PRUNING KERNEL REGRESSION TREES FOR SECURITY ASSESSMENT OF THE CRETE NETWORK

M. Helena Vasconcelos
INESC-Porto, Portugal Largo de Mompilher, 22, 4007 Porto Codex, Portugal
e-mail: hvasconcelos@inescn.pt

J. A. Peças Lopes
Faculdade de Engenharia da Universidade do Porto (FEUP) and INESC-Porto, Portugal
Largo de Mompilher, 22, 4007 Porto Codex, Portugal
e-mail: jpl@riff.fe.up.pt

ABSTRACT

This paper presents the capabilities provided by Kernel Regression Trees - a hybrid non-parametric regression technique - to on-line dynamic security assessment and monitoring of isolated power systems with high penetration of wind power. In the applied technique, to avoid overfitting a pruning algorithm is used to extract the security structure. This approach, which is demonstrated on the electrical power system of Crete island, proved to extract simple, interpretable, and reliable security structures. A description of the security problem and the data set generation procedure are included. Comparative results regarding performances of Regression Trees and Decision Trees are presented and discussed.

INTRODUCTION

In isolated power systems, like the ones operating in large islands, electric power is usually produced by Diesel units and gas turbines, resulting in high costs due to fuel imports and transportation. In these systems the production of electric energy from wind presents particular interest, especially when important wind energy potential exists, which is usual in many islands. Therefore, in these cases, significant savings of conventional fuels can be obtained by a high wind power penetration. However, it is important to ensure that the electric power system operation will not be adversely affected by an increased connection of this volatile form of energy.

The main problems faced by isolated electrical power systems are related to system security, control of frequency and management of system generation reserve. A common aspect to all these problems is the requirement to ensure that sufficient reserve capacity exists within the system to compensate for sudden loss of generation. Thus, mismatches in generation and load and/or unstable system frequency control might lead to system failures. This type of instability is termed frequency instability and depends on the ability of the system to restore balance between generation and load following a severe system disturbance with minimum loss of load (Kundur et al., 1997). Generally, frequency instability problems are associated with inadequacies in equipment responses, poor coordination of control and protection equipment or insufficient generation reserve.

In order to guard isolated power systems against foreseen disturbances and retain acceptable security levels, on-line dynamic security assessment functions can prove very valuable for their operation. Such functions have been developed and are integrated within an advanced control system tailored to the needs of small isolated power systems with increased wind power penetration. A pilot control system has been installed on the Greek island of Lemnos [6], an isolated Diesel-wind system with a peak load of approximately 10 MW. In this system, dynamic security assessment and monitoring are taken care of by two modules based on Decision Trees and Artificial Neural Networks. Decision Trees are used to check security for the operating schedules proposed by the economic dispatch module, with respect to characteristic wind power fluctuations. Neural Networks are used to give a real-time quantitative security evaluation of the current operating state system, by emulating the expected frequency deviation to the pre-define wind disturbance. In this way, the wind power penetration can be increased without jeopardizing the system security. To a more detailed description of the control system and the dynamic security assessment modules see (Peças Lopes et al., 1994), (Hatziargyriou et al., 1995) and [6].

The control system developed for small isolated power systems is currently being extended within the frame of the European R&D JOULE (JOR3-CT96-0119) project to cover the needs of larger isolated systems with high wind power penetration. Larger systems are characterized by several conventional fossil-fuelled generation plants and meshed transmission networks. The dynamic behavior performance of these systems depends not only on the total load and the size of the conventional units in operation, but also on their location and the response of the available spinning reserve [6].

The objective of this paper is to present the capabilities provided by Kernel Regression Trees - an hybrid non-parametric regression technique presented by Torgo in 1997 - to on-line dynamic security assessment and monitoring of these systems. The security evaluation structures provided by this approach are being integrated into CARE [7], the advanced control system that aims to achieve optimal utilization of renewable energy sources, in a wide variety of medium and large size isolated systems with diverse structures and operating conditions. The security evaluation structures that can be obtained provide a classification on dynamic security. Moreover, they also obtain the degree of security, which, in the Crete studied case addressed in this paper, is evaluated by emulating the expected minimum value of system frequency and maximal rate of frequency change for a selected disturbance.

It is shown that based on the Kernel Regression Tree proposed technique, simple, interpretable and reliable security structures can be provided. There are considered two approaches to design the security structures, differing in the way applied to avoid overfitting problems. The first one fights overfitting by applying directly stop-splitting rules during the growing algorithm of the tree structure. This first technique, although avoiding the tree to grow until having only pure leafs, does not looks for the right sized tree. In fact, much work was made centered on finding the appropriate stop-splitting rules for generating the tree with the right size (i.e. with a trade-off between *bias* and *variance*), where many variants were invented and tested (Breiman et al., 1984). From this work it was concluded that searching for the right stopping rule was the wrong way of looking at the problem. A more satisfactory procedure was found that consists of pruning instead of stopping. For this reason a pruning algorithm, which is described by Breiman et al. (1984), was also applied to design the tree structure. Regarding performance of Regression Trees and Decision Trees, it is shown that by applying the pruning algorithm to design the Kernel Regression Tree structures, besides obtaining reliable security structures, it is also possible to achieve simpler security rules. This last issue is considered highly relevant when applying machine learning techniques to medium and large power systems. Moreover, Kernel Regression Trees can provide both security classification and evaluation of security degree, whereas Decision Trees can only perform security classification.

