). We use a random subsample of 3000 (Pu,Qu) states together with the output security information (class or CCT, as appropriate).

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

4.3 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].

[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.



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 4.3 *Hypothetical decision tree and equivalent if-then rules*

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 4.3 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. 4.3 translates into the rules indicated beneath it.

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 4.4 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. 4.4a. 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 4.4b 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.

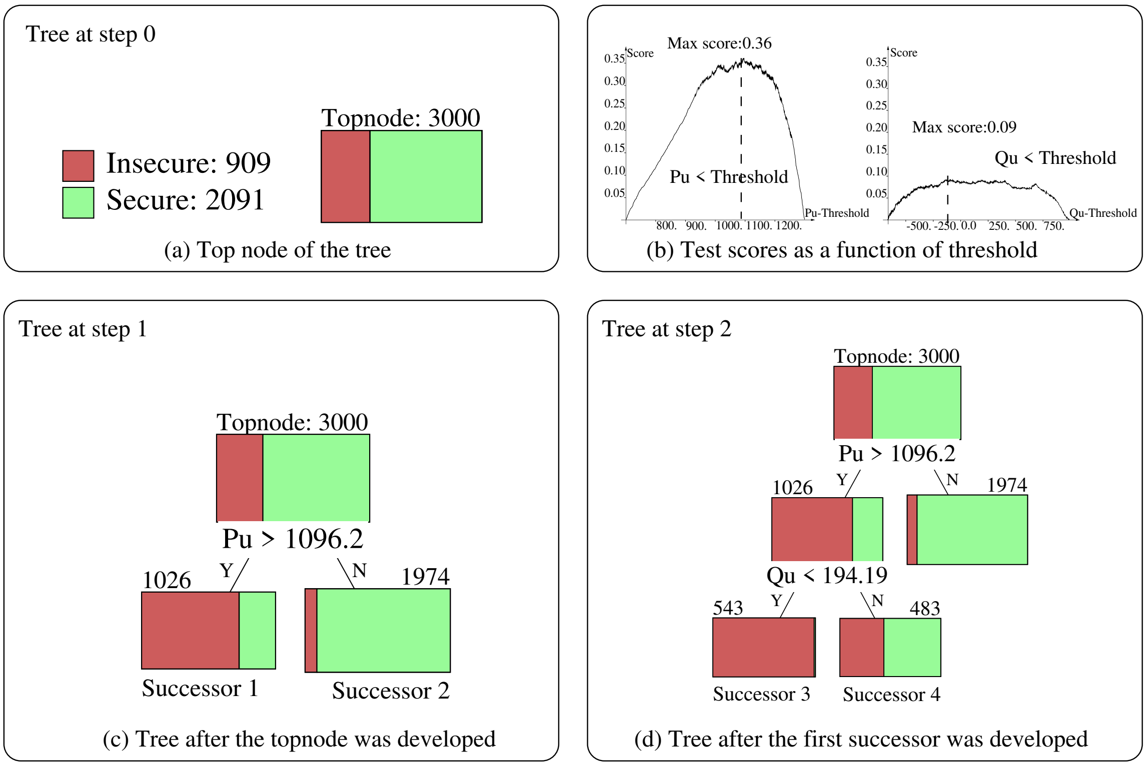


Figure 4.4 *Three first steps of decision tree growing*

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. 4.4c. Note that the number on 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. 4.4d, the procedure continues recursively to split the recently created successors, gradually separating the secure and insecure states until a stop splitting criterion is met. The stop splitting criterion decides whether a node should indeed be further developed or not. 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. 4.5 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

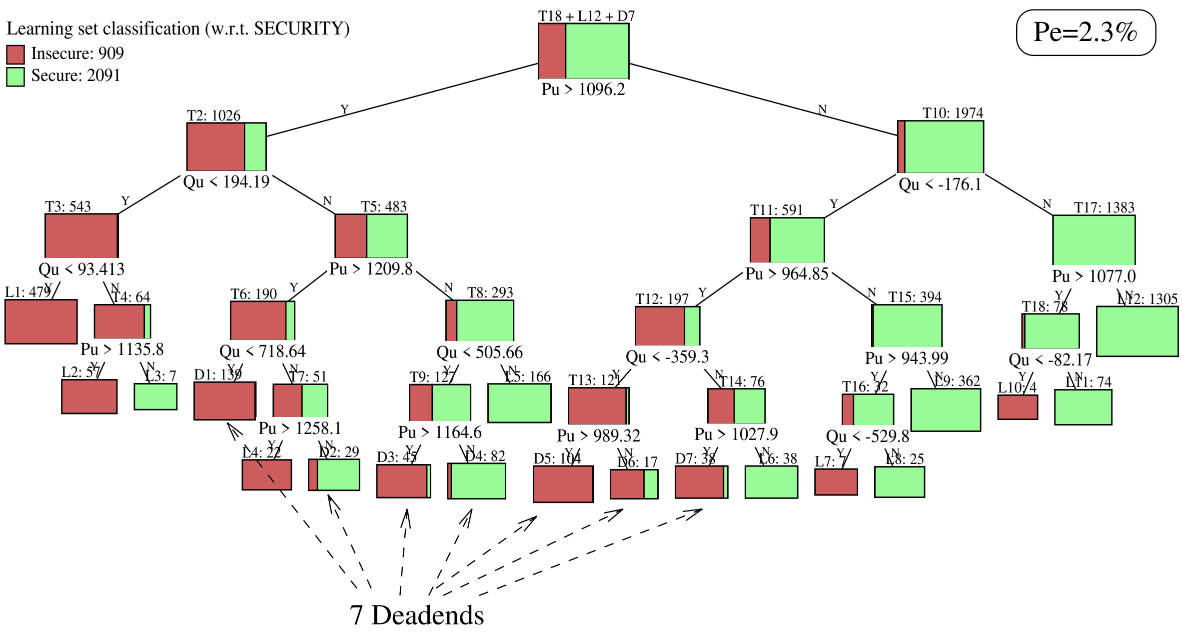


Figure 4.5 *"Orthogonal" decision tree (end result)*

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 dichotomous tests encountered to select the appropriate successor. When a terminal node is reached, the output majority class of the corresponding sub-learning-set stored there is retrieved and the test state is classified into this class. 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%.

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 4.6 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

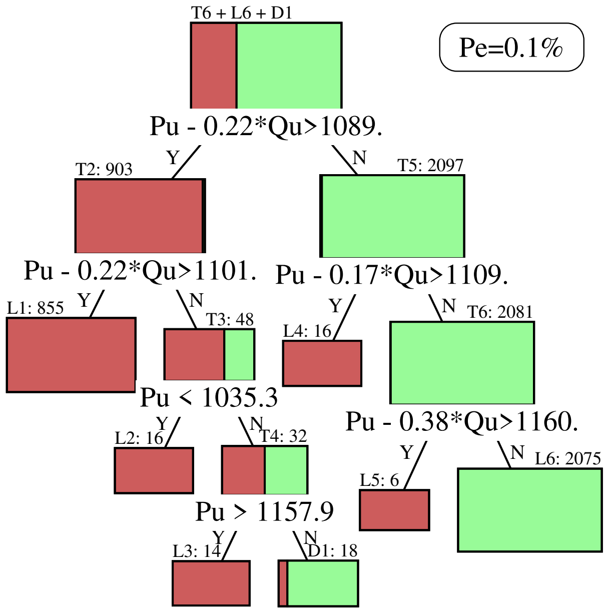


Figure 4.6 *"Oblique" decision tree*

"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. 4.5 (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 4.7 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 about a numerical output variable, and *fuzzy* trees which use fuzzy logic instead of standard logic to represent output information in a smooth fashion (see §5.3). Both approaches allow us to infer information about security margins, similarly to the techniques discussed below in §§4.4 and 4.5.

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 (see Chapter 5).

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

[2]CPU times on a SUN Sparc10 workstation.

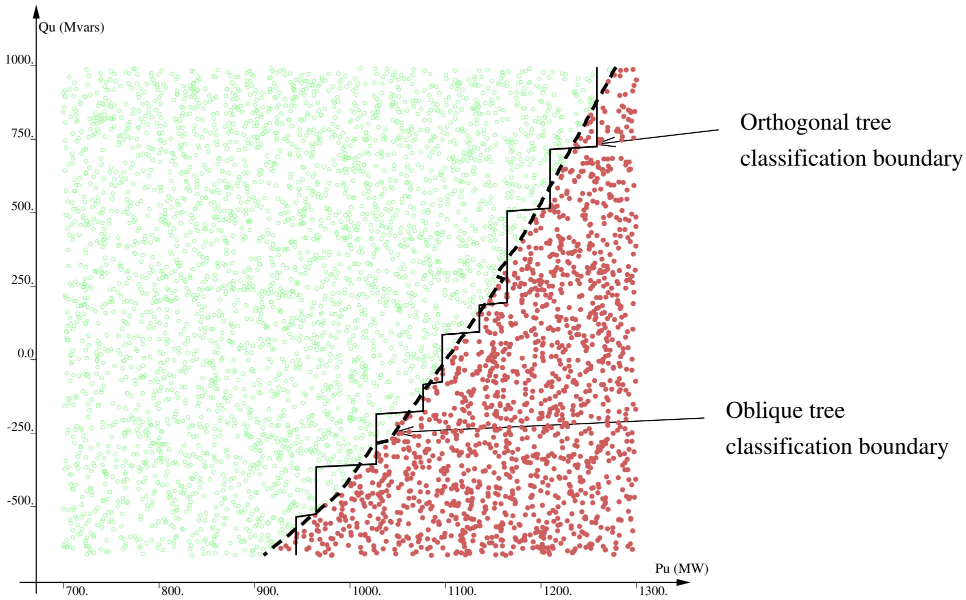


Figure 4.7 *Classification boundaries of the DTs of Figs. 4.5 and 4.6*

applications typically less than twenty percent of the candidate attributes are actually selected while growing a tree. This will be illustrated in Chapter 10 on a real-world example.

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.

4.4 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 (see Chapter 5).

Multilayer perceptrons

Work on artificial neural networks started several decades ago with the work on perceptrons. Figure 4.8 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

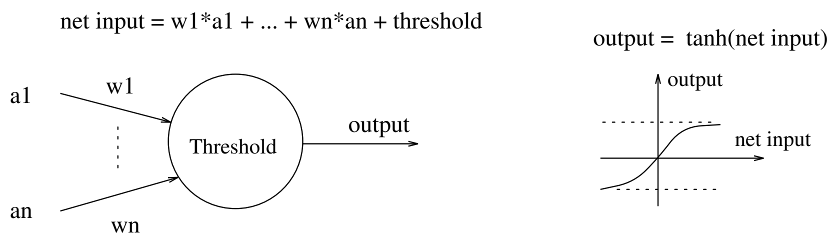


Figure 4.8 *Perceptron (neuron)*

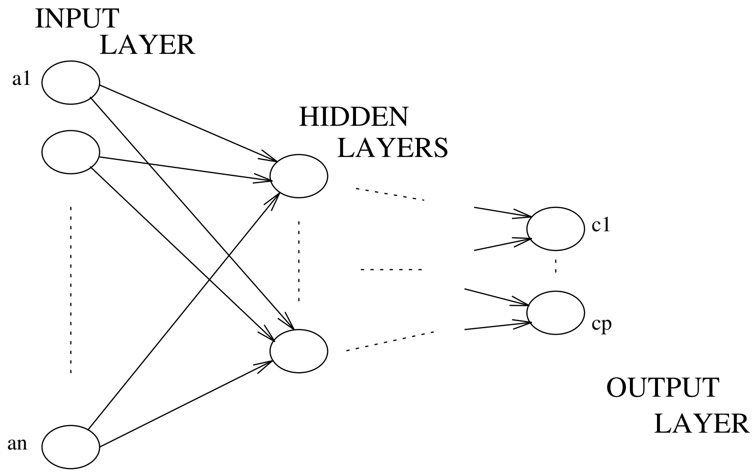


Figure 4.9 *Multilayer perceptron*

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.

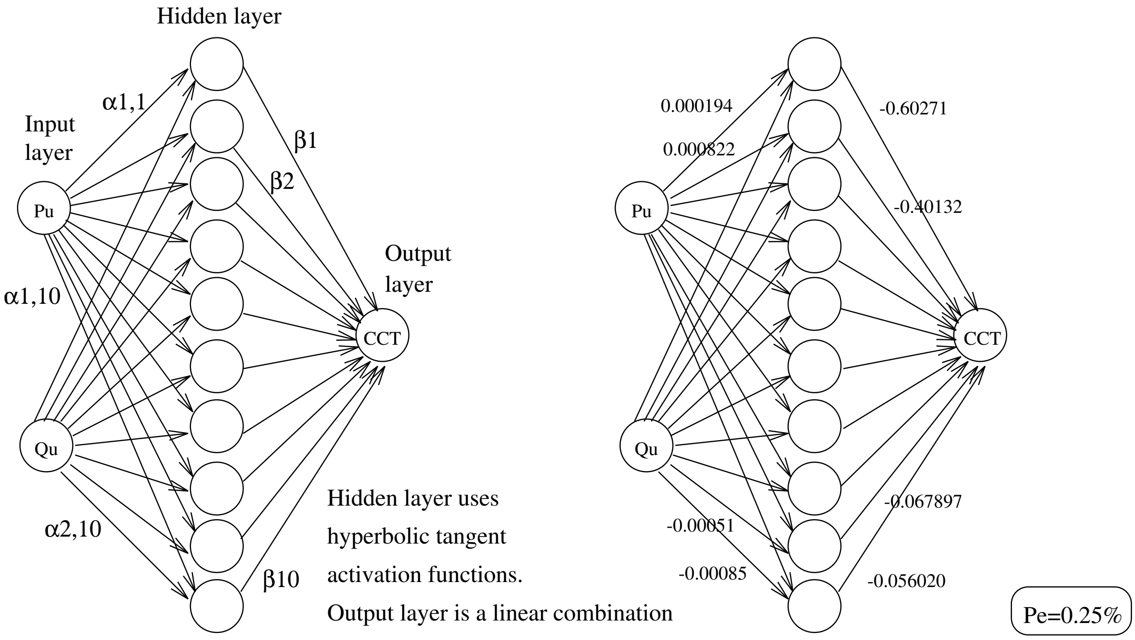
Figure 4.9 illustrates a typical multilayer perceptron. Each neuron is a perceptron : input layer neurons are fed with linear combinations of the input attributes; hidden and output layer neurons receive linear combinations of outputs from neurons in the preceding layers.

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, nowadays there exist various algorithms to determine the structure automatically, one of which is described in Chapter 5.

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.



(a) Initial values of the weights are random (b) After training the weights are tuned to fit the problem

Figure 4.10 *Single hidden layer perceptron*

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 (see §5.4). 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 4.10 graphically sketches the MLP that we have used, containing 10 hidden neurons. Each hidden neuron i has an input/output relationship of the form

$$\text{Output}_i(\text{state}) = \tanh(\alpha_{i1} Pu(\text{state}) + \alpha_{i2} Qu(\text{state}) + \theta_i), \tag{4.1}$$

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.

$$\text{CCT}_{\text{MLP}}(\text{state}) = \sum_{i=1\ldots10} \beta_i \tanh(\alpha_{i1} Pu(\text{state}) + \alpha_{i2} Qu(\text{state}) + \theta_i), \tag{4.2}$$

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)

$$\text{TSE} = \sum_{\text{state} \in LS} |\text{CCT}(\text{state}) - \text{CCT}_{\text{MLP}}(\text{state})|^2, \tag{4.3}$$

which is a smooth, although complex and 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

$$\begin{aligned}
\mathrm{CCT}_{\mathrm{MLP}} = & -0.602710 \tanh(0.000194 Pu - 0.00034 Qu - 0.93219) \\
& -0.401320 \tanh(0.000822 Pu - 0.00020 Qu - 0.76681) \\
& +0.318249 \tanh(0.000239 Pu - 0.00050 Qu - 0.29351) \\
& -0.287230 \tanh(0.002004 Pu - 0.00034 Qu - 1.20080) \\
& +0.184522 \tanh(0.000131 Pu - 0.00057 Qu - 0.03152) \\
& +0.177701 \tanh(0.001799 Pu - 0.00011 Qu - 2.08190) \\
& -0.150720 \tanh(0.001530 Pu - 0.00056 Qu - 1.68040) \\
& +0.142678 \tanh(0.002152 Pu - 0.00046 Qu - 1.72280) \\
& -0.067897 \tanh(0.001910 Pu - 0.00051 Qu - 1.71343) \\
& -0.056020 \tanh(0.000202 Pu - 0.00085 Qu - 0.39876)
\end{aligned} \tag{4.4}$$

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 4.11 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.

