Radial Basis Functions for Dynamic Security Assessment of Power Systems with Increased Wind Power Penetration

A. E. Gavoyiannis, E. M. Voumvoulakis and N. D. Hatziargyriou
Department of Electrical and Computer Engineering,
National Technical University of Athens,
Zographou 15773, Athens, Greece

Abstract— **This paper addresses the problem of dynamic security classification of electrical power systems using class pattern recognition with Radial Basis Functions (RBFs) by off-line supervised learning. A weight decay with a regularization learning algorithm is employed and the interest is focused on single-layer networks with functions which are fixed in position and size. Thus the expensive nonlinear gradient descent algorithms is avoided. Also using linear algebra makes analysis easier and computations quicker. Two cases are used and compared for RBFs. The first case is RBFs with regularization and the second case is RBFs without regularization. The classification is achieved by estimation of the output variables and its subsequent classification according to this estimation. The proposed method has been applied to security classification of the Greek Mainland system projected to 2005 with increased wind power penetration in a critical area. The voltage level in the area is used to clarify security. For the verification of the method experiments have been performed using independent input load and wind data produced with a random procedure and the classification of each testing sample has been compared with the one of the corresponding stored testing sample.**

Keywords: Dynamic Security, Radial Basis Function, off-line learning.

I. INTRODUCTION

Dynamic Security assessment is probably the most versatile field of application of automatic learning techniques in power systems. Moreover with the present evolutions of power systems, security assessment is becoming more and more challenging. Several Dynamic Security Assessment (DSA) methods have been developed aiming to assess the dynamic performance of the system for each critical contingency. Several of these methods are based on Artificial Intelligence techniques, like Decision Trees (DTs), Kernel Regression Trees (KRTs) and Artificial Neural Networks (ANNs) [1], [2], [3], [4], [5], [6]. In the last years, Support Vector Machines (SVMs) [7], [8], [9], [10] and combined SVMs [11] have been applied to DSA with good results. All the above methods belong to the category of supervised off-line learning. Several of the above methods have proved suitable for DSA classification, while other methods for DSA regression. In classification problems the task is to assign new inputs to one of a number of discrete classes. In regression problems [12], the outputs represent the values of continuous variables. In fact, the term *regression* refers to a specific kind of function defined in terms of an average over a random quantity. Both classification and regression problems can be seen as particular cases of *function approximation*. In the case of regression problems it is the regression function which will be approximated, while for classification problems the functions which are to be approximated are the probabilities of membership of the different classes expressed as functions of the input variables.

This paper focuses on the application of RBFs on DSA by off-line supervised learning. The last few years a number of approaches, that use RBFNs, are used as alternative methods [13], [14], [15], for the prediction of power systems functions. Neural networks, including radial basis functions, are *nonparametric* models and their parameters (e.g. weights) have no particular meaning in relation to the problems to which they are applied. Estimating values for the weights of a neural network is never the primary goal in supervised learning. The primary goal is to estimate the underlying function or at least to estimate its output at certain desired values of the input. On the other hand, the main goal of parametric regression often is the estimation of parameter values because of their intrinsic meaning. In this paper a weight decay with regularization learning algorithm is employed and the interest is focused on single-layer networks with functions which are fixed in position and size. Thus the expensive nonlinear gradient descent algorithms (such as the conjugate gradient and variable metric methods) that are employed in explicitly nonlinear networks, is avoided. Also using linear algebra makes analysis easier and computations quicker.

The proposed method has been applied to security classification of the Greek mainland system with increased wind power penetration in a critical area. The *voltage level* in the area is used to classify security. For the verification of the method experiments have been performed using independent input load and wind data produced with a random procedure and the classification of each testing sample has been compared with the one of the corresponding stored testing sample.

II. RADIAL BASIS FUNCTIONS NETWORKS

Radial functions are a special class of function. Their characteristic feature is that their response decreases (or increases) monotonically with distance from a central point. The centre, the distance scale, and the precise shape of the radial function are parameters of the model, all fixed if it is linear.

The most general formula for any RBF is

$$h(x) = \phi\left((x-c)^T R^{-1}(x-c)\right), \qquad (1)$$

where ϕ is the function used, c is the centre and R is the metric. The term $(x-c)^T R^{-1}(x-c)$ is the distance between the input vector x and the centre c in the metric defined by R. There are several types of functions used, for example, the Gaussian, $\phi(z) = \exp^{-z}$, the multiquadric, $\phi(z) = (1+z)^{1/2}$, the inverse