THE STUDY CASE SYSTEM

The study case system is a realistic model of the power system of Crete, projected for the year 2000. It comprises several types of oil-fired units and a meshed 150 kV transmission network. The conventional generation system consists of two major power plants with twenty generating units installed. These are 6 Steam units of total capacity 103.5 MW, 4 Diesel units with 48 MW, 7 Gas turbines with 185 MW and one combined cycle plant with 132 MW. The plants are located near to the major load points. The system peak load is equal to 360 MW. The annual peak load demand occurs on a winter day and overnight loads can be assumed to be approximately equal to 25% of the corresponding daily peak loads. The base-load is mainly supplied by the steam and also by the Diesel units. The Gas turbine units normally supply the peak load at a high running cost, which increases significantly the average cost of the electricity being supplied.

A total of 11 Wind Parks (WPs) consisting of 160 Wind Turbines (WTs) with an installed capacity of more than 80 MW are or will be installed (have been approved) in Crete by the year 2000. These WPs will be connected at the MV (15 or 20 kV) network, which will be properly reinforced by new HV/MV substations. It is noted that with few exceptions, all WPs will be installed at the eastern part of the island, which presents the most favorable wind conditions. As a result, in case of faults on some particular lines the majority of the wind parks will be disconnected. Furthermore, the protections of the WTs might be activated in case of frequency variations, decreasing additionally the dynamic stability of the system. This might be caused by wind fluctuations, conventional unit outages, faults or other disturbing conditions.

Extensive simulations on the power system model have been performed using EUROSTAG software by NTUA, as described in Hatziargyriou et al. (1999). It is shown that for the most common wind power variations, the system remains satisfactorily stable, if sufficient spinning reserve is provided. On the other hand for various short-circuits and conventional unit outages, the system frequency undergoes fast changes and might reach very low values. In any case, the dynamic security of the system depends critically on the amount of spinning reserve provided by the conventional machines and the response of their speed governors. As an example, Figure 1 shows the change of the system frequency in two different dispatching conditions (1– fast thermal units that provide fast spinning reserve; 2– slower machines that provide slow spinning reserve), following the disconnection of three wind parks producing approximately 30 MW.

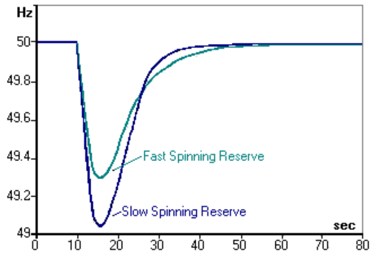


Figure 1 - Frequency change due to wind power loss

CREATION OF THE LEARNING & TEST SETS

The application of "learning from examples" techniques, such as the Kernel Regression Trees (KRTs) dealt within this paper, extract information from a large data set of pre-analyzed operating states of a power system, screened off-line via massive random sampling. For the Crete case study, the generation of the data set was developed by National Technical University of Athens (NTUA), within the framework of the CARE project. Each sampled scenario was pre-analyzed using an analytical tool of dynamic simulation – EUROSTAG software – to extract the minimum value of system frequency, f_{min}, and maximal rate of frequency change, df/dt_{max}, for each pre-defined disturbance. The generated data set (DS) was splitted in two sub-sets: a learning set (LS) and a testing set (TS). The learning set was required to extract the knowledge needed to derive automatic security evaluation structures, whereas the testing set was required to estimate their accuracy. Both consist of a large number of samples covering all possible states of the power system under study in order to ensure its representativity. Each sample is characterized by a vector of pre-disturbance steady-state variables, called candidate attributes, to define the system operating point (OP), which is labeled with the security indices f_{min} and df/dt_{max}. Candidate attributes can be either directly measured (powers, voltages etc.) or indirectly calculated quantities (wind penetration, spinning reserve etc.). The quality of the selected candidate attributes and the representativity of the LS and TS are very important for the successful implementation of the automatic structures.

For the creation of the global data set of Crete, a large number of initial operating points (OPs) was obtained by varying randomly the load for each load busbar, the wind power for each wind park and the wind margin. These variables were assumed to follow normal distributions around three operating profiles.

For each one of the 11 load busbars and each one of the 4 aggregate wind parks in operation, a perturbation of approximately ±10% was applied around each one of the three above referred operating profiles. A dispatch algorithm approximating actual operating practices followed in the control system of Crete was applied next in order to complete the pre-disturbance OPs.

For each one of the produced OPs a number of possible disturbances has been simulated, where EUROSTAG was used to obtain the system dynamic behavior. Two major disturbances have been finally selected. These are:

a) Outage of a major gas turbine;
b) Three-phase short-circuit at a critical bus near the wind parks.

These disturbances were selected according to utility criterion. In fact, a unit disconnection is a frequent event and a tree-phase fault, although rare, is a severe event that can occur during stormy conditions.

For each OP, both f_{min} and df/dt_{max} security indices were checked against the values that activate the frequency relays that protect the WPs, and the OPs were classified as "secure/insecure" accordingly. In this paper the variable used to verify security was the minimum frequency, f_{min}, the system experiments after each disturbance, where the security criteria used was:

If $f_{min} \leq$ 49 Hz **then** the OP is "insecure";
else the OP is "secure".