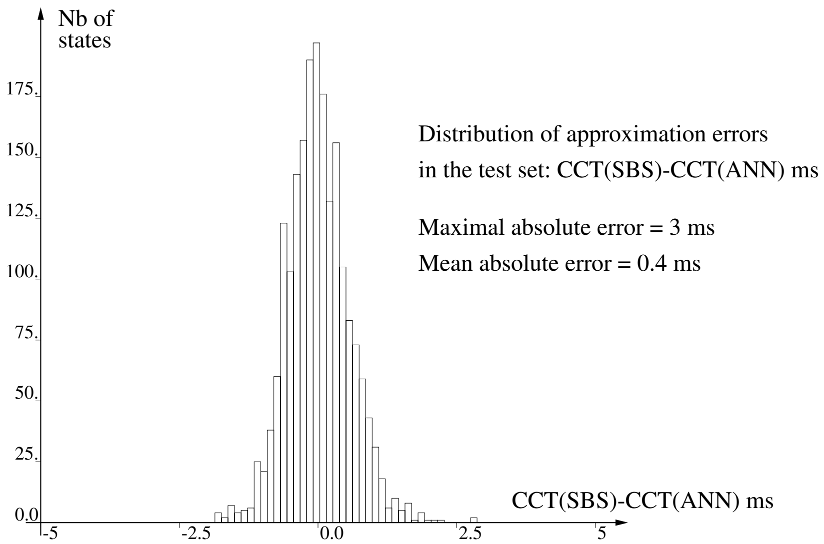


Figure 4.11 *Distribution of MLP approximation errors*

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 described in Chapter 5.

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.

4.5 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.

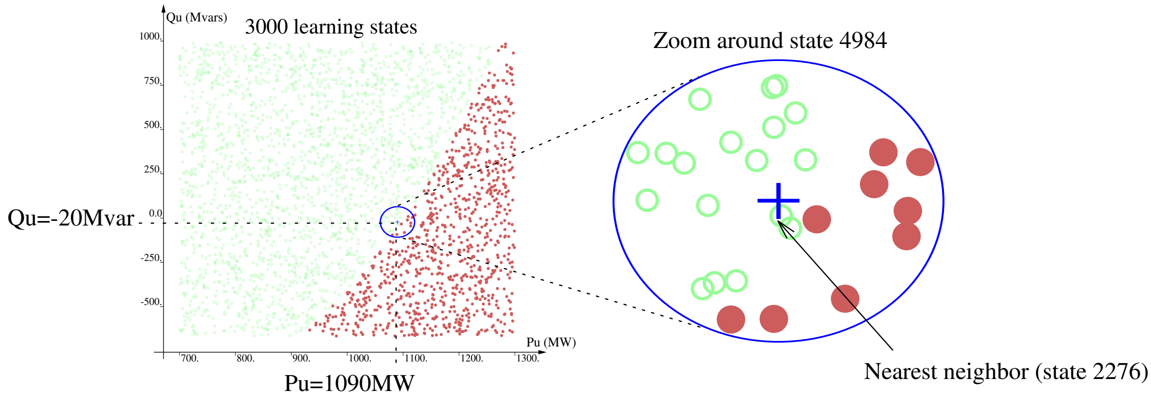


Figure 4.12 *Learning set of 3000 random states and nearest neighbors of state 4984*

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.

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 4.12 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. 4.12 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 4.13 shows the distribution of CCT approximation errors. Comparing with Fig. 4.11, 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.

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

[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.

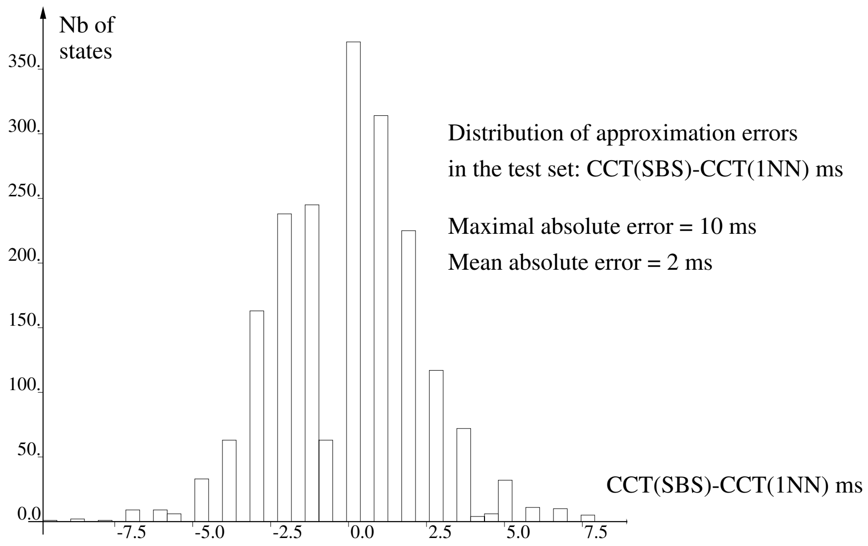


Figure 4.13 *Distribution of 1NN approximation errors*

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 §7.6.1, 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. One such method will be briefly described in Chapter 5.

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.

4.6 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

Table 4.1 *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

the other two types of techniques, thus improving their performances in terms of computing times and accuracy.

Table 4.1 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.

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.

4.7 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. We will have to wait until the presentation of a real-life power system security problem in Chapter 10 to provide an interesting illustration.

5

Technical aspects of supervised learning

In this chapter, we will provide some important methodological (mathematical and algorithmic) details relevant in automatic learning. Due to lack of time, only a small part of this material can be covered during the tutorial presentation.

Note. In this chapter we use the term model *to denote generically the information extracted by automatic learning (e.g. decision trees, multilayer perceptrons, nearest neighbor classifiers, ...).*

5.1 Main steps in automatic learning

In general, the application of automatic learning to a given practical problem is decomposed into the following subtasks [MST94].[1]

Representation consists of (i) choosing appropriate input attributes to represent the practical problem instances, (ii) defining the output information, and (iii) choosing a class of models suitable to represent input/output relations and (e.g. decision trees or multilayer perceptrons).

Feature selection aims at reducing the dimensionality of the input space by dismissing attributes which don't carry useful information to predict the considered output information.

Model selection (or learning per se) will typically identify in the predefined class of models the one which best fits the learning states. This generally requires choice of model structure and parameters, using an ad hoc search technique adapted to the considered type of model.

Interpretation and validation are very important in order to understand the physical meaning of the synthesized model and to determine its range of validity. It consists of testing the model on a set of unseen test examples and comparing its information with prior expertise about the problem.

Model use consists of applying the model to predict outputs of new situations on the basis of the values assumed by the input parameters, and if necessary to "invert" the model in order to provide information on how to modify input parameters so as to achieve a given output.

[1] In the context of a security information data base, these tasks may be carried out several times with different objectives, in order to extract many different types of informations from the data base.

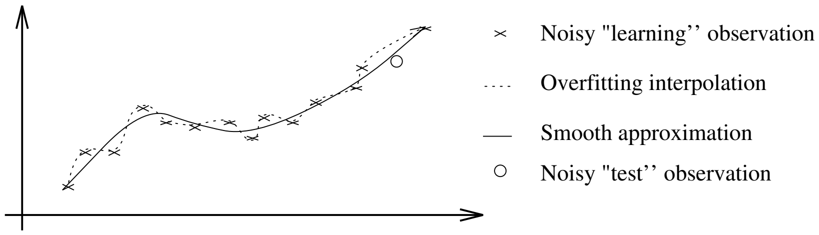


Figure 5.1 *Illustration of overfitting*

Solving the *representation problem* is normally left to engineer, although some methods are able construct features automatically. Thus choosing an appropriate set of candidate attributes is done in an iterative fashion, during the first trials of applying a learning algorithm to a new problem. Similarly, the choice of an appropriate model (or models) is done in trial and error fashion. Notice also that it may be necessary to preprocess the attribute values in order to apply a given learning algorithm. Preprocessing, includes scaling (e.g. pre-whitening) and filling in missing values.

The distinction between *feature selection* and *model selection* is somewhat arbitrary. For example some of the methods (in particular decision tree induction) actually solve these two problems simultaneously.

From the *interpretation and validation* point of view, we have already seen that some methods provide rather black-box information, difficult to interpret, while some others provide explicit and very transparent models, easy to compare with prior knowledge.

Finally, as far as the *use* of the model for fast decision making is concerned, we recall that speed variations of several orders of magnitude may exist between various techniques. This may reduce the usefulness of some methods in time-critical real-time applications.

5.2 Overfitting

Overfitting is a generic problem encountered in automatic learning. It generally appears when the model extracted is too complex with respect to the information provided in the learning set.

The complexity of a model measures the number of “parameters” in the model which are identified during learning. For example, the complexity of binary decision trees is proportional to the number of test nodes, the complexity of a multilayer perceptron is proportional to the number of weights, and the complexity of a nearest neighbor rule is proportional to the number of attributes used in the distance calculation.

A model which overfits the training set will be suboptimal in terms of generalization capabilities. Moreover, its parameters will present a large variance with respect to the random nature of the learning set, and the model will thus exploit irrelevant information, be uselessly cumbersome, and difficult to interpret. Figure 5.1 illustrates the overfitting phenomenon on a simple one-dimensional curve-fitting problem. For example, in spline approximation, if we use a too large number of parameters we may be able to fit the curve to the noisy learning data, but the interpolation/extrapolation on test states may become very poor. One can see that reducing the order of the approximation will actually allow us to improve the approximation.

The overfitting problem leads to the so-called bias/variance tradeoff, well known in statistics. If a model extracted by automatic learning is too simple it will present a large bias : the model is too rough, and at

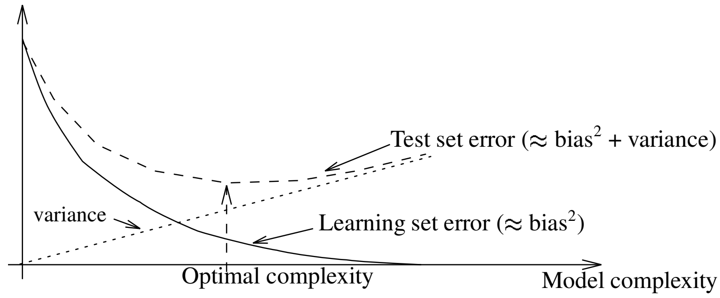


Figure 5.2 *Bias/variance tradeoff via cross-validation*

most points in the input space its (expected) output value will be very different from the desired output. If the learning algorithm is well designed, bias will decrease when the model complexity increases. At the same time, the model will depend in a stronger fashion on the random nature of learning samples, thus its variance will increase. Bias and variance lead both to generalization errors; it is thus necessary to reach a tradeoff (see Fig. 5.2).

In practice, bias is an increasing function of the physical problem complexity (and the type of automatic learning method used), not of the learning set size. Variance, on the other hand, is a decreasing function of the learning set size. Thus, the optimal model complexity will be an increasing function of both problem complexity and learning set size. While from a qualitative point of view this is well known since the early days of automatic learning, in the last few years theoretical work has lead to quantitative characterizations within different theoretical frameworks [Wol94, Vap95]. While a detailed discussion of these falls however out of the scope of this course, in §5.4 we provide some hints on the particular theory which was recently developed in the context of multilayer perceptrons.

While theory explains, at least qualitatively, the nature of the overfitting problem, in practice there are many different ways to fight against it. Some are specific to a particular type of method, some others are generic. One generic approach, the so-called cross-validation technique, consists of learning a sequence of models of growing complexity, then using an independent test set to evaluate their generalization capabilities and finally selecting the best one. This procedure is illustrated in Fig. 5.2. Its main drawbacks are computational overhead (one to two orders of magnitude) and data greediness.

Cross-validation may also be used in order to prune models of large complexity. The pruning starts with a very complex model, which is supposed to be overparametrized. Then, a sequence of models of decreasing complexity is obtained by progressively simplifying this model. E.g. decision trees are pruned by replacing test nodes by terminal ones [BFOS84, Weh93]; multilayer perceptrons are pruned by removing some connections (i.e. setting some weights to zero, and/or removing some neurons) and adapting the remaining weights [Hay94]. Cross-validation is used to select among the models of decreasing complexity the best one.

We will see below that the stop-splitting criterion used in decision trees is a very efficient and direct way to control the complexity, which doesn't require a tedious cross-validation.

To complete the topic, let us mention two other generic approaches to reduce overfitting. The first one is *regularization* which is is mainly used in the context of regression. It consists in modifying the TSE criterion in order to penalize models which are not smooth enough, e.g. by adding a term proportional to the model curvature [GJP95]. The second one is *model averaging* which consists in building several models and aggregate their outputs in the form of an average, thus reducing variance. In particular, Bayesian averaging consists in using the Bayesian framework to compute posterior model probabilities

Table 5.1 *Splitting of the data base by a test*

	Stability classes			
TRBJ$<$ 7308.5	Stable	Unstable	Total	
true	n_{11}=3234	n_{21}=2408	$n_{.1}$=5642	$H_{C\|T=true} = 0.984$
false	n_{12}=704	n_{22}=6151	$n_{.2}$=6855	$H_{C\|T=false} = 0.477$
Total	$n_{1.}$=3938	$n_{2.}$=8559	$n_{..}$=12497	$H_C = 0.899$
			$H_T = 0.993$	

and in averaging the models according to the latter [BW91, Bun92, Weh97].

5.3 Decision trees

In this section we provide the technical details concerning decision trees. Relevant topics are the optimal splitting criterion, the stop-splitting rule, and as an alternative to the latter decision tree pruning. To save space, we will limit our presentation to the two first questions and briefly discuss fuzzy decision trees, since they offer an interesting extension of practical interest.

Optimal splitting criterion

As was explained earlier, while growing a decision tree the main computational subproblem consists of identifying optimal tests to split its nodes. The procedure used in practice is a brute force approach. It consists of enumerating a large number of candidate tests (in practice several hundred thousand), and applying each test to the learning subset corresponding to the current node in order to compute a score measure which evaluates how well the test purifies the learning subsets. The test which obtains the highest score is then eventually used to split the node.

In the literature many different kinds of score measures have been proposed. We will describe the one that we use in our own method and in the examples provided in this course. It is based on normalized information quantity derived from information theory [WVRP89, WP91]. Let us use an example, derived from the Hydro-Québec data base described in §7.6.1.

Let us consider our example problem, and let us compute the score obtained by the test $T \triangleq TRBJ < 7308.5MW$, used to partition the complete data base composed of the 12497 states. This test splits the data base into two subsets composed respectively of 3234 stable and 2408 unstable states for which the condition is true, and 704 stable and 6151 unstable states, for which the condition is not true. This is graphically represented in Table 5.1.

Denoting by $n_{i.}$ the number of learning states of class c_i at the current node and $n_{..}$ its total number of learning states, the <u>*prior classification entropy*</u> of the complete learning set is first computed by

$$H_C \triangleq -\sum_{i=1,m} \frac{n_{i.}}{n_{..}} \log_2 \frac{n_{i.}}{n_{..}} = -[\frac{3938}{12497} \log_2 \frac{3938}{12497} + \frac{8559}{12497} \log_2 \frac{8559}{12497}]$$

$$= 0.899bit.$$

H_C evaluates the degree of "impurity" of classes in the learning set.

The next step consists of computing the *mean posterior entropy* in the subsets defined by the test T. Denoting by n_{ij} the number of learning states of class c_i which correspond to the outcome j and by $n_{.j}$ the total number of states corresponding to outcome t_j, the mean posterior entropy is computed as follows (in the test T above there are only two possible outcomes)

$$\begin{aligned} H_{C|T} &\triangleq -\sum_{j=1,p} \frac{n_{.j}}{n_{..}} \sum_{i=1,m} \frac{n_{ij}}{n_{.j}} \log_2 \frac{n_{ij}}{n_{.j}} = \frac{5642}{12497} H_{C|T=true} + \frac{6855}{12497} H_{C|T=false} \\ &= -[\frac{5642}{12497}\left\{\frac{3234}{5642}\log_2\frac{3234}{5642} + \frac{2408}{5642}\log_2\frac{2408}{5642}\right\} + \frac{6855}{12497}\left\{\frac{704}{6855}\log_2\frac{704}{6855} + \frac{6151}{6855}\log_2\frac{6151}{6855}\right\}] \\ &= 0.706bit. \end{aligned}$$

The *information quantity* I_C^T provided by the test is defined as the difference

$$I_C^T \triangleq H_C - H_{C|T} = 0.899 - 0.706 = 0.193bit.$$

Further, the entropy related to the test outcome is defined (and computed) by

$$\begin{aligned} H_T &\triangleq -\sum_{j=1,p} \frac{n_{.j}}{n_{..}} \log_2 \frac{n_{.j}}{n_{..}} = -[\frac{5642}{12497}\log_2\frac{5642}{12497} + \frac{6855}{12497}\log_2\frac{6855}{12497}] \\ &= 0.993bit, \end{aligned}$$

and the *score* is defined by

$$SCORE(T) \triangleq \frac{2 * I_C^T}{H_C + H_T} = \frac{2 * 0.193}{0.993 + 0.899} = 0.204\ .$$

The score is a normalized measure of the degree of purification. If there is no purification it is equal to zero. On the other hand, it is equal to one if and only if the purification is total, i.e. if the test is able to sort out the stable from the unstable states perfectly. Thus, we may interpret the above result by saying that in the complete data base the test $T \triangleq TRBJ < 7308.5MW$ provides 20.4% purification.