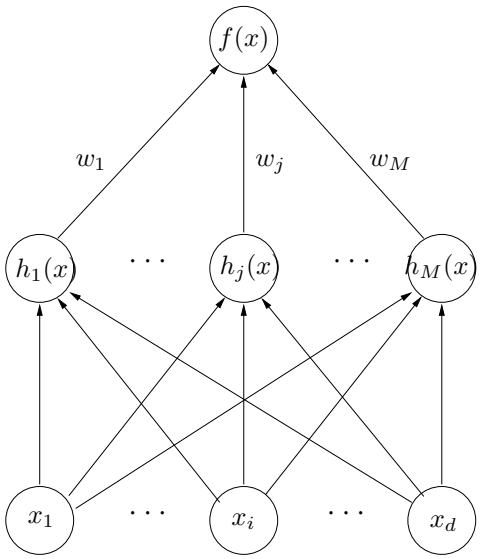

Fig. 1. The traditional RBFN.

multiquadric, $\phi(z) = (1+z)^{-1/2}$ and the Cauchy $\phi(z) = (1+z)^{-1}$.

Often the metric is Euclidean. In this case $R = r^2 I$ for some scalar radius r and where I is unity diagonal. If Gaussian is used then (1) simplifies to transfer function,

$$h(x) = \exp\left(-\frac{(x-c)^T(x-c)}{r^2}\right). \tag{2}$$

Its parameters are its centre c and its radius r and monotonically decreases with distance from the centre.

Radial functions could be employed in any sort of model (linear or nonlinear) and any sort of neural network (single-layer or multi-layer). However, RBFNs have traditionally been associated with radial functions in a single-layer network such as the one shown in the Fig. 1. Each of d components of the input vector x feeds forward to M basis functions whose outputs are linearly combined with weights $w_j, j = 1 \ldots M$ into the network output $f(x)$, as,

$$f(x) = \sum_{j=1}^{M} \omega_j h_j(x), \tag{3}$$

and the free variables are the weights ω. The model f is expressed as a linear combination of a set of M fixed functions h which are often called *basis functions*.

The flexibility of f, its ability to fit many different functions, derives only from the freedom to choose different values for the weights. The basis functions and any parameters which they might contain are fixed. If this is not the case, that is, the basis functions can change during the learning process or if there are more than one hidden layer, then the model is nonlinear.

Linear models are simpler to analyze mathematically. In particular, if supervised learning problems are solved by *least squares* then it is possible to derive and solve a set of equations for the optimal weight values implied by the training set. The same does not apply for nonlinear models, such as multilayer perceptrons, which require iterative numerical procedures for their optimization.

A. Least Squares and the Optimal Weight Vector (Learning Process)

When applied to supervised learning with linear models the least squares principle leads to a particularly easy optimization problem. Since the applied model is like (3) and the training set is $(x_i, y_i), i = 1, \ldots, N$, then the least squares recipe is to minimize the *sum-squared-error*,

$$S = \sum_{i=1}^{N} \left(y_i - f(x_i)\right)^2, \tag{4}$$

with respect to the weights of the model.

If a weight penalty term is added to the sum-squared-error, as is the case with ridge regression, then the following *cost function* is minimized [16],

$$C = \sum_{i=1}^{N} \left(y_i - f(x_i)\right)^2 + \sum_{j=1}^{M} \lambda_j \omega_j^2, \tag{5}$$

where the $\lambda_j, j = 1 \ldots M$ are non-negative regularization parameters.

For the optimization of the j-th weight, the differential of the cost function (5) is carried out, that is,

$$\frac{\vartheta C}{\vartheta \omega_j} = 2 \sum_{i=1}^{N} \left(f(x_i) - y_i\right) \frac{\vartheta f}{\vartheta \omega_j}(x_i) + 2\lambda_j \omega_j. \tag{6}$$

Substituting the derivative of the model (3), $\frac{\vartheta f}{\vartheta \omega_j}(x_i) = h_j(x_i)$, into (6) and equating the result to zero leads to,

$$\sum_{i=1}^{N} f(x_i) h_j(x_i) + \lambda_j \omega_j = \sum_{i=1}^{N} y_i h_j(x_i). \tag{7}$$

There are M such equations, for $1 \leq j \leq M$, each representing one constraint on the solution. Since there are exactly as many constraints as there are unknowns the system of equations has, except under certain pathological conditions, a unique solution. To find that unique solution linear algebra is employed. The equation (7) can be rewritten in vector notation as follows,

$$h_j^T f + \lambda_j w_j = h_j^T y. \tag{8}$$

Since there is one of these equations (8) for each value of $j = 1, \ldots, M$, can be formed them in matrix form,

$$H^T f + \Lambda w = H^T y, \tag{9}$$

where,

$$\Lambda = \operatorname{diag}\begin{bmatrix} \lambda_1 & \lambda_2 & \cdots & \lambda_M \end{bmatrix}, \tag{10}$$

and where H, which is called the *design matrix*, has the vectors $h_j, j = 1, \ldots, M$ as its columns, $H = [h_1\, h_2 \ldots h_M]$, and has N rows, one for each pattern in the training set.