For the vector of candidate attributes that characterizes each OP, 22 operating parameters were selected, including:

- Total active and reactive load – $\sum P_L$ and $\sum Q_L$;
- Total conventional active generation – $\sum P_C$;
- Active and reactive power in the wind parks – P_W, $\sum P_W$, $\sum Q_W$;
- Spinning reserve and active generation in the conventional power plants – SR_i, P_{gi} and $\sum P_g$;

- Reactive generation in capacitor banks – $\sum Q_{cap}$;
- Wind power penetration – $WP = \sum P_W \,/ \sum P_L$;
- Wind margin – $WM = \sum SR / \sum P_W$.

Using the approach described in this section, 2765 acceptable samples have been obtained, which were divided in the two sets mentioned before, (by sending 2 samples to the LS and 1 to the TS). The LS comprises 1844 samples and the TS 921 samples. This partition was made having in mind that if the majority of the DS is used for testing purposes in order to ensure good estimates, then the quality of the extracted security structure will be reduced. On the other hand, if the majority of the DS is used for training purposes, then the testing errors will confer a wrong idea about the quality of the designed structure. According to Breiman et. al (1984) the TS is frequently taken as approximately 1/3 of the total samples, where the rest of samples belong to the LS. In the context of the pruning regression trees algorithm, Torgo (1998) claims that the best results are obtained using the following method for deciding the size of the TS:

$$\#\{TS\} = min(0.3 \times \#\{DS\}, 1000) \tag{1}$$

APPLICATION OF KERNEL REGRESSION TREES

As the Kernel Regression Tree (KRT) approach is being applied for the first time in this field, a short description of the main stages of the method are included in the next paragraphs. The first application of RTs in dynamic security assessment is due to Wehenkel (1995), and recently an application of a KRT approach in the voltage stability assessment problem was recently presented by Peças Lopes et al. (1998).

The Kernel Regression Tree (Torgo, 1997) is an hybrid algorithm that integrates recursive partitioning by Regression Trees (RT) with Kernel Regression (KR), dealing with continuous goal variables (i.e. regression problems). The regression problem consists in obtaining a functional model that relates the *output y* with the *inputs* a_1, a_2, ..., a_n (OP candidate attributes), where the *output y* (denominate as goal variable) is, in this case, a numerical value of any electrical security index of the power system. For the problem under analysis, the security index adopted is the minimum frequency - f_{min} (Hz).

The design of a RT (Breiman et al., 1984) consists in the extraction of interpretable security rules. The existing RT approaches differ in the predicting function used in the leafs. For instance, Breiman (1984) uses a mean value of *y*, whereas Karalic (1992) and Quinlan (1992) use a linear regression function. Kernel Regression models (Watson, 1964; Nadaraya, 1964), which is a non-parametric statistical methodology, provide quite opaque models of the data, but, on the other hand, are able to approximate highly non-linear functions. By integrating this regression procedure in the tree leafs, we can obtain a model that keeps the efficiency and interpretability of a RT, but with a better accuracy, by increasing the non-linearity of the functions used at the leafs. Moreover, KRTs achieve significantly better accuracy than RTs with smaller trees. By doing so, this hybrid model provides a better tradeoff between accuracy and simplicity than RTs, which is considerate highly relevant in real life applications. This last property can be seen in this paper, in the results obtained for the Crete power system.

Design of a Kernel Regression Tree

The design of a KRT involves two stages:

- Design of the regression tree (RT);
- Definition of the kernel regression model to make prediction in the tree leafs.

Starting with the learning set (LS), the design of a RT consists in explaining as much as possible the mean squared error of the security index *y* there observed. This corresponds to divide the samples of the LS into disjoint regions, in such a way that in each region the security index *y* is as constant as possible. This partition is defined by the leafs of the designed tree. In this paper there are considered two approaches to design the RT, differing in the way applied to avoid overfitting problems. The first one fights overfitting by applying directly stop-splitting rules during the growing algorithm of the RT. This first technique, although avoiding the tree to grow until having only pure leafs, does not looks for the right sized tree. In fact, much work was made centered on finding the appropriate stop-splitting rules for generating the tree with the right size (i.e. with a trade-off between *bias* and *variance*), where many variants were invented and tested (Breiman et al., 1984). From this work it was concluded that searching for the right stopping rule was the wrong way of looking at the problem. A more satisfactory procedure was found that consists of pruning instead of stopping. For this reason subsequently a pruning algorithm, which is described by Breiman et al. (1984), was applied to design the RT.

Design of a RT with Stop-Splitting Rules

In this approach, the design of a RT is determined by the following two issues:

- the optimal splitting test;
- the stop-splitting rules.