Note that during a tree building typically several million score evaluations are carried out in a very efficient way. Note also that the score measure, being derived from a random sample, is a random variable. It is however possible, as a byproduct of the score evaluation to estimate the standard deviation of the score [Kvå87]. In the above example, the standard deviation is equal to 0.006, which is very small due to the large sample size.

Stop-splitting rule

The stop splitting rule aims at detecting two types of terminal nodes : (i) leaves, correspond to pure enough learning subsets (i.e. $H_C \leq \epsilon$); (ii) deadends, correspond to nodes were the optimal test found leads to a score which is not significantly larger than zero (in the statistical sense).

In our method, deadend detection amounts to apply a hypothesis test on the information quantity. More precisely, under the hypothesis of zero score, the quantity

$$G^2 \triangleq 2n_{..} * \ln 2 * I_C^T, \tag{5.1}$$

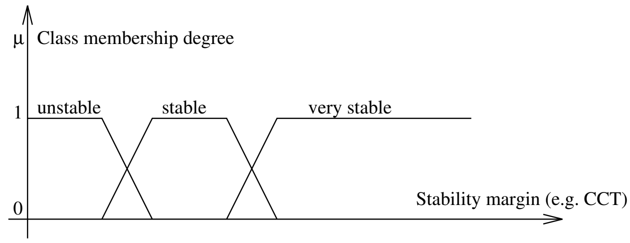


Figure 5.3 *Fuzzy transient stability classes*

is distributed according to a $\chi-$square distribution with $(m-1)*(p-1)$ degrees of freedom, where m denotes the number of classes, and p the number of successors (i.e. of test outcomes). In our case, $m = p = 2$ and the $\chi-$square distribution has a single degree of freedom.

Thus, the deadend detection rule amounts to fixing a priori a value of the non-detection risk α of the hypothesis test, and to comparing the value of G^2 obtained for the optimal test, with the threshold value obtained from the $\chi-$square table. If G^2 is smaller or equal to the tabulated value the node will become a deadend.

Using a value of $\alpha = 0.0001$ (a good choice in practice) yields a threshold equal to 15.2. In the above example, the value of G^2 is equal to 3343.6. Thus the hypothesis is rejected and the node would be developed.

The stop splitting rule is the most efficient way to avoid overfitting in decision trees. It is easy to understand, and allows in general to reduce tree sizes by a factor of two to tree, and computational burden by a factor of two. All in all, it increases the interpretability and the accuracy of the trees.

During the computer demonstration we will further illustrate these aspects on a real life data base.

Fuzzy decision trees

A very active research field concerns fuzzy decision trees, i.e. decision trees using fuzzy logic instead of classical "crisp" logic. Fuzzy logic allows one to reason about partial memberships of object to sets. In the context of security assessment, fuzzy logic may be useful in order to exploit security margins to define fuzzy security classes, as illustrated in Fig. 5.3.

In order to exploit this type of information, fuzzy decision trees use fuzzy (smooth) splits at test nodes instead of crisp ones. They are therefore able to provide smooth input/output mappings, in the form of membership degrees. Figure 5.4 depicts salient differences between crisp and fuzzy trees.

A systematic approach to fuzzy tree induction is proposed in [BW95, BW96]. It is restricted to two-class problems and binary trees. Thus, it consists of using as learning target a security degree derived from a security margin via **two** thresholds (rather than a single one used in crisp two-class trees), and of choosing at the tree growing step at each test node an optimal attribute together with **two** thresholds, defining a transition region between left and right successors (see Fig. 5.4). In order to adapt the tree complexity automatically to the problem complexity and information at hand in the learning set, it is pruned by cross-validation.

While this technique has not yet reached maturity comparable to the hybrid DT-ANN approach, it appears to be more effective in combining the data interpretation capability of symbolic machine learning with

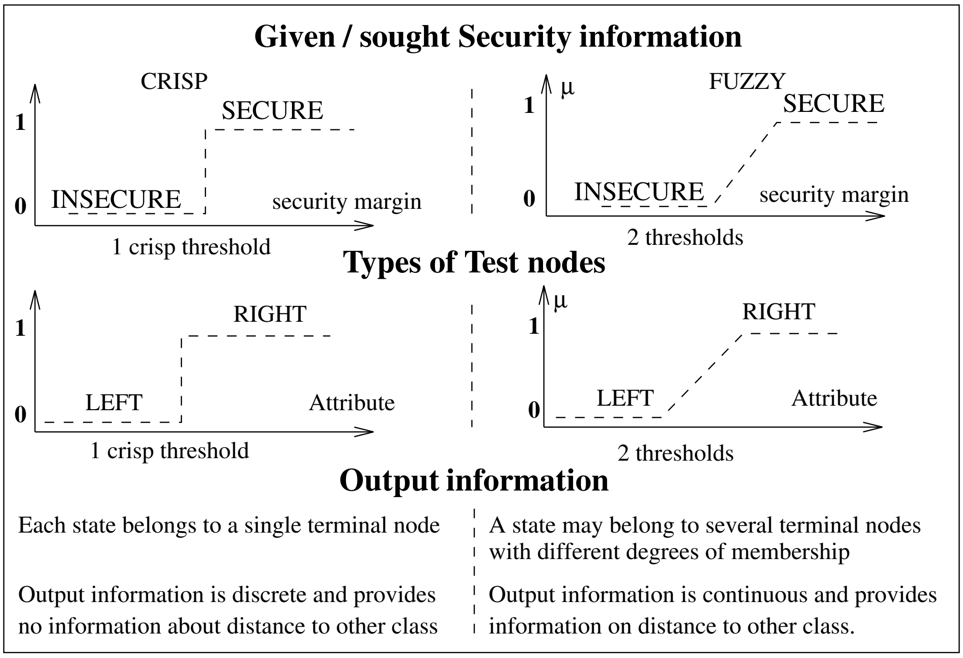


Figure 5.4 *Main differences between crisp and fuzzy decision trees*

a smooth modeling capacity. In particular, its application to a real transient stability test problem has yielded a significant increase in accuracy [BW95]. Also the non-iterative tree growing algorithm is inherently more efficient from the computational point of view than the iterative optimization algorithms used for MLP tuning.

Nevertheless, the main drawbacks with respect to crisp trees is the much higher computational burden of the fuzzy tree learning algorithms. Further developments are underway to make the algorithm faster.

5.4 Non linear regression

The multilayer perceptrons were described in the previous chapter as an example method for non linear regression. We have seen that the learning algorithms used may be sophisticated non-linear regression techniques. However, rather than elaborating on these algorithmic details, we will give a brief introduction to recent theoretical work which is highly relevant in this context. Then, we will come back to algorithms and give a brief description of the projection pursuit method, which is representative of many other modern statistical regression techniques.

Notes.

While decision trees and other machine learning techniques may use both numerical and symbolic input attributes, the non linear regression techniques assume that inputs and outputs are numerical vectors. Below, we assume that the input space is the $n-$dimensional real Euclidean space R^n, and that outputs are in the $d-$dimensional real Euclidean space R^d.

To apply the regression techniques to symbolic attributes, it is thus necessary to code them as numbers. There are many different ways to do this and the quality of results may strongly depend on the type of coding. Unfortunately, there is no space here to elaborate on this topic. Anyhow, generally the best way *of coding is problem specific and must be defined in a pragmatic way.*

*Moreover, to exploit numerical atttributes it is generally preferable to normalize attributes values prior to applying the learning algorithm, in order to avoid numerical problems. We refer the reader to [**?**] for further considerations on these topics.*

Multilayer perceptrons

In our intuitive introduction to multilayer perceptrons we indicated that that single hidden layer perceptrons are sufficiently powerful to represent continuous input/output relationships. In fact, it has been shown that provided that a large enough number of hidden neurons is used, they can arbitrarily well approximate any continuous function on a compact subset of R^n [Cyb89, HSW89]. More recently Barron has studied the bias/variance tradeoff in such kind of networks [Bar93], which is a much more practically relevant topic. Below we will merely explain the type of results he has obtained, and how they may be exploited practically.

Barron shows that in single hidden layer sigmoidal perceptrons bias is upper bounded as follows

$$\text{Bias}^2 \leq \frac{C^2(\text{problem})}{M} \tag{5.2}$$

where $C(\text{problem})$ is a measure of problem complexity he defines, and M the number of hidden neurons. The interpretation is that for any continuous function of finite complexity, there exists a single hidden layer approximation with M neurons whose error is smaller than $\frac{C^2(\text{problem})}{M}$. Thus by increasing the number of neurons M it is possible to reduce approximation errors in a linear fashion.

However, this very good approximation property does not prevent the overfitting problem. Indeed, the practical problem is to choose the right weight values on the basis of the learning set, and here variance comes into play. Thus, Barron further shows that variance is upper bounded in the following fashion

$$\text{Variance} \leq \frac{Mn \log N}{N} \tag{5.3}$$

where N denotes the learning set size and n the number of inputs. In particular, variance is proportional to the product of the number of neurons and the input space dimensionality and decreases less than linearly with the sample size.

Summing the above upper bounds yields an overall approximation bound for single hidden layer perceptrons learned from data

$$\text{Mean square error} = \text{Bias}^2 + \text{Variance} \leq \frac{C^2(\text{problem})}{M} + \frac{Mn \log N}{N}. \tag{5.4}$$

Thus, one can define the optimal number of neurons minimizing this upper bound, yielding

$$M^* = C(\text{problem})\sqrt{\frac{N}{n \log N}}. \tag{5.5}$$

The latter formula is useful from a qualitative viewpoint, because it shows that the optimal number of neurons increases rather slowly with the number of learning states. Thus, the linear decrease in approximation error when the number of neurons increases, can only be achieved provided that the learning sample size increases more than quadratically.

For practical use, the formula needs the evaluation of $C(\text{problem})$ which is either not possible or computationally not feasible. Thus, it needs to be approximated.

Nevertheless, the ideas presented above have been practically exploited in the following way : with a few repetitive trials on small and medium sized samples it is possible to determine the mean optimal number of neurons for a few values of N. Then eqn. (5.5) can be used to approximate $C(\text{problem})$, and together with eqn. (5.4) to compute the sample size that would be required to reach a given level of accuracy.

Projection pursuit

The term "projection pursuit" refers to a class of sophisticated statistical methods which may be used for regression, classification and distribution fitting as well as for exploratory data analysis [FS81, FSS84, Fri87]. We will focus on projection pursuit regression, which outperformed multilayer perceptrons on some power system security information data bases [MST94].

This projection pursuit regression technique models a vector regression function $\boldsymbol{r}(\cdot)$ as a linear combination of smooth functions of linear combinations (i.e. projections) of the attribute values. Thus the model assumes the following formulation

$$\boldsymbol{r}(\boldsymbol{a}) \stackrel{\triangle}{=} \overline{\boldsymbol{y}} + \sum_{i=1,M} \boldsymbol{v}_i f_i(\boldsymbol{w}_i^T \boldsymbol{a}), \tag{5.6}$$

where the order M, the d-vectors $\boldsymbol{v}_i$ and $\overline{\boldsymbol{y}}$, the n-vectors $\boldsymbol{w}_i$ and the scalar functions $f_i(\cdot)$ are determined on the basis of the learning set, in an iterative attempt to minimize the total square error

$$\text{TSE}(\boldsymbol{r}) \stackrel{\triangle}{=} \sum_{o \in LS} ||\boldsymbol{y}(o) - \boldsymbol{r}(\boldsymbol{a}(o))||^2, \tag{5.7}$$

where the notation $||\cdot||$ denotes the Euclidean norm in the output space R^d. For classification problems, the standard class-indicator encoding is used, which is defined by

$$y_i(o) = \delta_{c(o),c_i}, \ \ \forall \ \ i = 1, \ldots, m. \tag{5.8}$$

In the basic approach the functions f_i are special scatter-plot smoothers, which are normalized in the following way

$$\sum_{o \in LS} f_i(\boldsymbol{w}_i^T \boldsymbol{a}(o)) = 0 \ \text{ and } \ \sum_{o \in LS} f_i^2(\boldsymbol{w}_i^T \boldsymbol{a}(o)) = 1, \tag{5.9}$$

and the projection vectors $\boldsymbol{w}_i$ are normed

$$\sum_{j=1,n} w_{ij}^2 = 1. \tag{5.10}$$

The striking similarity of this model with a single hidden layer feed-forward neural network is shown in Fig. 5.5. However, the originality of the projection pursuit regression technique is that both model complexity (the order M) and the smooth activation functions $f_i(\cdot)$ are determined on the basis of the learning set data, while in the basic multi-layer perceptron they are chosen a priori by the user, which leads in general to overly complex structures with many redundant parameters.

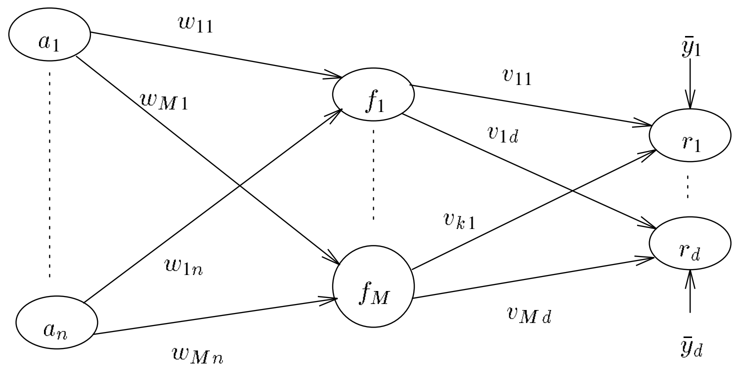


Figure 5.5 *Graphical representation of the projection pursuit model*

Forward growing of projection pursuit

At each step j of the procedure, the order of the model is increased by one unit, by adding an additional projection direction $\boldsymbol{w}_j$ and smooth function f_j and determining the vector $\boldsymbol{v}_j$. During this first step, the parameters of the preceding directions are kept constant.

The second step consists in adjusting in a back-fitting approach all the parameters of all directions $k \leq j$ in a cyclic fashion, so as to minimize the TSE (5.7).

Finally, the model growing procedure stops when the TSE is sufficiently low or when it does not improve sufficiently anymore.

When the model growing is finished, a sequence of models in decreasing order of their complexity is generated, by pruning at each step the least active part of the model, corresponding to the projection direction which influences the least strongly the output values. This is defined as the direction i which minimizes the sum

$$I_i \triangleq \sum_{j=1,d} |v_{ij}|. \tag{5.11}$$

The appropriate model of complexity M is selected by cross-validation.

Back-fitting

The heart of the algorithm consists of back-fitting a group of parameters, $\boldsymbol{w}_i$, f_i, and $\boldsymbol{v}_i$, corresponding to one of the current projection directions $i \leq j$. This is done in an iterative fashion.

1. Adjusting $\boldsymbol{v}_i$ is done directly by setting the derivatives of the TSE to zero with respect to each component of $\boldsymbol{v}_i$. This yields a linear equation, since the TSE is quadratic in $\boldsymbol{v}_i$.

2. To adjust the smooth functions $f_i(\cdot)$, we proceed in two steps. First, non-smooth function output values $f_i(\boldsymbol{w}_i^T\boldsymbol{a}(o))$ are determined for each object $o \in \boldsymbol{LS}$. Again, since the TSE is quadratic in f_i, this can be done in a direct linear computation, setting the partial derivatives of the TSE w.r.t. $f_i(\boldsymbol{w}_i^T\boldsymbol{a}(o))$ ($\forall\ o \in \boldsymbol{LS}$) to zero. Second, the resulting "optimal" output values together with the inputs $\boldsymbol{w}_i^T\boldsymbol{a}(o)$

$$\left(\boldsymbol{w}_i^T\boldsymbol{a}(o), f_i^*(\boldsymbol{w}_i^T\boldsymbol{a}(o))\right),\ \ \forall\ o \in LS, \tag{5.12}$$

   are used as target values to determine the smooth scatter plot function. Note that this problem is a unidimensional approximation problem which may be solved through various techniques. We refer the interested reader to [HYLJ93, SW96] for a further discussion of various alternative schemes to solve this problem.

3. Finally, to adjust the projection direction $\boldsymbol{w}_i$, an iterative gradient descent or Newton method should be used, since the TSE is neither a quadratic nor a linear function of $\boldsymbol{w}_i$.

Discussion

One of the advantages of the *projection pursuit* regression method with respect to standard feed-forward neural network techniques lies in the greater simplicity of the resulting structure. This is due to the automatic determination of the neuron activation function together with the adaptation of the model complexity to the data. While similar neural network growing techniques have been proposed in the literature, the projection pursuit approach has been found to be superior in performance.

Friedman and Stuetzle have proposed various extensions to the basic method to improve its data exploration features [FS81] . For example, by restricting the number of attributes combined in any projection, the method may provide interesting two or three dimensional directions for data exploration. With these extensions this method would provide similar features to the TDIDT approaches discussed in the preceding chapters, with the additional capability of providing a *smooth* non-linear input/output modeling capability, which would be particularly interesting for the estimation of power system security *margins*.