Starting with the root node, which corresponds to the LS, the growing of the RT is made by successively splitting their nodes. The splitting of a node is performed by a test defined as:

$$\{ a_k(sample) > u_k \}? \tag{2}$$

where u_k is the optimal threshold value of the chosen candidate attribute a_k. By applying this test to all the samples in the node, two successor nodes are created, which correspond to the two possible instances of the test $\{ a_k(sample) > u_k \}$ and $\{ a_k(sample) \leq u_k \}$. The split of each node must be performed according to an *optimal splitting test*, which corresponds to the splitting test that provides a maximum amount of information. Considering the mean value of *y* as the predicting function to use in the leafs, the optimal splitting test *"s"* of a node *"t"* is the one that minimizes the variance of *y* in the two successor nodes *"t_L"* and *"t_R"* resulting from the split, i.e. that maximizes:

$$\Delta s^2(y)_{s,t} = s^2(y)_t - \mathrm{P_L} \times s^2(y)_{t_\mathrm{L}} - \mathrm{P_R} \times s^2(y)_{t_\mathrm{R}} \tag{3}$$

where $s^2(y)_t$ is the variance of *y* at the learning samples stored in node *"t"*, P_L and P_R are the proportion of learning samples at the left and right successor nodes, and $\mathrm{s}^2(y)_{t_\mathrm{L}}$ and $s^2(y)_{t_\mathrm{R}}$ are the variance at the left and right successor nodes. This splitting rule is the one described by Breiman et al. (1984) and employed in CART.

The procedure continues splitting the created successor nodes, until a stop-splitting criterion is met for all the non-split nodes. This criterion used was the one described by Luís Torgo (1997), being defined by the two stop-splitting rules:

- <u>Rule 1</u>: It is not possible to further reduce variance of *y* in a statistically significant way. This corresponds to verify if a minimum number of learning samples, N_{min}, has been reached in the node;
- <u>Rule 2</u>: The variance of *y* has been sufficiently reduced. This corresponds to verify if a minimum value $s^2(y)_{min}$ as been reached, which corresponds to a perceptual value of the variance in the root.

Predicting with Kernel regressors

Given a new unseen operating point *Q*, a prediction for its security index is obtained by applying a regression model to the learning samples stored in the RT leaf that verifies the *Q* operating conditions. Kernel Regression models (Watson, 1964; Nadaraya, 1964) make prediction by a weighted average of the response *y* of the form:

$$y(Q) = \frac{\sum_{i=1}^{samples} K_h\left[D(Q,OP_i)\right] \times y_i}{\sum_{i=1}^{samples} K_h\left[D(Q,OP_i)\right]} \tag{4}$$

where $D(Q,OP)$ is the distance function (measures normalized distance between samples in the candidate attributes hyperspace), h is the bandwidth value and $K_h[x] = K[x/h]$, being $K(.)$ the Kernel function. The prediction is obtained using the samples (also denominated by *neighbors*) that are "most similar" to *Q*. This similarity is measured by means of the distance function. The Kernel function estimates the weight of each neighbor, given more weight to neighbors that are nearest to *Q*. The design of the kernel regression model includes the choice of the distance function, the bandwidth value, and the kernel function. In the implemented model it was used an Euclidean distance, a k-nearest neighbor (KNN) rule to define the bandwidth, and a Gaussian $K(d) = e^{-d^2}$ to define the kernel function. KNN method sets the bandwidth value *h* as the distance *D* to the k-nearest neighbor of *Q*. It also sets that only the k-nearest neighbors will be used to make prediction.

Kernel regression, and generally local modeling, can be very sensitive to the presence of irrelevant features, and so weighing can help to reduce this influence (Torgo, 1997). Atkeson et al. (1996) claims that the choice of the kernel function is not a critical design issue, as long as the function is reasonably smooth. These authors provide an extensive list of alternative kernel functions and discuss some of their merits.

Design of a RT with Pruning Algorithm

In the implemented KRT algorithm, it was applied the pruning procedure presented by Breiman et al. (1984), comprising the following stages:

- Design a very large regression tree, RT_{max}, which is supposed to overfit the LS.
- Generation of a sequence of pruned trees with decreasing complexity, $RT_1 \succ RT_2 \succ \ldots \succ root$ where $RT_1 \prec = RT_{max}$, by progressively pruning RT_{max} upward in the "right way" until being reached the root. Note that a subtree RT_1 of RT is referred as a pruned tree of RT if $root(RT_1) = root(RT)$, which can be denoted by $RT \succ RT_1$.
- Selection, among the sequence of pruned trees $\{RT\} = \{RT_1, RT_2, \ldots, root\}$ the right sized one, according to the minimization of an accurate estimation of the true predicting error of the corresponding KRTs structures.

To grow RT_{max}, one applied the previously described design procedure that exploits only the stop-splitting rules. The size of this initial tree is not critical as long as it is large enough to overfitt the LS. Then a selective pruning process is applied, that generates a reasonable number of pruned trees of RT_{max}, with decreasing size, such that each subtree is the "best" pruned tree in its size range. To select the "best" pruned tree a *minimal error-complexity criterion* is used. Considering that *T* is the binary tree structure of a regression tree *RT*, the *error-complexity measur*e of *RT* is defined by:

$$MSE^{LS}{}_{\alpha}(RT) = MSE^{LS}(RT) + \alpha \times \left|\tilde{T}\right| \tag{5}$$

where $MSE^{LS}(RT)$ (the error of *RT*) is the mean squared error of the *RT* when applied to the learning set, used to estimate the predicting error of the *RT* by taking as predicting function in the leafs to be the mean value of *y*; $\left|\tilde{T}\right|$ (the complexity of *RT*) is the number of leafs in the tree, and α (the penalty of the complexity) is a real number ≥ 0. Starting with $\alpha=0$, while α runs through a continuos value, the pruning process produces a finite sequence of pruned regression trees RT_1, RT_2, ..., *root* with progressively fewer terminal nodes. This is because each $RT(\alpha)$ is the minimizing subtree for a range of values of α, and therefore as α increases it continues being minimizing until a jump point α' is reached, where a new smaller subtree $RT(\alpha')$ becomes minimizing. The pruning process stops when the minimizing subtree becomes the root of RT_{max}.