5.5 Similarity based methods

There exists a variety of so called "similarity based methods", conceptually similar to the KNN approach. We mentioned earlier that one of the main problems of these methods is their high sensitivity to the used similarity or distance measures. Thus, in the literature many different algorithms have been proposed in order to adapt these latter to problem specifics, on the basis of the learning set.

Below, we will merely describe the method that we have developed in the context of power system transient stability assessment, which has shown to provide very effective results [HWP95, WHP95a, HWP97].

Genetic algorithm based K nearest neighbors

The method is a hybrid technique, combining KNN with decision trees and genetic algorithms.

Decision trees are used in order to select among the candidate attributes the relevant ones, and to provide an initial guess of the weights that should be used in the distance computation. Then genetic algorithms are used in order to find the appropriate value of K (the number of nearest neighbors) and to further adjust weights in order to minimize the learning set error rate. Only this last step is computationally involved; the first two are byproducts obtained without computational overhead from the decision tree building.

In order to illustrate the effectiveness, we comment the results obtained in [HWP97] on a transient stability data base corresponding to the EDF system. In this data base, the pure KNN using all candidate attributes obtains a test set error rate of 6.6%; using only the DT test attributes, the error rate decreases to 2.9%. Further, weighting them according to the information quantity they provide to the decision tree (this is a global measure of an attribute's discrimination power, derived as a byproduct during decision tree building), the error rate decreases further to 2.1%. Finally, adjusting the latter weight values by the genetic algorithm, further decreases the error rate to 1.3%.

5.6 Comments on parallel computations

To conclude this chapter, we mention that all the automatic learning methods presented in this tutorial may exploit (sometimes trivially) parallel computations.

In particular, while we will see that the data base generation nicely fits to distributed computer systems, the automatic learning methods would take the best advantage of shared memory multiprocessor architectures.

Part II

Application to dynamic security assessment

6

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.

6.1 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 [DL68]. Figure 6.1 shows a more detailed description of the "Dy Liacco state diagram".

<u>*Preventive*</u> 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.

<u>*Emergency*</u> 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

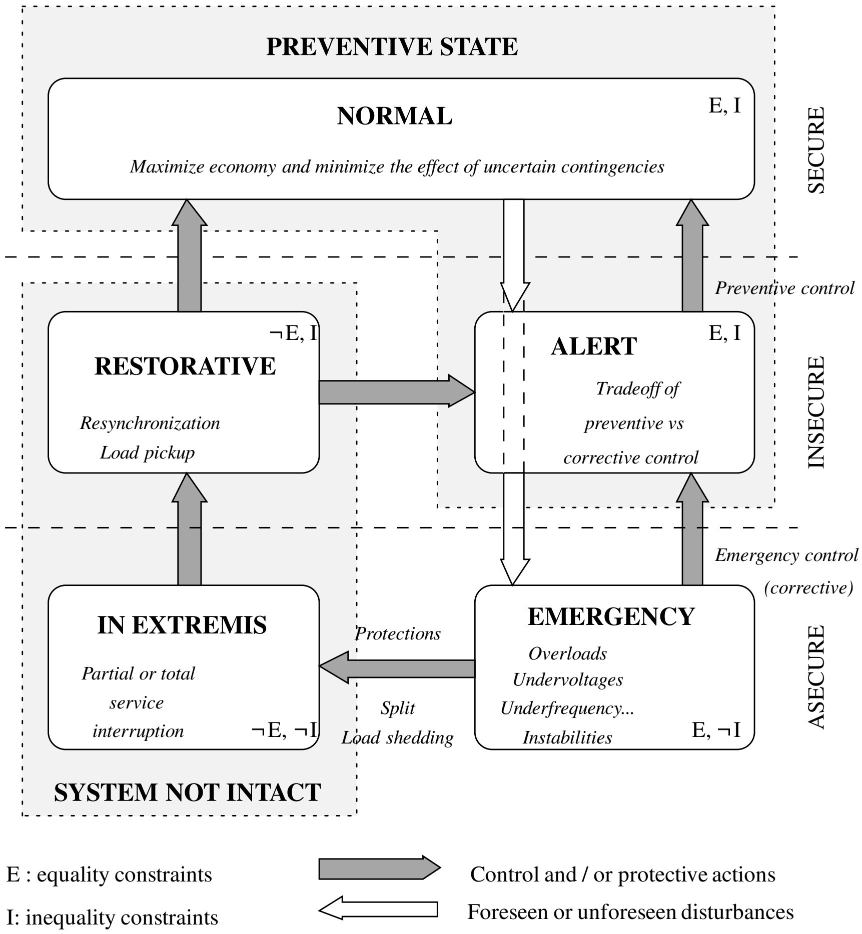


Figure 6.1 *Operating states and transitions. Adapted from [FC78]*

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.

6.2 Physical classification of DSA problems

Figure 6.2 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.

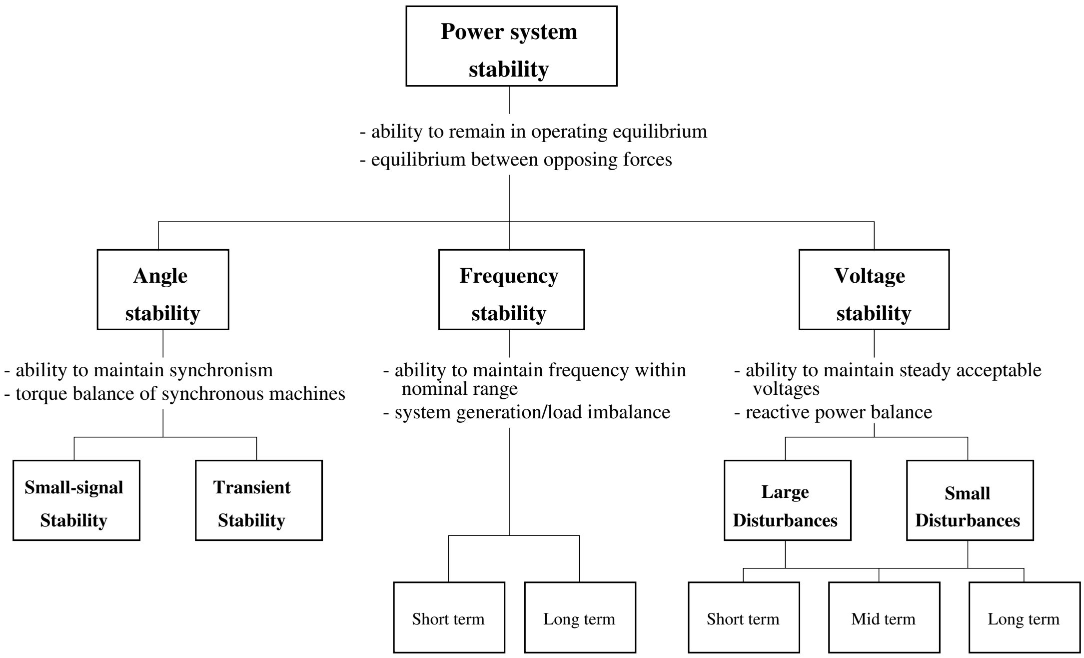


Figure 6.2 *Types of power system stability phenomena. Taken from [KM97]*

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.

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.

Table 6.1 *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

6.3 Practical application environments and possible uses of automatic learning

Table 6.1 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].

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. 6.2 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 7, 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 6.1, 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.

6.4 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 [CM97]. 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.

6.5 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.

7

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.

7.1 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 4, only the operating point was variable. In the case study described in Chapter 9, all three components are 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.

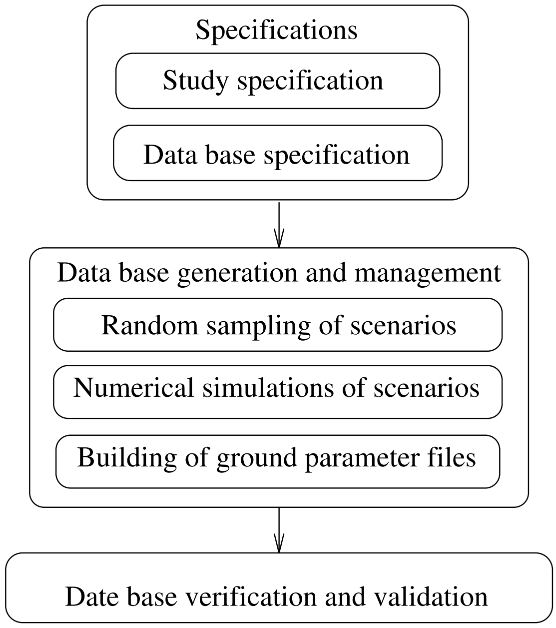


Figure 7.1 *Overall data base generation process*

7.2 Overall process of data base generation

Figure 7.1 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, and may be carried out by appropriate software tools. However, presently there exists no single software package able to encounter all the needs of the various types of security studies. Thus, in the context of our research we have developed a series of specific tools to encounter the needs of specific types of applications. One of them was used in the study described in Chapter 9. Below we will describe the outlook of a general tool which is under development.

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.

7.3 Specifications

7.3.1 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.

7.3.2 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 rather be dynamic system measurements available in real time (voltages, rotor velocities . . .) and output information would measure severity, e.g. incumbent 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 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.

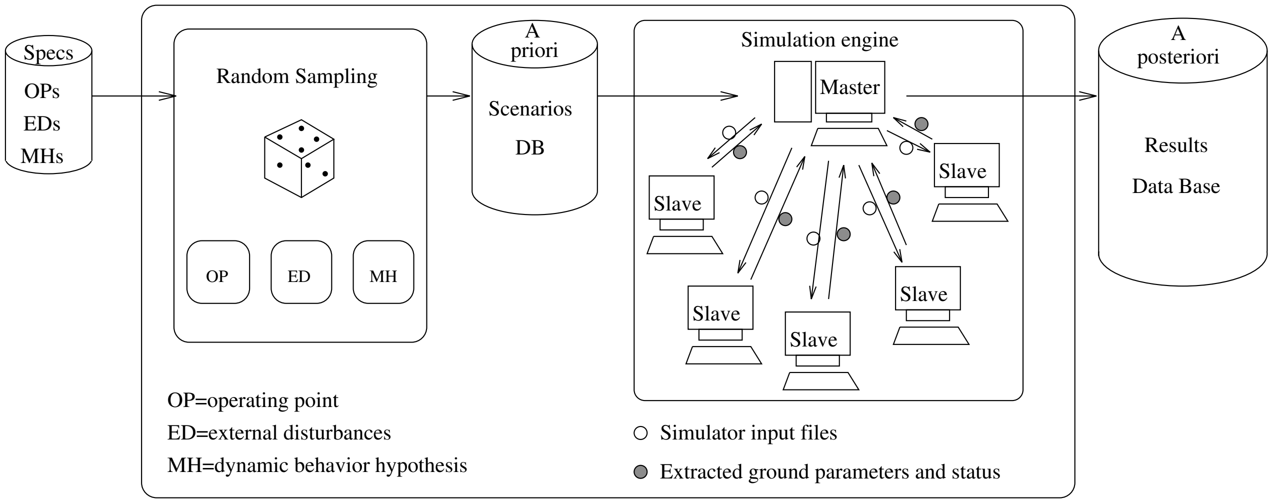


Figure 7.2 *Data base generation tool. Adapted from [WLTB97a].*

7.4 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

The study described in Chapter 9 uses a sequential data base generation tool which was developed a few years ago in the context of voltage security assessment of the EDF system, and which is now used in real life studies. Here, let us consider the more advanced parallel scheme depicted on Fig. 7.2, very close to the tool used in the study reported in §7.6.2. It is composed of the following modules

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. 7.3).
   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. 7.3) : (i) OP building; (ii) dynamic simulation; (iii) extraction of ground parameters.

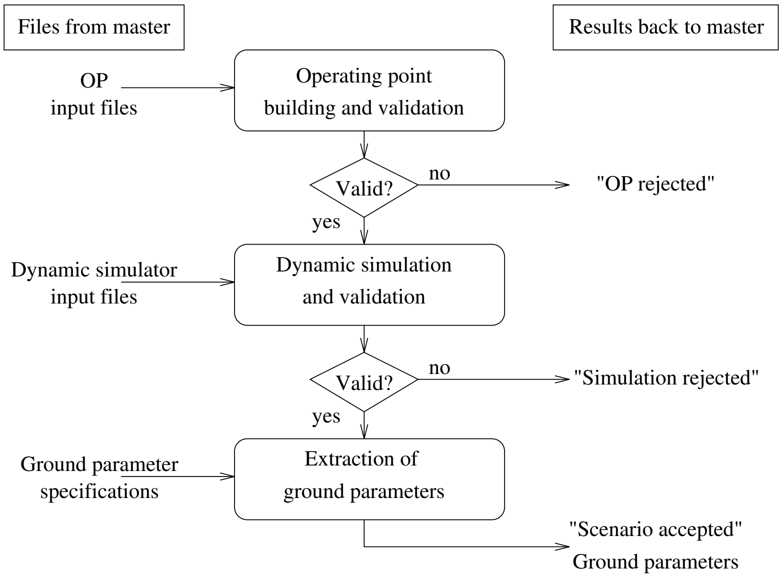


Figure 7.3 *Scenario simulation.*

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.

7.5 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.

7.6 Examples

7.6.1 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.

The power flows in the three important corridors of the Hydro-Québec system are drawn independently in the intervals indicated in Fig. 7.4. 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. 7.4, are widely variable during the random sampling since

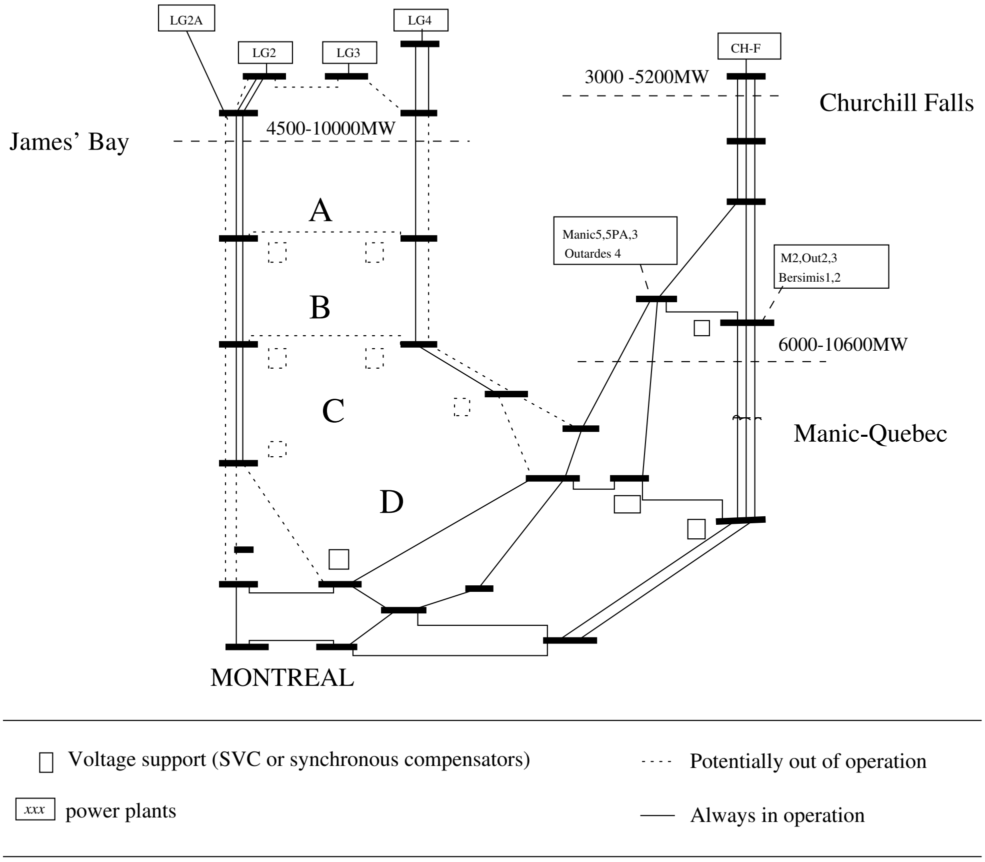


Figure 7.4 *Main transmission corridors of the Hydro-Québec system*

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.

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. 7.5 shows a typical frequency diagram, similar to those obtained in the first a priori

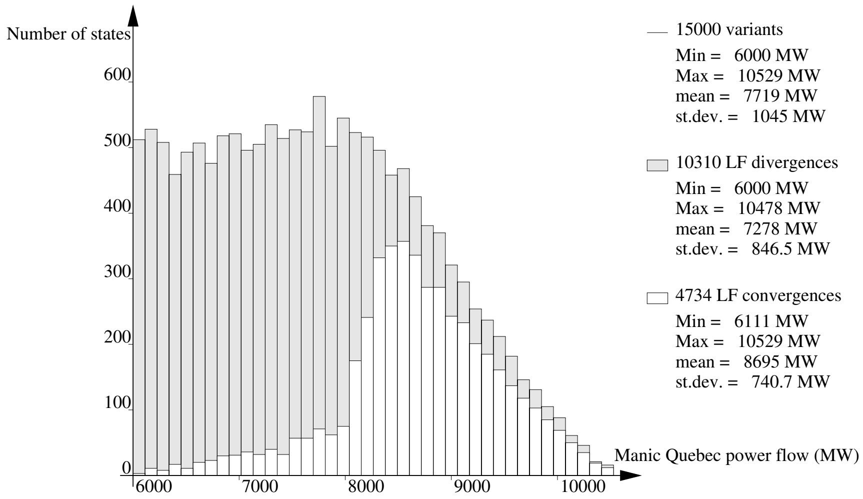


Figure 7.5 *Convergence diagram of Manic-Québec power flow (6 base case files)*

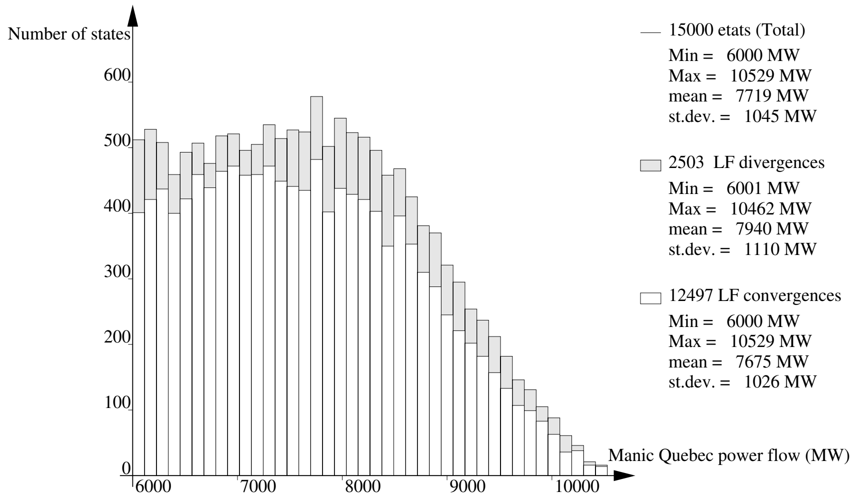


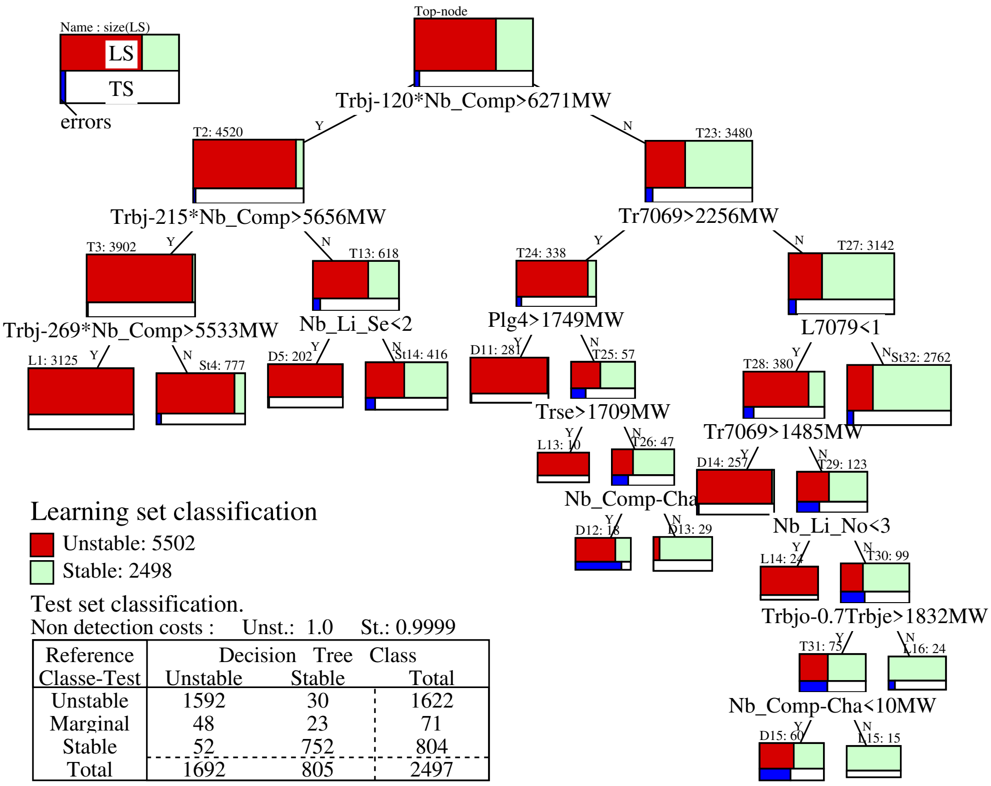
Figure 7.6 *Convergence diagram of Manic-Québec power flow (12 base case files)*

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 7.6 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. 7.5, 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.

The operating points were then sent back to Montréal and fed into the LIMSEL data base in order to



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

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. 7.7 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. 7.7 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

Table 7.1 *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. 7.7					
P_e% (TS)	6.93	6.73	6.13	6.13	6.61

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.

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 7.1 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. 7.7 : 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.

7.6.2 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.

[1] The attribute values are however normalized.

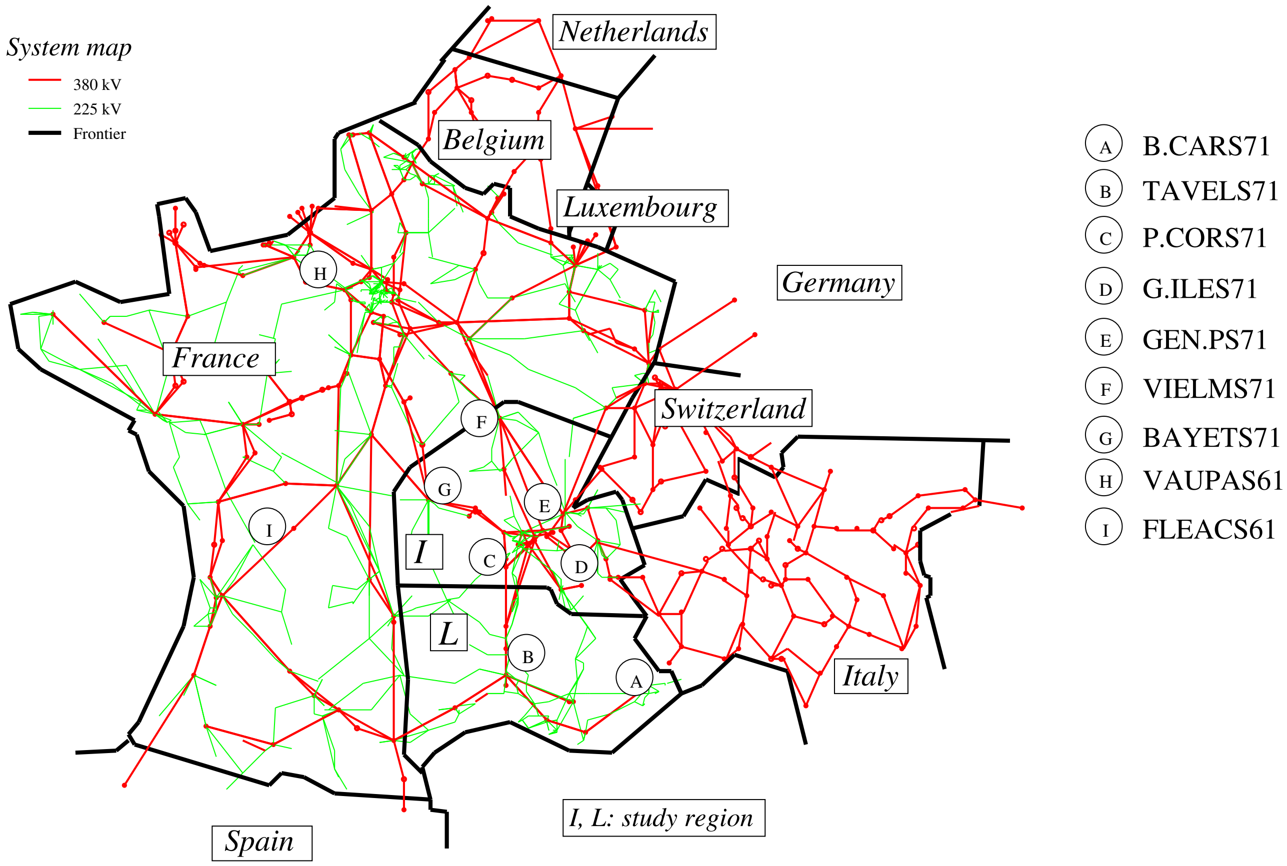


Figure 7.8 *One-line diagram of the study region and surroundings*

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. 7.8) 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 §7.4. 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 7.2 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 7.2 *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. 7.8)				
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

Part III

Applications at Electricité de France

8

History of applications at EDF

EDF has been interested for long in studying the applications of automatic learning methods to power system security assessment. The work in this field has mainly been achieved through a series of collaborations with the University of Liege. It all began in 1990 . . .

8.1 First researches about transient stability studies

During 3 years, the AL framework has been applied to transient stability problems. The question was to know in what conditions the generators in a new nuclear power plant might lose synchronism [WPEH94].

This was actually the very first application of the automatic learning framework to a real large scale power system DSA problem. It contributed to develop the data base generation methodology and allowed to gain better insight into the capabilities of automatic learning in large scale problems. In particular it yielded several improvements in the decision tree method [WPEH94, WA93, AWP+93].

The method has proved to be useful, but the kind of study considered was also successfully carried out in a more traditional manner. The reason for this lies in the problem physics, which implies rather few elements (presence of neighboring lines, power generated by the machine, voltage, . . .). In addition, it did not appear possible at that time to use the state-of-the-art dynamic simulation tool in the AL framework, due to computer resource limitations.

We note that the AL framework could have improved such transient stability studies (e.g. by taking into account modeling uncertainties, or the risk of protective relay malfunctioning), but at the time its interest for this problem was not deemed important enough to apply it in operational studies.

8.2 Second, the voltage security assessment problem

Emergency state detection

EDF system has experienced an actual voltage collapse in January 1987. This subject since then became of great importance in the company. The main objective was to find out an emergency state criterion, able to alert the operator and trigger some corrective actions.

After some first works, the attention came to the automatic learning approach to determine which kinds of criteria would best fit the problem. Some research work in this field began also in 1990.

This work went much farther than the preceding one, because the application methodology had become more mature in the meanwhile. Researches went on until 1992. The diagnosis is that the method allows to determine emergency state detection criteria (e.g. to trigger on-load tap changer blocking schemes) potentially more effective than those presently in use. However, it was also found out that the thresholds used in such criteria would be rather sensitive to various parameters (time of measurement, load models. . .) thus leading to practical difficulties. Finally, it was deemed that the approach was indeed very promising but required further research to become applicable in real-life.

Again, this was the very first attempt to apply automatic learning to emergency control of a large scale power system. Its two main outcomes from the methodological viewpoint were : (i) modeling uncertainties need to be properly included into the data base specification and generation; (ii) in emergency control the attributes should be treated as temporal variables rather than snapshots.

The needed research and software developments were postponed until later (see §8.3).

Preventive security assessment

Thus, given the good results in preventive transient stability assessment, the decision has been taken in 1993 to focus on the problem of determining some preventive security rules with respect to the risk of voltage collapse.

The methodology has proved here again its effectiveness. Moreover, the use of dynamic simulation tools in order to assess voltage security was still recent in EDF operation planning environment, and one can say that the research on the application of automatic learning has actually influenced the approach presently used there.

Due to the lack of industrial grade software tools, the AL framework is however not yet implemented in these very operational environments. Nevertheless, some very operational studies are carried out now in the R&D division using the advanced research prototype software [JWP96]. For instance, one of them aims at determining whether the current security rule (a limit of power flows on parallel lines) could be improved by monitoring some other parameters (such as reactive reserves), and whether this criterion should be different with respect to different families of system situations, to be also defined.

8.3 Global security study

More recently, a very prospective research study has been launched.

The underlying idea is that the AL approach could do more than improving the way to carry out classical security studies. Indeed, it can be a means to approach dynamic security in a global fashion, so as to analyze and improve the dynamic performance of the power system, whatever kind of element and phenomena come into play.

The potential interests are quite important, since (i) it is well known that power systems will have to operate closer to their limits, and (ii) they are more and more equipped with complex automatic devices making their behavior change radically. While dynamic phenomena in present day electric systems are rather well circumscribed, it seems reasonable that these changes might lead to the occurrence of new, more intricate problems on the networks. One can then expect that future incidents on the systems would come from such automatic devices malfunctioning or unforeseen coupling effects, and mix several dynamic phenomena (line currents, voltages, angles)[1].

The idea of the proposed global approach is to model all kinds of possible *causes* which may lead to a breakdown scenario, then screen a representative sample of combinations of these causes and simulate the behavior of the system in order to find out what will actually happen in terms of *consequences*. The aim is to determine which among the possible consequences are the most probable ones, what are the consequences in terms of severity, how far do the phenomena spread in the system, which protective devices act most frequently, which malfunctionings are the most dangerous ones, what are the interactions among different special stability control systems. . . and what would be the most useful improvements.

Our study, described with more details in §7.6.2, considers a very detailed dynamic model of a part of the French network, and intends to analyze any phenomena which might occur on this system when subject to severe disturbances, independently of any a priori physical decomposition. The system model also takes into account possible malfunctionings of protective relays and other protection devices.

It implies the generation of a huge data base storing some temporal informations (parameters variation curves), instead of snapshots as in the previous works.

It is too early now to draw any conclusion about the analysis results, but the potential of such an approach seems very important.

8.4 How research was carried out

During these seven years of research collaboration many efforts were made in order to transfer knowledge, first from the utility to the university about actual security problems and existing practices to tackle them, then from the university to the utility about the developed framework and its potentials.

From the very beginning several EDF experts were involved in all the discussions in order to provide insight on how to approach problems. The software developments and actual research work were carried out at the university.

During the more recent research projects three EDF engineers stayed successively during 16 months at the university and participated actively in the research and development activities. This was found to be a very effective means to share know-how among utility and university and thus increase the probability that the research results would eventually be used in real-life.

[1]This seems to be already the case for recent incidents such as the one which affected the Western North American system on August, 10th 1996.

9

A real voltage security assessment example

We describe here in some details what a voltage security study carried out with this approach looks like.

It begins with the study definition; then the data base is specified, and generated; a data base validation phase is also required; finally the validated data base is exploited by applying various automatic learning techniques.

9.1 The presented study

The problem considered is to identify the risk of mid-term voltage instabilities due to an important disturbance. In this context, the main physical phenomena which influence the system behavior in response to the contingency are (i) the load restoration thanks to either the action of On-Load-Tap-Changers (OLTC) of transformers, which act so as to restore high voltage (HV) and medium voltage (MV) voltages to their nominal values, or actual load dynamic characteristics, and (ii) the change of reactive productions of the machines, possibly through a secondary voltage control, and their limits (over-excitation limiters action).

The presented study aims at demonstrating the approach feasibility. Its scope is thus very broad : important variations are considered for the network situations parameters, and both preventive and emergency aspects are treated. It is representative of an off-line security study, carried out at least one year ahead, say for the next winter. The objective could be to determine the conditions when a non-economical generation unit must be started for security purposes, or the tuning of a protection device criterion (tap changer blocking remote control. . .).

The power system considered is the western part of the EDF system, which has experienced a real voltage collapse in the past [HTL+90].

9.2 Data base specification

While concentrating on preventive security assessment the data base specifications were set up in order to allow also various possible investigations in the context of emergency control.