Among the sequence of pruned trees {RT}, the algorithm selects the right sized one according to a *1 SE rule*. Following this rule, the chosen tree is the smallest one such that:

$$MSE^{TS}(KRT_K) \leq MSE^{TS}(KRT_{K_0}) + SE \tag{6}$$

where $MSE^{TS}(KRT_{K_0}) = \min\limits_{RT_i \in \{RT\}} MSE^{TS}(KRT_i)$

The $MSE^{TS}(KRT_i)$ of each RT_i is the estimation of the predicting error of its KRT_i structure (i.e., structure composed by the binary regression tree RT_i with a kernel regression function in the leafs) measured by the mean squared error that is obtained when applied to the testing set. *SE* is the standard error estimation of $MSE^{TS}(KRT_{k0})$, which is used to define the uncertainties of the $MSE^{TS}(KRT_{k0})$ estimation. Note that the selection of the right sized tree must be done according to the minimization of an accurate estimation of the true predicting error of the KRTs, whereas the application of the *1 SE rule* must be used instead of the minimization of $MSE^{TS}(KRT)$. One of the reasons is because the minimum position of $MSE^{TS}(KRT)$ might be unstable. In fact, small changes in parameter values, or even in how the LS and TS result from randomly separating the DS, might cause large changes in $\left|\tilde{T}\right|$ for the tree that minimizes $MSE^{TS}(KRT)$. By applying the *1 SE rule* it is possible to reduce that instability. Another reason to apply this rule is that it allows choosing the simplest tree whose accuracy is comparable to the one that minimizes $MSE^{TS}(KRT)$, and thus obtaining a better tradeoff between comprehensibility and accuracy.

NUMERICAL RESULTS

This section presents the results obtained with the proposed Kernel Regression Tree approach, to perform the dynamic security assessment of the Crete power system. Comparative results regarding performances of Regression Trees (RT) and Decision Trees (DT) are presented and discussed. Because of lack of space, only the results obtained for the goal variable f_{min} regarding a three-phase short-circuit disturbance are presented in this document.

As previously referred, the predicting accuracy of the results was estimated by using an independent pre-analyzed testing set (TS) with 921 samples. It was measured through the classification errors:

$$Global\ Classification\ Error\,(S) = \frac{\#\{\text{TS samples incorrectly classified by } S\}}{\#\{\text{TS samples}\}} \times 100\% \tag{7}$$

$$False\ Alarm\ Error\,(S) = \frac{\#\{\text{"secure" TS samples classified by } S \text{ as "insecure"}\}}{\#\{\text{"secure" TS samples}\}} \times 100\% \tag{8}$$

$$Missed\ Alarm\ Error\,(S) = \frac{\#\{\text{"insecure" TS samples classified by } S \text{ as "secure"}\}}{\#\{\text{"insecure" TS samples}\}} \times 100\% \tag{9}$$

and through quantifying mismatches relatively to the true goal values *y*, where the indicators used were the mean absolute error (*MAE*) and the root mean squared error (*RMSE*), i.e.:

$$MAE(S) = \frac{1}{N(TS)} \sum_{OP_i \in TS} \left| y_i - f_S(OP_i) \right| \,;\, RMSE(S) = \sqrt{\frac{1}{N(TS)} \sum_{OP_i \in TS} \left(y_i - f_S(OP_i) \right)^2} \tag{10}$$

In eqn. (11), $f_S(OP_i)$ is the *y* value assigned by the security structure *S* to the operating point i of the TS, whereas y_i is its true (pre-computed) value of *y*.

Regarding the sequence of KRT structures generated by the pruning algorithm previously explained, the graphical evolution of their predicting error (measured by the *RMSE*) as a function of their complexity (measured by |T| = number of nodes of the tree structure) is presented in Figure 2. Figure 3 presents a zoom of that evolution, being also presented the *predicting error/complexity* evolution for the set of generated RT structures.

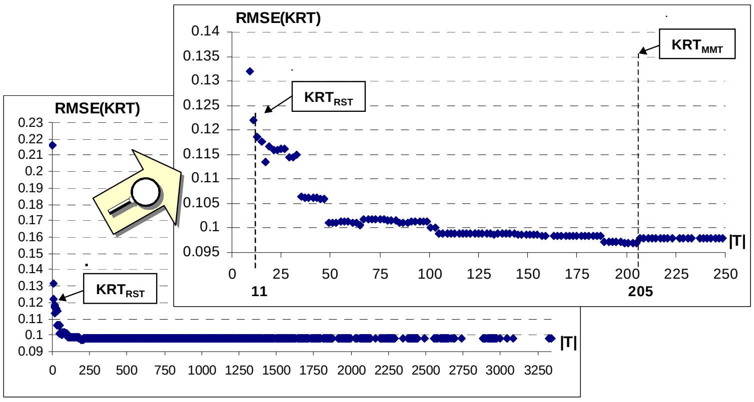


Figure 2 – *RMSE(KRT) versus |T|*, for the extracted set of pruned trees (Short-circuit, f_{min})