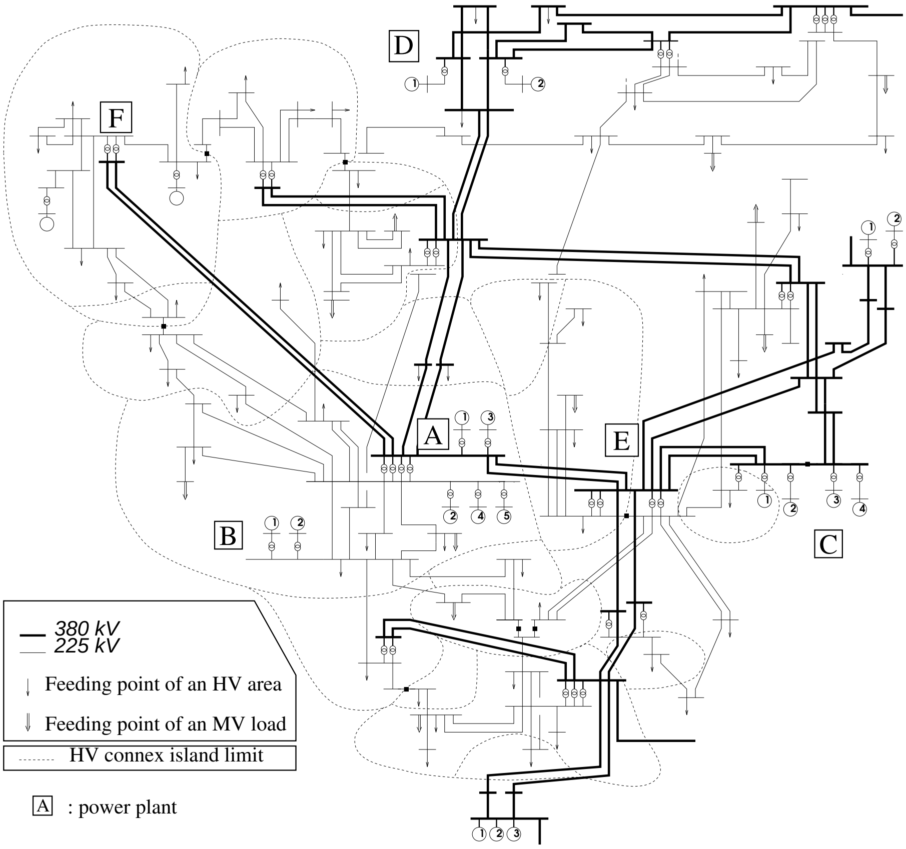


Figure 9.1 *The Western part of the EDF system (study region)*

9.2.1 Network model

The power system model is a quite detailed one (see Fig. 9.1). It includes more than 1200 buses, comprising (i) the full French 400 kV network, (ii) the western part of the 225 kV level, (iii) the detail of the HV levels (90 and 63 kV) delimited by the dotted lines on the figure, and (iv) the loads being represented behind the HV/MV transformers. All EHV/HV and HV/MV transformers, which amount to 450, are equipped with OLTCs. The 320 loads connected to the MV level are sensitive to the voltage[1]. The machines of the study region all have their over-excitation limiter considered. Moreover the French Coordinated Secondary Voltage Control (CSVC) [VPLH96] algorithm is applied to the two regions covering the detailed part of the network.

9.2.2 Range of situations

The range of system situations is determined by the variations of two kinds of parameters.

The *primary parameters* are those suspected to have a direct, strong effect on the power system security. Let us quote for instance topological conditions, unit commitment, load level, power flows in important links or EHV voltages.

On the contrary, the *secondary parameters* are less well known and/or less controllable variables. The objective is not to interpret their direct influence on the security, but to ensure the results robustness

[1] $P = P^0 \left(\frac{V}{V^0}\right)^\alpha, Q = Q^0 \left(\frac{V}{V^0}\right)^\beta$

by guaranteeing their independence from a particular choice of values. These secondary parameters variations are thus somewhat like noises introduced in the data base, liable to be reflected by the attributes. Then the automatic learning algorithm may discard the most sensitive attributes, or at least bias thresholds in order to cope with this effect. Most often, these secondary parameters concern unobservable parts of the network (low voltage levels, or external network).

Primary parameters

The topology has been treated in the same manner as other parameters. Three kinds of situations were considered, each with a probability attached : full network, one element less and two elements less. The disconnected elements were then chosen at random in predefined lists, according to the kind of situation. The main branches of the network shown in figure 9.1 were part of these lists.

The load level has been widely varied. The aim clearly was to hit the maximum loadability limit, in order to ensure a sufficient representation of weak situations. The variation applied homothetically to all 320 MV load buses. The base case load (a very highly loaded situation) was about 7000 MW and 2100 Mvar over this region; a uniform variation between 5000 and 9000 MW was specified.

The regional unit commitment was considered with a slight dependency on the load level : the regional importation had to stay within limits. Apart from this, the decision was taken power plant after power plant (A to D), according to fixed probabilities to have a certain number of units started. The production dispatch was also fixed by unit type, except for the non-nuclear power plant A where a continuous distribution law has been considered, representing both cases with intermediate and full production. No general algorithm, such as an Optimal Power Flow for instance has been used, in order to ensure results validity even on unusual generation patterns situations, which are neither foreseeable nor avoidable when studying security problems one year or so ahead.

The voltage profile is defined via the Coordinated Secondary Voltage Control set-points. They have been varied with a Gaussian law.

Secondary parameters

In addition to these important variations, several secondary parameters were considered, and corresponding noises defined.

The load distribution over the HV networks. Considering homothetical load variations might have led to over-interpretation of some particular power flows in branches feeding load regions. To avoid this independent, per-load Gaussian noises have been used in addition with the global variation.

The load sensitivities to voltage (α and β) have also been randomized. Indeed these parameters are of paramount importance when seeking an emergency control criterion. However they are very badly known; the expected issue of the noise is thus to guarantee the independence of the results from a particular choice for it. Standard uniform laws have been used for the independent, random choice of these parameters.

The production dispatch among generating units in power plant A can also be seen as a secondary parameter, since this precise information was not intended to be directly used in a security rule.

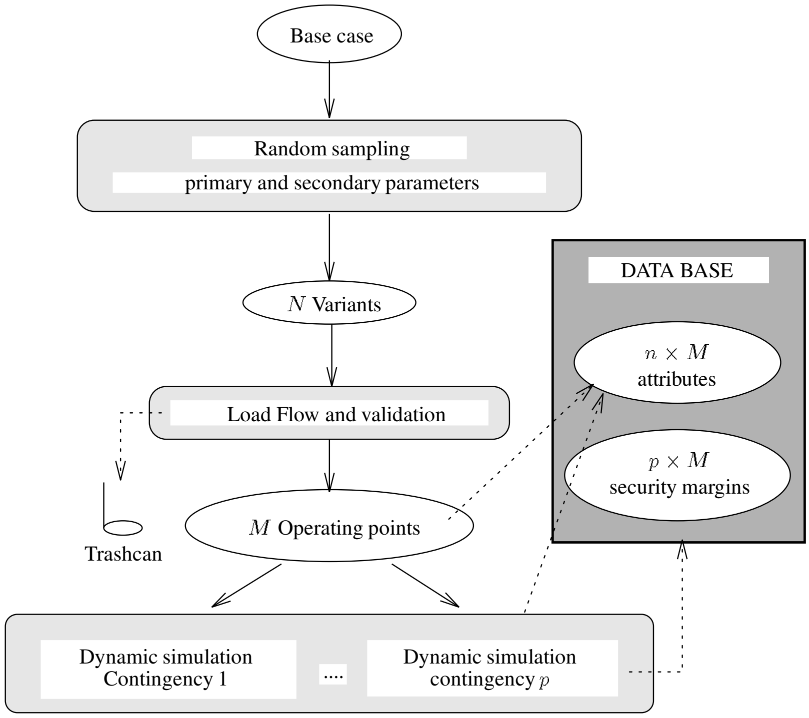


Figure 9.2 *Data base generation process. Adapted from [JWVP95]*

9.3 Data base generation

9.3.1 The generation process

First step : Operating point building

In addition to the random generation itself, the operating point building involves several operations, adapted to the modeling (Fig. 9.2).

1. From a base case situation, a *variant* is randomly generated according to the data base specifications.
2. Then a load-flow program is run on the base case data modified by the variant definition.
3. If this calculation converges, the voltage profile is adjusted with respect to EDF practices, using both the HV shunt compensation and the CSVC.
4. This final situation is validated : a voltage stability index is computed (the sensitivities of the total reactive power generation to a reactive power consumption, known as "reactive power dispatch coefficients").

This process results in a validated *operating point*. Should any part of this process fail, then the variant would be definitely discarded.

In our study, 5,000 such operating points have been generated.

Contingencies effect and security assessment

The study aimed at determining the main problems in the region with respect to voltage security. Thus a list of the main equipment outages has been considered. 26 contingencies have been identified : losses of one or two lines, one or two generators or synchronous condensers, and busbar faults. This choice was the expert's : it has nothing to do with probabilities of any kind.

Each of these 26 contingencies has been applied to each of the 5,000 operating points, and the consecutive system behavior analyzed with the dynamic simulation tool. The post-contingency system robustness was characterized by two complementary measures :

1. the stability verdict, assessed by the use of the "reactive power dispatch coefficients";
2. the load power margin, which provides a continuous security measure expressed in terms of the additional load (P and Q) which may be supplied by the system under acceptable conditions. It is obtained by simulating a steady load increase from the post-contingency equilibrium point, while scanning the system stability; the simulation stops when a sensitivity becomes negative, and the effective load increase until then is measured.

To estimate the dynamic behavior and assess the situations robustness, an especially voltage analysis dedicated simulation tool is used. The underlying idea is to simulate the interesting, mid-term dynamics (OLTCs, secondary controls, over-excitation limiters. . .) while filtering out the faster short-term transients (primary controls. . .). To this purpose, the dynamic, differential equations corresponding to the faster phenomena are simplified to algebraic ones by assuming these dynamics have reached their equilibrium point before the mid-term phenomena come into play [VJMP96]. With the limitation of being unable to highlight problems caused by the fast dynamics, this kind of approach allows drastic reduction in computing times.

Nonetheless, it must be clear that the use of a simplified version of simulation tool is not a requirement of the AL approach.

A single software

The whole data base generation process has been integrated in a single software, thus rendering this operation fully automatic. One "merely" has to specify the number of operating points to generate, the random distributions for the network parameters, and the list of attributes to be collected including the stability verdicts and margins.

The result is a set of files containing the attributes and/or the verdict or margin for the range of situations. It is directly interpretable by the data mining software.

Computation time

All in all, 135,000 ($27 \times 5{,}000$) dynamic simulations have been performed, to determine the pre-contingency and the 26 post-contingency load power margins. The CPU time requirement was about 40 h per contingency, for all operating states on a Sun Sparc 10 workstation. This time is obviously indicative, since the simulation length directly depends on the states robustness : the greater the margin, the longer the simulation. Reducing this time to a per state scale yields an average of 30 s, which highlights the efficiency of the simulation tool used.

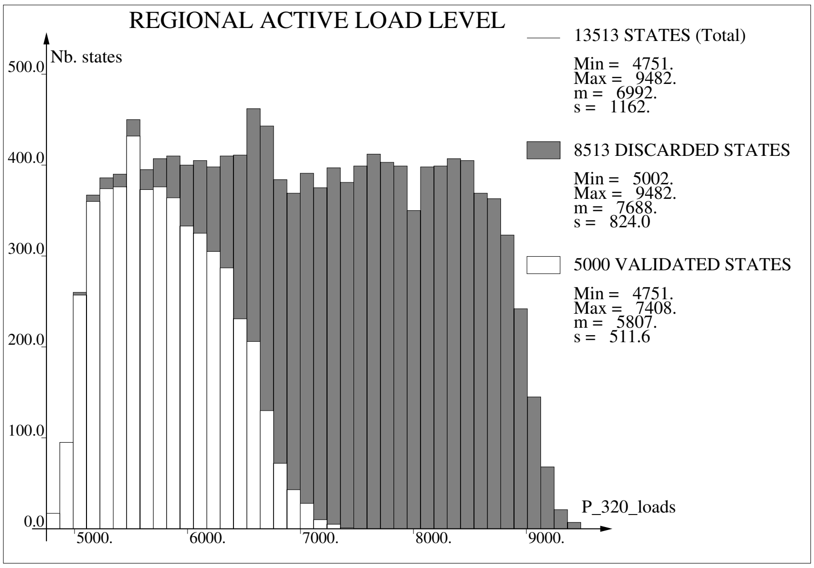


Figure 9.3 *Variants filtering with respect to the load level. Adapted from [JWVP95]*

Of paramount importance are the trivial parallelization possibilities of this process : first by running several contingencies studies on different workstations in the same time, but also by allocating the dynamic simulations among as many as available workstations. The direct result is obviously the possibility to reduce the CPU time indicated above, but also to really claim the data base generation feasibility, independently of particular standard system-theory models. Should anyone want to use a more refined simulation tool, thus increasing the per-simulation CPU time requirement, then the only consequence would be an increase in the need for CPU, which become more and more powerful and affordable. Nearly nothing changes from the user point of view, since the process is fully automatic, and thanks to the parallel computing operating systems improvements.

A last, but not least, remark about these computation times : they have shown to represent only a small proportion of the global study length. Indeed, it takes much more time on the one hand to think precisely of what has to be done in the study, and the way to achieve it, and on the other hand to analyze the huge amount of information contained in such a data base and get out of it the sought synthetic information. To our opinion, the engineers responsible for the study get actually much more concerned by the fundamentals of the study i.e. the physical problem, the software and data management part of it being fully automated.

9.3.2 Data base validation

5,000 operating points have been validated, but 13,513 variants had to be generated for this, 8,513 having led to load-flow divergence or direct voltage stability problems. The main reason of the low percentage of load-flow convergences lies on the excessive load level specified, as illustrated on figure 9.3.

Due to correlations introduced in the random generation, other variables can suffer from side-effects. For instance, figure 9.4 shows the production distributions alteration for power plant D. In this case, the problem arises from the constraints put on the regional power importation.

But still a wide range of situations

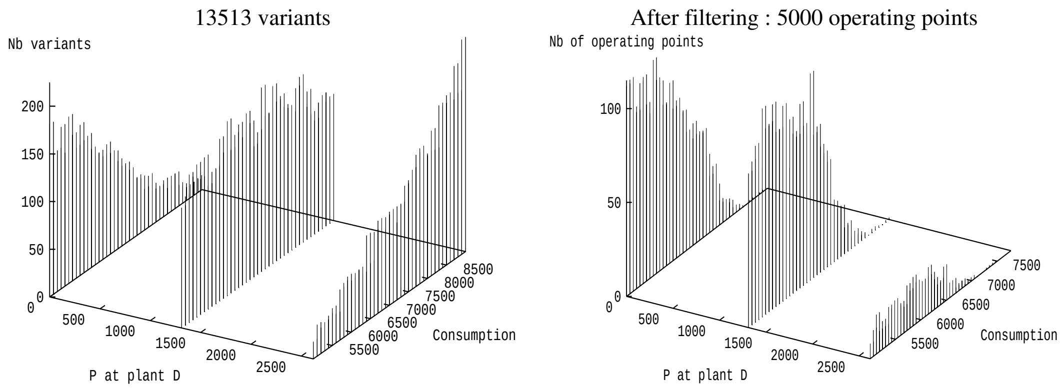


Figure 9.4 *Effect of the load statistical distribution alteration on the production at power plant D. Adapted from [JWVP95]*

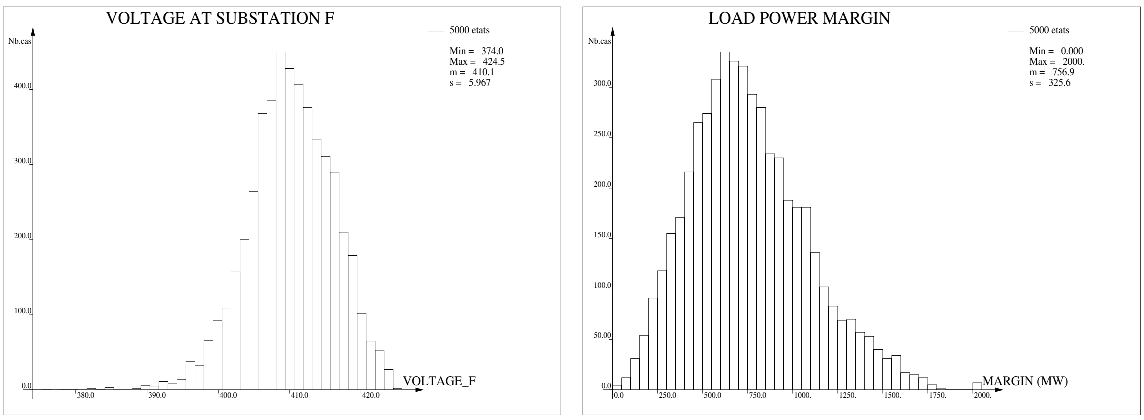


Figure 9.5 *Distributions charts for Voltage at substation F and pre-contingency margin*

A systematic analysis was carried out in order to assess the variability of various attributes, all in all about 400 different ones. It was found that, despite the effects of the variants filtering during the generation, the data base covers a very wide range of situations.

As an illustration, figure 9.5 sketches the distribution of the voltage in substation F, at the end of the network antenna (see Fig.9.1). One can readily see how different the situations included in the data base are : some have very low voltages (lower than 400 kV), while some others present a very high voltage profile due to a large amount of shunt compensation in operation.

This diversity appears also in terms of system robustness. The right part of figure 9.5 shows the distribution of the pre-contingency load power margin. An important feature of the data base is to have both weak cases, with small margins, and robust ones with load power margins greater than 1500 MW. In these conditions only one may hope being able to find out what makes a situation be weak.

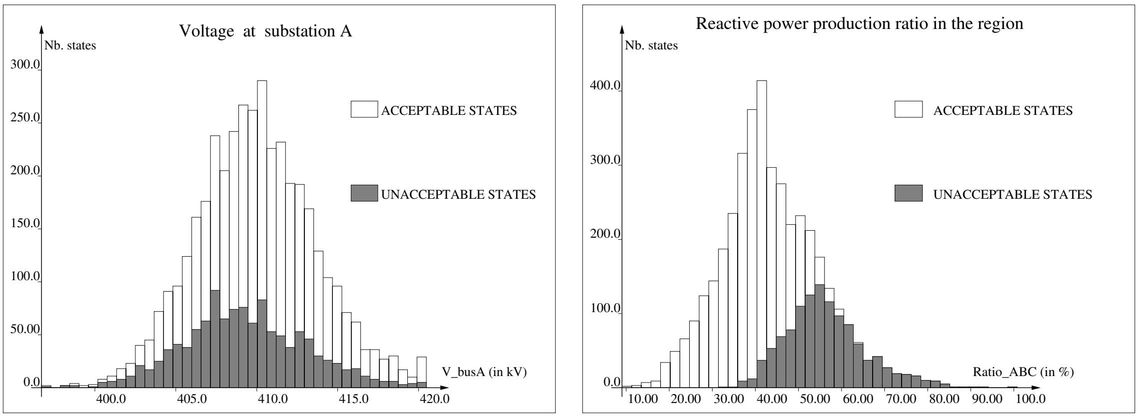


Figure 9.6 *Conditional distributions charts for two pre-contingency parameters. Adapted from [JWVP95]*

9.4 Data base mining

9.4.1 Security with respect to a contingency

Let us focus on a specific problem : determining which operating points lead to a voltage collapse when facing a given contingency : the loss of one generator in power plant A, for instance.

To classify the states, let us declare *unacceptable* any operating point leading after the contingency to a situation with a load power margin lower than say 250 MW (including those directly unstable, i.e. without any margin).

Test defined at a glance

The first analysis merely consists of listing the system parameters liable to have an information on the operating point security : voltages, power flows, load level, productions. . .

A single look at their conditional distribution charts proves to be instructive. For instance, figure 9.6 sketches these charts for the voltage at sub-station A and the reactive production ratio (Q/Q_{max}) over power plants A, B and C. One can easily see how difficult a separation of unacceptable (grey) and acceptable (white) cases thanks to any threshold on the voltage may be. On the contrary, this task appears much easier when considering the ratio of reactive production.

Clearly, no test defined on this latter parameter will be perfect either (figure 9.7) [2]. However one is readily able to know how efficient any of these rules is.

Automatic learning by decision trees

The preceding analysis thus has shown the power of the regional reactive production ratio as an indicator for voltage security with respect to the loss of a generating unit. Decision trees actually generalize the preceding analysis, in an automatic and multi-variable context. They can reveal complementary attributes. For instance, figure 9.8 sketches a decision tree demonstrating the complementary aspect of

[2] Obviously, some safety margins may be considered, especially to reduce the non-detection rate, but this results in a false alarm rate increase.

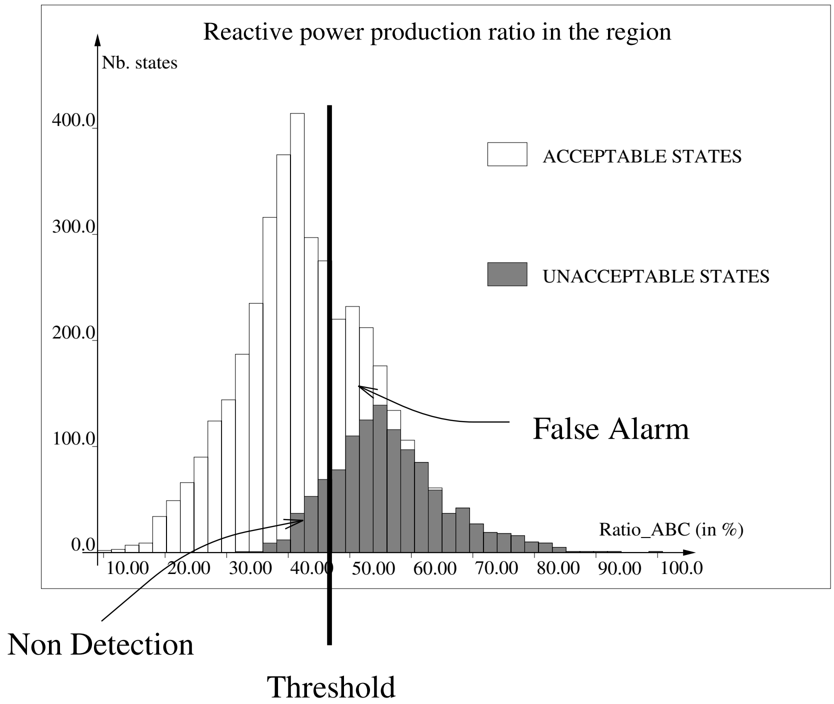


Figure 9.7 *Test classification errors.*

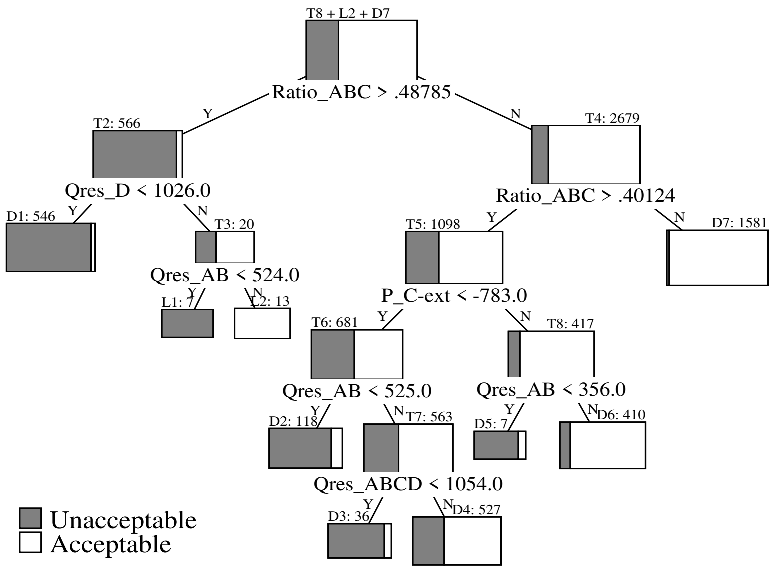


Figure 9.8 *Decision tree for preventive security assessment*

actual reactive reserves, and their localization among the plants.

Starting at the top of the tree, the set of situations (represented by the top box, with the proportion of unacceptable cases illustrated by the grey part of the box) is first split with respect to the reactive production ratio, leading to one set with a great majority of unacceptable cases, and another including a majority of acceptable ones. Then the reactive reserve at plant D is used to identify a family of definitely unacceptable situations and a little set of 20 mitigated cases. The latter is correctly sorted by a test defined on the reactive reserve at plants A and B.

Such a decision tree is obtained in a very simple, and thus fast, fashion :

1. for each candidate attribute, the algorithm scans all possible tests, measures their efficiency, and determines the best threshold;

2. the best test among all candidate attributes is retained;

3. if the test is found effective enough, the set of situations is split according to this test and the algorithm is repeated on each subset. Otherwise, the current node is said to be a terminal node, and a class label is attached to it according to the majority class among its states.

To use this tree as an operating rule, only the variables used in the tests have to be available. Then one traverses the tree, starting at the top-node, and applying sequentially the dichotomous tests encountered to select the appropriate successor. When a terminal node is reached, the label attached to it is retrieved.

This tree has been tested this way with 800 states which had not been used in the learning phase : the result is an error rate of 10.2 %. Will discuss this figure in some more detail below.

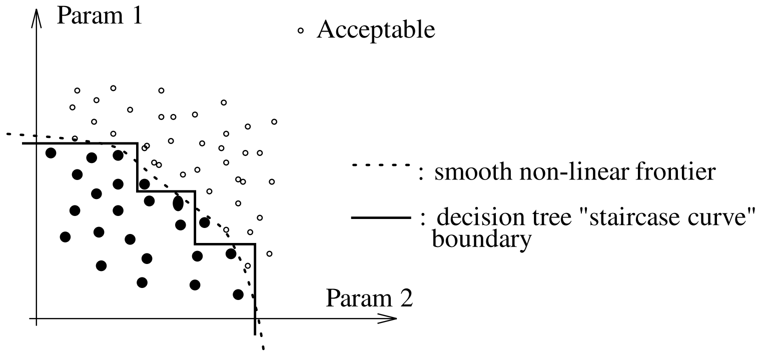


Figure 9.9 *Different security boundary shapes*

Test refinement by other learning techniques

Decision trees induction is thus an easy to use method, allowing to systematically explore large data bases and extract the relevant parameters out of it, and simple rules easy to interpret.

However, the interpretability is obtained thanks to a simplification : the security boundary in the power system state space is approximated by a “staircase curve”(see figure 9.9), leading to an increase of the classification errors located nearby the security boundary.

To reduce the number of such errors, some other learning techniques can be used. Among them, the artificial neural networks (ANN), and particularly the multi-layer perceptrons, appear to be effective [WVP$^+$94]. For our tree, the error rate is reduced from 10.2 % to 6.8 % on the same problem and even 5.3 % by using the continuous margin instead of a mere classification.

Our conclusion is thus that the artificial neural networks are relatively efficient in the task of refining the decision trees criteria, in an hybrid DT-ANN approach as described in [WA93], but not so interesting when used on their own.

Indeed, ANN do not explicitly select the key variables of the problem among a list which may be 200 elements long. This results in a complex, black-box result, which makes it difficult for the system security experts as well as for the operators to appraise and accept this kind of rule. Moreover, the computing time for ANN learning is in the order of magnitudes of hours, versus a few minutes for a decision tree growing. This prevents from a systematic, extensive exploration of the data base, which however constitutes a first, necessary stage in the methodology.

So decision trees induction is deemed to be a good central method for data base analysis and exploration, which can be completed by other learning techniques to gain accuracy, useful in the context of an on-line use. We mentioned the multi-layer perceptrons, but some other hybrid approaches have also been developed to smoothen the security boundaries found by the decision trees, using fuzzy sets for instance [BW95].

An important remark about rules accuracy

The error rates given above for the security rules found may seem rather high at first glance. These figures have however no meaning by themselves.

First because the data base is not generated according to actual probabilities of system elements outages.

Table 9.1 *Load-power-margin computation error vs classification error*

Noise	Classification error rate	
standard deviation	Gaussian distribution	Uniform distribution
10MW	1.23%	1.45%
15MW	1.71%	1.88%
20MW	2.17%	2.28%
30MW	3.33%	3.40%
100MW	10.67%	11.54%

So the 5% of cases which lead to some errors here may represent only 0.1% of the system operating conditions.

Second because an important part of the errors is merely due to numerical inaccuracies in the load power margin computations and other uncertainties. Thus, we considered a threshold of 250 MW on the load power margin as the frontier. But this margin is not 1 MW precise, far from that.

Indeed, the standard deviation of the *numerical* computation error of the margin was estimated to about 15MW. Further, for a fixed regional load level, the standard deviation of the margin variation due to uncertainty in the load *distribution* as modeled in the data base is larger than 60MW. This estimates the load-power-margin accuracy obtained via "exact" numerical computations. Using the latter to classify operating states of the data base, this uncertainty translates into classification *errors* in a way depending on the density of states in the neighborhood of the classification threshold. E.g., Table 9.1 shows the relationship between LPM computation noise and error rates, for a contingency corresponding to the loss of a generator in Plant 1, and classification with respect to a LPM threshold of 250 MW. The error rates are obtained by comparing the classification obtained by the computed margin, with the classification obtained by adding a noise term (assuming either a *Gaussian* or a *uniform* random distribution) to the computed margin. The figures are mean values from 20 passes through the data base with different random seeds.

As a conclusion, we must emphasize the fact that error rates do not have a meaning by themselves. They just provide a practical means to compare different criteria efficiencies. More important is to check the kind of errors that a security rule makes : either near the fixed boundary, and thus somewhat unavoidable, or genuine misclassification errors, which cause the rule to be inapplicable in some cases.

It is thus of paramount importance to be able to know which operating points lead to the classification errors, in order to check their actual security margin first, but also to reload them in the simulation tool in order to analyze why the rule does not correctly interpret their robustness, and have a feedback on a possible missing parameter in this rule.

9.4.2 Contingency severity analysis

The preceding analysis directly answers the practical question of the system security. But some better results might be expected by decomposing this problem in two :

1. the pre-contingency state robustness ;
2. the impact of the contingency on the system.

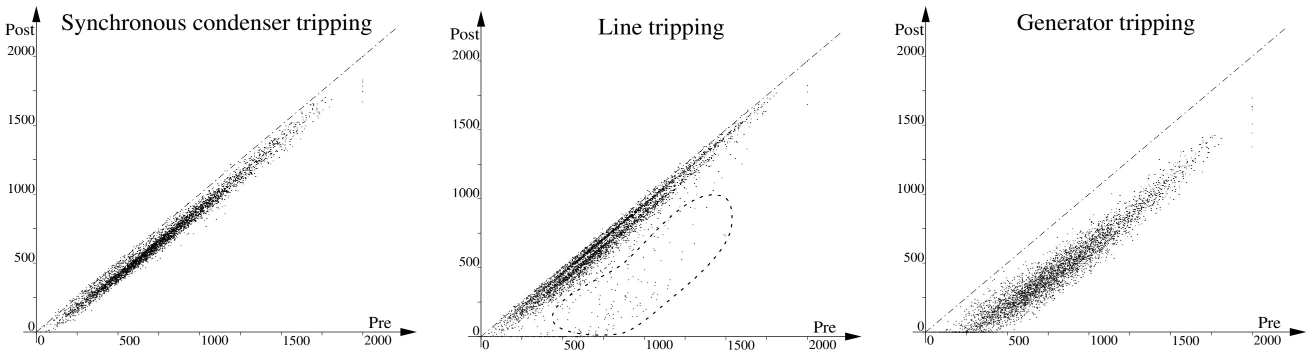


Figure 9.10 *Relation between pre- and post-contingency security margins. Adapted from [Weh96]*

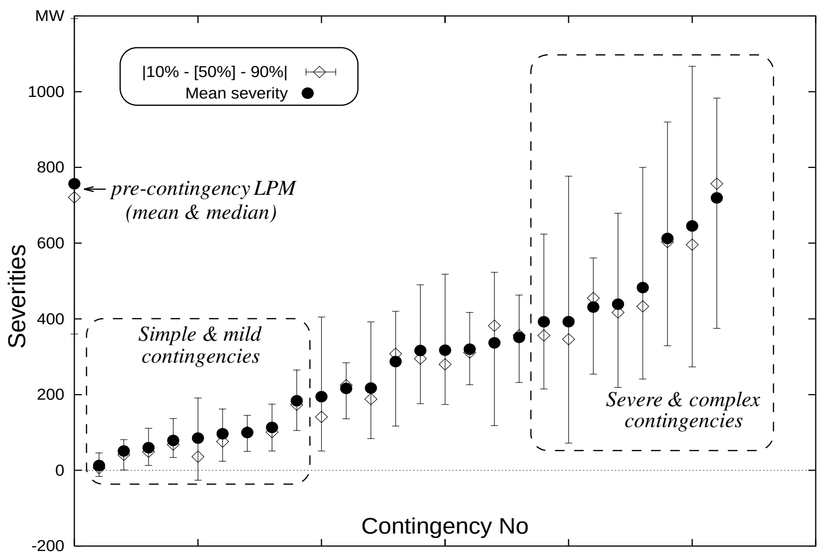


Figure 9.11 *Contingencies ranking by severities. Adapted from [Weh96])*

Indeed, it has been shown in our analysis that it is very difficult to find a security criterion efficient for several contingencies ; and on the other hand, a given contingency may have very different effects with respect to the system state characteristics. For instance, figure 9.10 illustrates the relation between pre and post-contingency margins for 3 different contingencies. Each point represents an operating state, and its vertical distance from the diagonal is equal to the severity of the contingency for this state. Thus the farther the location of the cloud center below the diagonal, the higher the mean severity of the contingency and the higher the spread of the points the more variable the severity from case to case.

These informations about the contingencies severities are summarized for the 26 contingencies by the chart of fig. 9.11, where the vertical bars show the diversity of these contingency severities. One can see how some contingencies have a rather constant effect, and how some others on the contrary lead to very diverse system robustness degradation, even for mild ones.

Use of regression techniques

In this context, it is interesting to be able to explain this diversity, which is expected to be linked to some very physical reasons, as an intermediate step towards the post-contingency state robustness assessment.

For this purpose, the regression techniques can be used. In order to get some interpretable results, the regression tree method has been employed [Weh96].

For the contingency considered above, i.e. the loss of one generator in power plant A, the tree selects 15 attributes among 138 : those which are the most correlated with the severity. These comprise by decreasing order of importance : the reactive flow through 400/225 kV transformers in substation E, the total reactive HV compensation in the region, the active flow through 400/225 kV transformers in plant A substation and the reactive reserve available in this plant. The regression tree remains however quite simple, since it is composed of 18 test nodes and 19 terminal nodes.

Like for the decision trees, the model can be further refined by using the continuous modeling capabilities of multilayer perceptrons.

Using this final result to approximate the value of severity of the test states yields a mean error of -0.8 MW and a standard error deviation of 43 MW. As compared with the margin computation precision (page 86), this result is considered as very satisfactory[3].