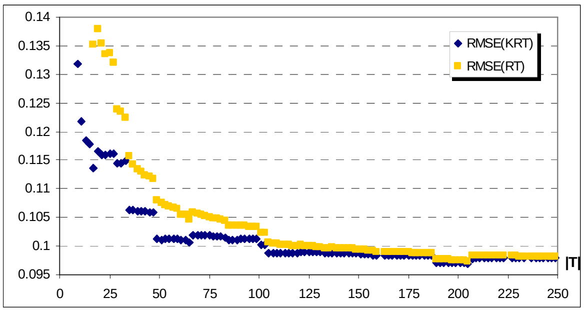


Figure 3 – Comparing accuracy and complexity between KRT and RT (Short-circuit, f_{min})

Figure 2 shows that starting with the most splitted tree (with 3341 nodes), as the tree initially decrease in size, the KRT predicting error decreases slowly. Then, at the tree with 205 nodes, it hits a minimum within in a valley region whereas the KRT predicting error has ups and downs. From this region forward, as the tree gets smaller the KRT predicting error increases rapidly. By applying the *1 SE rule* the tree with 11 nodes was selected as the right sized tree. In this figure, KRT_{RST} denotes the KRT structure that results from the right sized tree, whereas KRT_{MMT} denotes the KRT structure that results from the tree that minimizes *RMSE*.

As we can see in Figure 3, KRTs achieve significantly better accuracy than RTs with smaller trees. By doing so, this hybrid model provides a better tradeoff between accuracy and simplicity than RTs.

The performance evaluation results obtained for the KRT, RT and DT approaches are presented in Table 1. The Decision Tree results, obtained with an inductive inference procedure, are presented only for comparative purposes. A more complete description of the procedure used to derive DTs for this problem can be found in [5].

Besides the predicting errors, Table 1 also presents the K value used to define the bandwidth in the kernel regression model and the number of nodes of the binary tree structures. Regarding the performance results of the KRT approach, two security structures are addressed: the KRT_{RST} and the KRT_{MMT}. Regarding the performance results of the RT, the addressed RT structure is the one that resulted from applying the pruning algorithm previously described, where it was considered the mean value of y as the predicting function to use in the tree leafs. This structure is denoted by RT_{RST}.

Predicting Errors	KRT_{RST} (11 Nodes; K=7)	KRT_{MMT} (205 Nodes; K=7)	RT_{RST} (33 Nodes)	DT (23 Nodes)
MAE	0.03317	0.02495	0.05085	-
RMSE	0.12188	0.09699	0.12240	-
Global	2.71%	2.39%	6.51%	2.17%
False Alarm	2.38%	1.83%	7.88%	1.87%
Missed Alarm	3.20%	3.20%	4.53%	2.58%

Table 1 – Performance Evaluation Results (Short-circuit, f_{min})

Figure 4 presents the tree structure of the extracted KRT_{RST} security model, whereas the tree structure of the extracted *DT* can be observed in Figure 5. Nodes in KRT_{RST} are of two types: non-terminal and terminal nodes (leafs). In the root node (node number 1) we included information related with the total number of LS samples, the variance in the LS ($s^2(y)$ value) and the splitting test. In Figure 4 non-terminal nodes present the node number and also contain information related to the splitting test. In the leaf nodes we can get information related with the node number, the number of learning samples stored there (N), and the mean ($\bar{y}$) and variance ($s^2(y)$) of the security index y of those samples. To perform classification only based on these structures one can assigned a given degree of security to each leaf accordingly to the $\bar{y}$ value in the node. For the DT presented in Figure 5, each node presents the following: node number; number of learning samples stored in the node, safety ratio (= #{“secure” learning samples stored in the node}/#{learning samples stored in the node}) and the splitting test for non-terminal nodes. Leaf nodes with a safety ratio larger than 0,5 correspond to “secure” nodes.

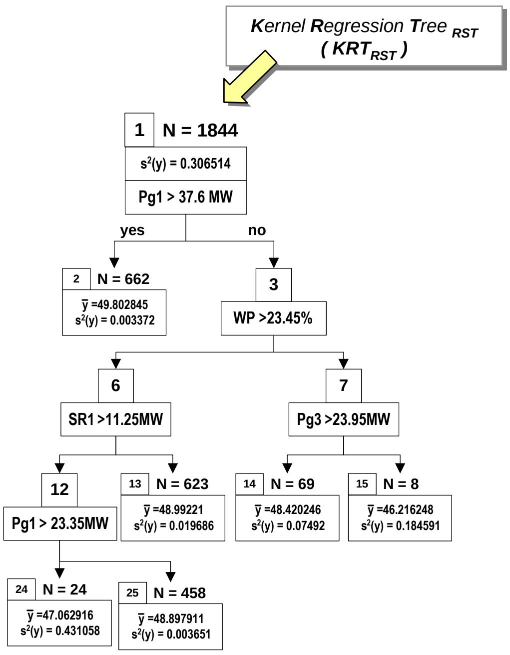


Figure 4 – Tree structure of the KRT_{RST} obtained for: [Short-circuit, f_{min}]

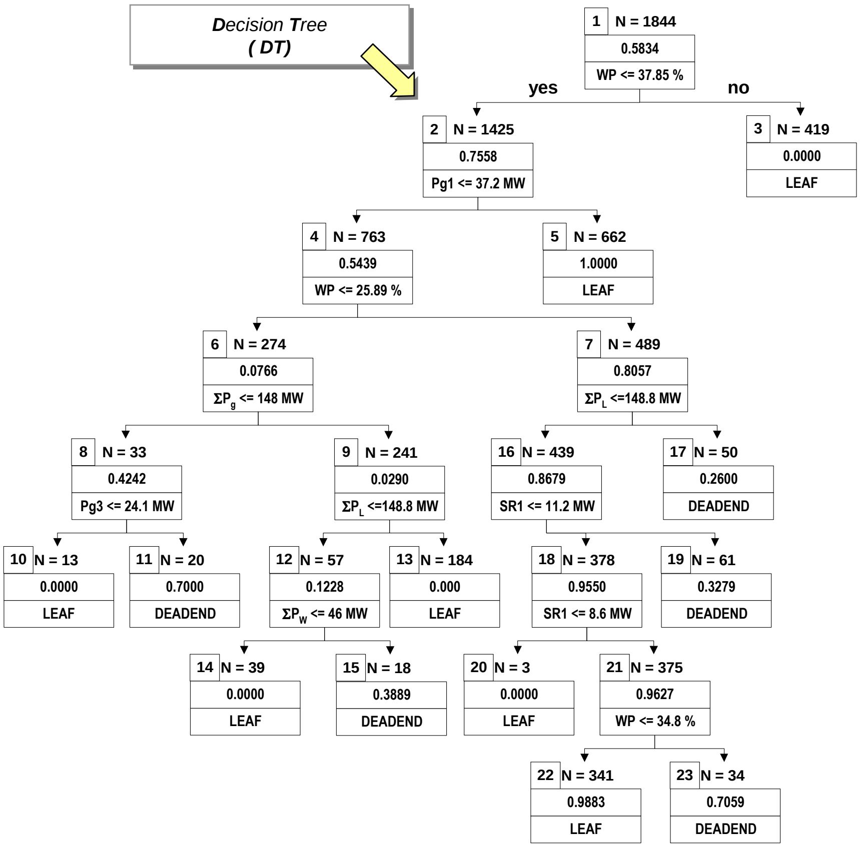
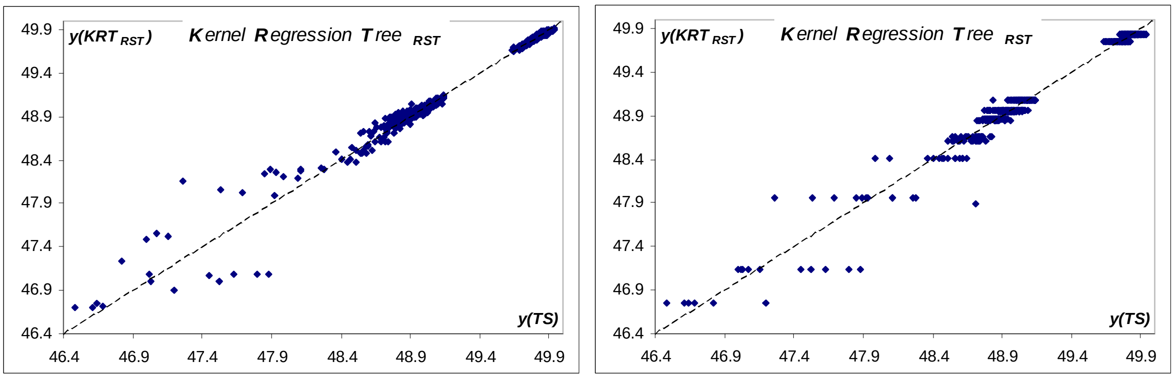


Figure 5 – Tree structure of the *DT* obtained for: [Short-circuit, f_{min}]

The scatter plot of the testing samples in term of their true value of *y – y(TS)* values – and obtained estimated value is presented in Figure 6a for the extracted KRT_{RST} structure and in Figure 6b for the extracted RT_{RST} structure.



(a) (b)

Figure 6 – True and predicted values obtained with KRT_{RST} and RT_{RST} for the TS samples (Short-circuit, f_{min})

Comparative Assessment

From the results obtained with the several approaches one can derive the following main conclusions:

– By selecting the kernel regression tree (KRT) that verifies the *1 SE rule*, instead of choosing the one that minimizes *MSE*, a significant reduction on the complexity of the extracted KRT structure was obtained (reduction from 205 to 11 nodes), keeping almost the same accuracy.