Interest of the contingency severity analysis

The first interest of this analysis is that it is suspected to provide more physical insight into the actual security problem, since it clearly decomposes the two aspects of pre-contingency state robustness, and contingency effect on that state. It should then lead to more interpretable results, and more precise security rules.

It may also be used in complement with an on-line DSA tool providing a pre-contingency margin computation. From one margin computation, one could benefit from the regression trees to get an estimation of all post-contingency margins, and thus assess the system security.

In operation planning, this kind of rules can provide a contingency ranking, leading some more detailed studies to the most dangerous contingencies. These rules have an advantage on any other ranking procedure : they indicate which changes can cause a contingency to become much more severe.

9.4.3 Another kind of application : emergency controls design

Tap changer blocking criterion

The preceding approach can also be applied for the system protection automata tuning problem. In this case, the candidate-attributes are any kind of variable liable to be available to the automaton; the output security classification can be kept unchanged.

For instance, for a transformer tap-changer blocking device, the efficiencies of criteria defined with local voltage measurements can be compared with some others defined with remote measurements, e.g. from some plants.

The best way to achieve this task would be to consider the variables as seen from the automata, i.e temporal sets of values. In our tests however, we simplified the problem by only considering two snapshots : one before an incident, and one 25 seconds after. This delay was chosen because the tap changers begin to act

[3] Regression trees for the two other contingencies shown at fig. 9.10 lead also to very satisfactory results, and can be found in [Weh96].

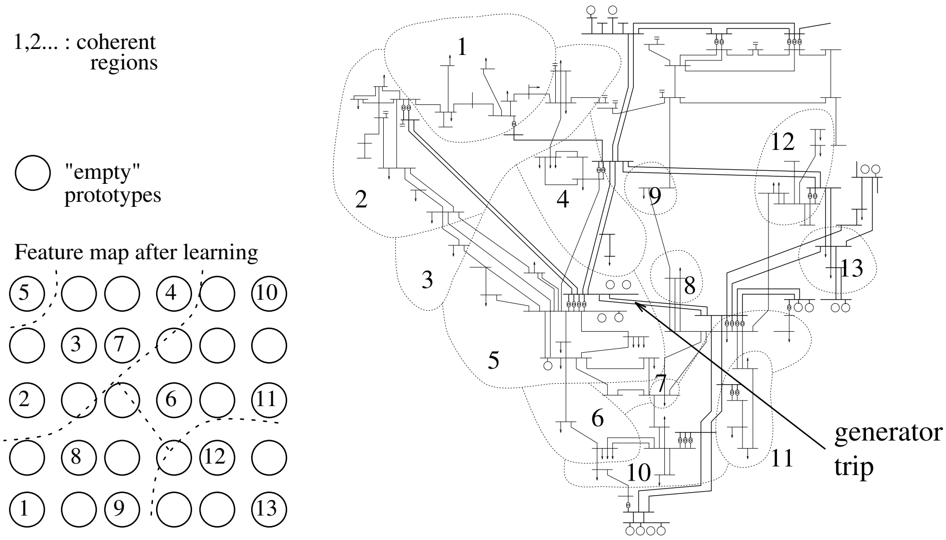


Figure 9.12 *Voltage coherent zones. Adapted from [WJ95]*

30 s after the voltages go outside the authorized dead-band. We refer the interested reader to [WVP+94] for more detailed results.

Actual design of emergency automata

The presented methodology can also be applied to investigations for an actual design and validation of various system protection schemes. Other classes of methods than automatic supervised learning may here be useful : unsupervised clustering techniques for instance allow the definition of coherent zones in the system. These zones can then be retained for load shedding schemes, or tap-changer blocking, . . .

Figure 9.12 illustrates this point by the use of a Kohonen feature map technique. The considered contingency is the loss of a generator in power plant A (This work is described in details in [WJ95]).

The feature map leads to the definition of 13 zones, according to the effect of the contingency on 39 HV voltages. From the top-left corner, we first encounter region 5 (the epicenter of the disturbance), then regions 4, 7, 3 and 2 (close to the disturbance epicenter), then two groups of peripheral regions (8, 9, 1) and (6, 10,11) and finally the remote regions 12 and 13, hardly affected by the disturbance. Thus from top-left to bottom-right corner of the map, we move from strongly to weakly disturbed load regions.

Such an automatically built map may help to find out the coherency zones in a network, and thus to design any kind of control or special protection.

10

Application perspectives in EDF

After 7 years of work in this field, EDF has a good experience in the applicability of the AL framework to actual power systems security assessment problems.

Two main kinds of applications are envisaged :

1. The definition of security rules, to be used either for on-line operation (to alert the operator) or for real-time monitoring and control (to trigger some automatic actions).

2. The design and validation of special protection and control systems.

EDF is also interested in a third subject : the possible links between the AL framework and the Monte-Carlo approaches used for system planning. Some work should be soon accomplished in this domain.

10.1 Security rules determination

For the moment, these researches brought the most directly applicable results for the preventive voltage security analysis problem. In this field, some operational studies are currently being carried out; due to the lack of industrial grade tools, it is however not yet possible to achieve these studies in operational environments.

However, the methodology proved its effectiveness on the other problems also.

- For the transient stability assessment problem, the improvements of computer resources make now possible to generate real-size data bases using the state-of-the-art analytical tools. Thus nothing prevents the approach from being applicable. Nevertheless, important transient stability studies are not a priority for EDF within the field of security assessment.

- For the emergency state detection problem, we expect a decisive improvement coming from the handling of temporal data, and the use of dedicated learning methods. This would indeed allow for considering attributes such as different snapshots, but also gradient. . .

10.2 Design and validation of protection and control systems

The global security study described in §7.6.2 is going on. It will be used to assess the interest and effectiveness of some special protection devices.

We do believe that such studies open a new dimension to security assessment. Future incidents may not look like the preceding ones, and this kind of study is the only way to appraise new system behaviors in an anticipative fashion.

But in the meantime, we intend to define a practical methodology to help the engineer designing some automatic protection systems, testing them and finally validating the retained one. This can be easily achieved by generating several data bases, considering once no such system, once one, once the other. . .

10.2.1 Towards a framework for better security/investment decisions

Another field in which EDF is interested is to experiment the coupling between the AL approach to dynamic security assessment and the classical Monte-Carlo approaches used for system planning.

The general objective consists of being able to better integrate the security problems in the investment decisions methodology. This becomes more and more needed since investment decisions concern more and more devices dedicated to alleviate dynamic security constraints, such as FACTS or more general protection schemes.

To do so, one must be able to evaluate for the different possible strategies, and in a probabilistic framework, (i) the probability and consequences of security problems, (ii) the operating costs and (iii) the investment costs. The best strategy is determined by combining all this information.

Here is a subject for work during the next few years.

11

Computer demonstration

During the tutorial a computer demonstration will be given in order to illustrate data mining of security information data bases.

The computer demonstration will show the use of low level statistical tools, various graphical visualizations of security scenarios, as well as application of decision tree induction and unsupervised learning. The coupling of the data mining software and the dynamic simulator will be illustrated on the basis of some interactive simulations of particular scenarios.

The illustrations will be carried out on the real-life data base described in Chapter 9. This data base is composed of 5000 security scenarios, each one analyzed with respect to 26 contingencies, and described by more than 400 attributes.

The data mining tool is a research software, continually developed at the University of Liège during the last ten years within the context of theoretical research on automatic learning and real-life security studies on the Hydro-Québec system and the system of Electricité de France. It is implemented in CommonLisp and runs on Unix workstations.

This software is presently used at the University of Liège in the context of research and at the R&D division of Electricité de France for field studies and research. It is also coupled with other automatic learning softwares, most of those described in this tutorial, and many more.

For those persons who are further interested by a private demonstration during PICA arrangements can be made after the tutorial.

Conclusions

12

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 material provided in Chapter 5 as well as 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. 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.

Finally, in order to make the preceding "theory" become credible, the last three chapters aimed at shedding some light on the way an electric utility which has extensively experimented on this topic considers the question.

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 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.

13

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.

13.1 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.

13.2 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.

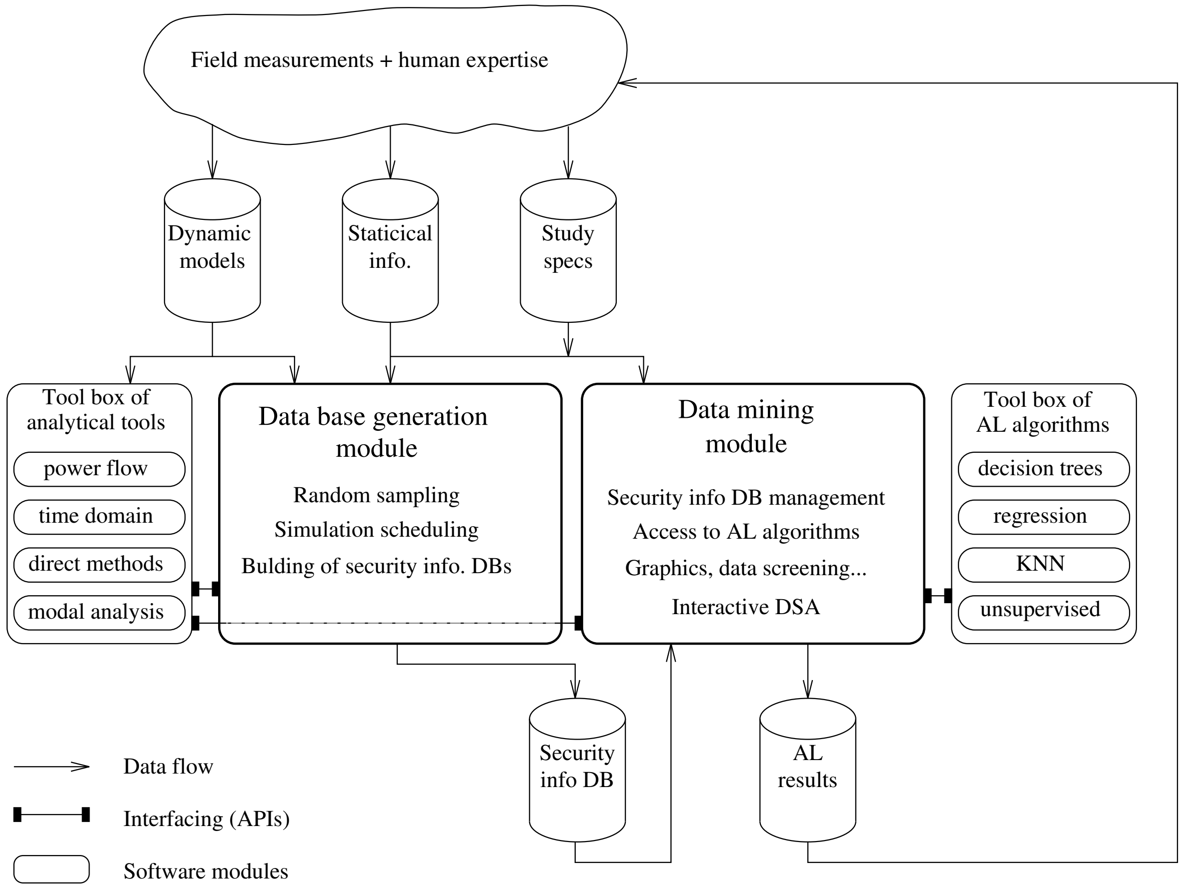


Figure 13.1 *Envisioned software architecture for the application of automatic learning to DSA*

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.

13.3 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 13.1 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.

[Bun92] W. L. Buntine. Learning classification trees. *Statistics and Computing*, 2:63–73, 1992.

[BW91] W. L. Buntine and A. S. Weigend. Bayesian back-propagation. *Complex Systems*, 5:603–643, 1991.

[Cyb89] G. Cybenko. Approximations by superpositions of a sigmoidal function. *Matt. Contro. Signals, Syst.*, 2:303–314, 1989.

[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.

[GJP95] F. Girosi, M. Jones, and T. Poggio. Regularization theory and neural networks architectures. *Neural Computation*, 7:219–269, 1995.

[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.

[HSW89] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[HYLJ93] J. N. Hwang, S. S. You, S. R. Lay, and I. C. Jou. What's wrong with a cascaded correlation learnign network : a projection pursuit learning perspective. Technical report, Info. Proc. Lab., Dep.t of Elec. Eng., University of Washington, September 1993.

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

[Kvå87] T. O. Kvålseth. Entropy and correlation: Some comments. *IEEE Trans. on Systems, Man and Cybernetics*, SMC-17(3):517–519, May - June 1987.

[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.

[MST94] D. Michie, D.J. Spiegelhalter, and C.C. Taylor, editors. *Machine learning, neural and statistical classification*. Ellis Horwood, 1994. Final rep. of ESPRIT project 5170 - StatLog.

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

[SW96] E. F. Sánchez-Úbeda and L. Wehenkel. The hinges model : a one-dimensional piecewise linear model. Technical report, University of Liège, September 1996. 48 pages.

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

[Weh97] L. Wehenkel. Discretization of continuous attributes for supervised learning. Variance evaluation and variance reduction. To appear in *Proc. of IFSA'97, Int. Fuzzy Systems Assoc. World Congress. Special session on Learning in a fuzzy framework*, 1997.

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

Security assessment

[CM97] TF 38.02.13 CIGRE and B. Meyer. New trends and requirements for dynamic security assessment. *To be submitted to ELECTRA*, 1997.

[DL68] 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.

[HTL+90] Y. Harmand, M. Trotignon, J. F. Lesigne, J. M. Tesseron, C. Lemaître, and F. Bourgin. Analyse d'un cas d'écroulement en tension et proposition d'une philosophie de parades fondées sur des horizons temporels différents. In *CIGRE Report 38/39-02, Paris*, August 1990.

[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.

[VPLH96] H. Vu, P. Pruvot, C. Launay, and Y. Harmand. An improved voltage control on large scale power systems. *IEEE Trans. on Power Syst.*, 11(3):1295–1303, August 1996.

[VJMP96] T. Van Cutsem, Y. Jacquemart, J.N. Marquet, and P. Pruvot. A comprehensive analysis of mid-term voltage stability. *IEEE Trans. on Power Syst.*, 1996.

Automatic learning applied to security assessment

[AWP+93] V.B. Akella, L. Wehenkel, M. Pavella, M. Trotignon, A. Duchamp, and B. Heilbronn. Multicontingency decision trees for transient stability assessment. In *Proc. of the 11th Power Systems Computation Conference*, pages 113–119, Aug-Sept 1993.

[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. *Revue E - SRBE - Special Issue on Automatic learning applied to power systems*, December 1996.

[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.

[JWP96] Y. Jacquemart, L. Wehenkel, and P. Pruvot. Practical contribution of a statistical methodology to voltage security criteria determination. In *Proc. of the 12th Power Systems Computation Conference*, pages 903–910, August 1996.

[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.

[VWP+93] T. Van Cutsem, L. Wehenkel, M. Pavella, B. Heilbronn, and M. Goubin. Decision tree approaches for voltage security assessment. *IEE Proceedings - Part C.*, 140(3):189–198, May 1993.

[WA93] L. Wehenkel and V.B. Akella. A hybrid decision tree - neural network approach for power system dynamic security assessment. In *Proc. of the 4th Int. Symp. on Expert Systems Application to Power Systems*, pages 285–291, Melbourne, Australia, January 1993.

[Weh93] L. Wehenkel. Decision tree pruning using an additive information quality measure. In B. Bouchon-Meunier, L. Valverde, and R.R. Yager, editors, *Uncertainty in Intelligent Systems*, pages 397–411. Elsevier - North Holland, 1993.

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

[Weh96] L. Wehenkel. Contingency severity assessment for voltage security using non-parametric regression techniques. *IEEE Trans. on Power Syst.*, PWRS-11(1):101–111, February 1996.

[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.

[WJ95] L. Wehenkel and Y. Jacquemart. Use of Kohonen feature maps for the analysis of voltage security related electrical distances. In *Proc. of ICANN'95, Int. Conf. on Artificial Neural Networks, NEURONîMES'95 (Industrial conference)*, pages 8.3.1–8.3.7, October 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.

[WP91] L. Wehenkel and M. Pavella. Decision trees and transient stability of electric power systems. *Automatica*, 27(1):115–134, 1991.

[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.

[WPEH94] L. Wehenkel, M. Pavella, E. Euxibie, and B. Heilbronn. Decision tree based transient stability method - a case study. *IEEE Trans. on Power Syst.*, PWRS-9(1):459–469, 1994.

[WVP+94] L. Wehenkel, T. Van Cutsem, M. Pavella, Y. Jacquemart, B. Heilbronn, and P. Pruvot. Machine learning, neural networks and statistical pattern recognition for voltage security: a comparative study. *Engineering Intelligent Systems for Electrical Engineering and Communications*, 2(4):233–245, December 1994.

[WVRP89] L. Wehenkel, T. Van Cutsem, and M. Ribbens-Pavella. An artificial intelligence framework for on-line transient stability assessment of power systems. *IEEE Trans. on Power Syst.*, PWRS-4:789–800, 1989.

Part IV

Copies of transparencies