– The Kernel Regression Tree approach is able to provide security classification results and emulation of the numerical security index f_{min} in a coherent way and with good accuracy. Besides, KRTs provide simple interpretable security rules that can be adopted by operators in the control rooms to help them operating the system. Namely, by assigning the mean value $\overline{y}$ as the predicting function to be used in the leafs of the KRT structure presented in Figure 4, regarding the expected f_{min} that results from the short-circuit disturbance the following security rule can be extracted for the stated study case:

$$If \ Pg_1 > 37.1MW \ then \ the \ system \ is \text{" secure"}$$
$$Else \ the \ system \ is \text{" insecure"}$$

- Regarding the Regression Tree (RT) approach, Kernel Regression Tree (KRT) approach was able to provide security structures with better accuracy and simplicity.
- Regarding the Decision Tree (DT) approach, KRTs showed to design a classification structure with comparable performance but with a simpler structure, which makes easier any interpretation of the phenomena and of the influence of the relevant parameters. KRTs have the advantage of producing simultaneously a classification structure and giving the degree of robustness of the system through the predicted value of f_{min}.

CONCLUSIONS

This paper described the application of a hybrid machine learning approach oriented to deal with the evaluation of the dynamic security of a medium size power system. The security structures extracted with this approach will be integrated in the dynamic security assessment module of the advanced control system of the Crete island, helping to identify the operating conditions and parameters, namely wind power penetration, that lead to a less robust operation of the system. Comparative results regarding performances of others already known and applied machine learning techniques are presented and discussed.

REFERENCES

[1] Kundur, P., Morison, G.K. ,*"A Review of Definitions and Classification of Stability Problems in Today's Power Systems"*, Panel Session on Stability Terms and Definitions, IEEE PES Meeting, Feb. 2-6, 1997, New York.

[2] Peças Lopes, J. A., Fidalgo, J. N., Miranda, V., Hatziargyriou, N., *"Neural networks used for on-line dynamic security assessment of isolated power systems with a large penetration from wind production- A real case study"*, Proc. of Rough Sets and Soft Computing Conference'94, San José, USA, November 1994.

[3] Peças Lopes, J.A., Fernandes, F. *"Fast Evaluation of Voltage Collapse Risk Using Machine Learning Techniques"*, Proc. of VI SEPOPE, S. Salvador ad Baia, Brazil, May 1998.

[4] Hatziargyriou, N., Papathanassiou, S., Papadopoulos, M., *"Decision Trees for Fast Security Assessment of Autonomous Power Systems with large Penetration from Renewables"*, IEEE Transactions on Energy Conversion, Vol. 10, Nr. 2, June 1995.

[5] Hatziargyriou, N., Peças Lopes, J. A., Karapidakis, E., Vasconcelos, M. H., *"On-Line Dynamic Security Assessment of Power Systems in Large Islands with High Wind Power Penetration"*, accepted to 13th PSCC – Power systems Computation Conference, Trondhein, June 1999.

[6] ARMINES, NTUA, INESC, RAL, PPC, *"Development and implementation of an advanced control system for the optimal operation and management of medium-sized power systems with a large penetration from renewable power sources"*, Final report of EU-DG XII JOULE II project JOU2-CT92-0053. Edited by the Office for Official Publications of the European Communities, Luxembourg 1996.

[7] *"CARE: Advanced Control Advice for power systems with large scale integration of Renewable Energy sources"*, contract JOR3-CT96-0119, 3rd Project Report, July 1998.

[8] Wehenkel, L. ,*"Contingency severity assessment for voltage security using non-parametric regression techniques"*, IEEE PES 1995 Winter Meeting.

[9] Breiman, L., Friedman, H. F., Olshen, R. A., Stone, C. J., *"Classification and Regression Trees"*, Wadsworth International, 1984.

[10] Torgo, L., *"Kernel Regression Trees"*, poster papers of the European Conference, on Machine Learning (ECML-97), Internal Report of the Faculty of Informatics and Statistics, University of Economics, Prague. ISBN:80-7079-368-6, 1997.

[11] Torgo, L., *"Functional Models for Regression Tree Leaves"*, in Proceedings of the International Conference on Machine Learning (ICML-97), Fisher,D.(ed.), Morgan Kaufmann Publishers, 1997.

[12] Torgo, L., *"Error Estimates for Pruning Regression Trees"*, in Proc. of the 10th European Conference on Machine Learning (ECML-98), Nedellec,C. and Rouveirol,C. (eds.). Lecture Notes in Artificial Intelligence 1398, Springer Verlag, 1998.

[13] Watson, G. S., *"Smooth Regression Analysis"*, Sankhya: The Indian Journal of Statistics, Series A, 26: 359-372, 1964.

[14] Nadaraya, E., *"On estimating regression"*, Theory of Probability and its Applications, 9:141-142, 1964.

[15] Karalic, A., "Employing Linear Regression in Regression Tree Leaves", in proceedings of ECAI-92, Wiley & Sons, 1992.

[16] Quinlan, J. R., *"Learning with Continuos Classes"*, In Proceedings of the 5th Australian Joint Conference on Artificial Intelligence,. World Scientific, 1992.

[17] Atkeson, C. G., Moore, A. W., Schaal, S., (in press), *"Locally Weighted Learning"*, Special issue on lazy learning, D. Aha (Ed.), Artificial Intelligence Review, 1996.

Acknowledgment: The authors would like to thank the financial support of PRAXIS XXI within the "Subprograma Ciência e Tecnologia do 2º Quadro Comunitário the Apoio" and to EU project JOR3-CT96-0119. They are also grateful to all the other members of the project JOR3-CT96-0119 for their contributions.