The vector f can be decomposed into the product of two terms, the design matrix and the weight vector, since each of its components is a dot-product between two M-dimensional vectors. For example, the i-th component of f when the weights w are at their optimal values is,

$$f_i = f(x_i) = \sum_{j=1}^{M} h_j(x_i) \omega_j \tag{11}$$

and in the matrix form, $f = Hw$

Finally, substituting this expression for f into (9) gives,

$$H^T y = H^T f + \Lambda w = H^T H w + \Lambda w = \left(H^T H + \Lambda\right) w, \quad (12)$$

and the solution to which is,

$$w = \left(H^T H + \Lambda\right)^{-1} H^T y. \quad (13)$$

The equation (13) is the most general form of the normal equation and is called *local ridge regression*. There are two special cases. In *standard ridge regression* $\lambda_j = \lambda, \quad 1 \leq j \leq M$, that is,

$$\omega = \left(H^T H + \lambda I_M\right)^{-1} H^T y, \quad (14)$$

and *ordinary least squares*, where there is no weight penalty, and is obtained by setting all regularization parameters λ to zero, that is,

$$w = \left(H^T H\right)^{-1} H^T y. \quad (15)$$

The linear equations (13,14,15) can all be written more conveniently as the matrix equation,

$$Aw = H^T y, \quad (16)$$

where H (*design matrix*), is orthogonal of dimensions $N \times M$, the *variance matrix*, A^{-1}, is

$$A^{-1} = \left(H^T H + \Lambda\right)^{-1}, \quad (17)$$

the matrix Λ is determined of (10) and $y = [y_1 y_2 \ldots y_N]^T$ is the vector of training set outputs. The solution is the so-called *normal equation*,

$$w = A^{-1} H^T y, \quad (18)$$

and $w = [\omega_1 \omega_2 \ldots \omega_M]^T$, is the vector of weights which minimizes the cost function (5).

B. Maximum Likelihood and the EM Algorithm

It has been seen that the model estimated by a linear neural network from noisy samples $(x_i, y_i), i = 1, \ldots, N$ can be written as (3), where the $h_j, j = 1, \ldots, M$ are fixed basis functions and $\omega_j, j = 1, \ldots, M$ are the unknown weights which must be estimated. The vector of residual errors between model and data is, $e = y - Hw$, where H is the design matrix and has elements $H_{ij} = h_j(x_i)$. In a Bayesian approach to analyzing the estimation process, the *a priori* probability of the weights w can be modelled as a Gaussian of variance ς^2,

$$p(w) \propto \varsigma^{-M} \exp\left(-\frac{w^T w}{2\varsigma^2}\right). \quad (19)$$

The conditional probability of the data y given the weights w can also be modeled as a Gaussian, with variance σ^2, to account for the noise included in the outputs of the training set, $y_i, i = 1, \ldots, N$,

$$p(y|w) \propto \sigma^{-N} \exp\left(-\frac{e^T e}{2\sigma^2}\right). \quad (20)$$

The joint probability of data and weights is the product of $p(w)$ with $p(y|w)$ and can be represented as an equivalent cost function by taking logarithms, multiplying by -2 and dropping constant terms to obtain,

$$E(y, w) = N \ln \sigma^2 + M \ln \varsigma^2 + \frac{e^T e}{\sigma^2} + \frac{w^T w}{\varsigma^2}. \quad (21)$$

The conditional probability of the weights w given the data y is found using Bayes rule, again involves the product of (19) and (20) and is another Gaussian,

$$p(w|y) \propto p(y|w)p(w) \propto |W|^{-1/2} \exp\left(-\frac{1}{2}(w - \hat{w})^T W^{-1} (w - \hat{w})\right), \quad (22)$$

where $\hat{w} = A^{-1} H^T y, \quad W = \sigma^2 A^{-1}, \quad A = H^T H + \lambda I_m$, and $\lambda = \frac{\sigma^2}{\varsigma^2}$ is the regularization parameter.

Finally, the marginal likelihood (probability) of the data is,

$$p(y) = \int p(y|w)p(w)dw \propto \sigma^{-p} |P|^{1/2} \exp\left(-\frac{y^T P y}{2\sigma^2}\right), \quad (23)$$

where $P = I_p - H A^{-1} H^T$, is the projection matrix.

Note that there is an equivalent cost function for $p(y)$ which is obtained in the same way as (21), that is,

$$E(y) = N \ln \sigma^2 - \ln |P| + \frac{y^T P y}{\sigma^2}. \quad (24)$$

The key elements in these criteria are the projection matrix P and the effective number of parameters γ in the network. In ridge regression the projection matrix P is not exactly a projection and the effective number of parameters is not equal to M, the number of weights. To be as general as possible will be used ridge versions for both the projection matrix and the effective number of parameters. However, the ordinary least squares versions of the selection criteria can always be obtained simply by setting all the regularization parameters to zero and remembering that the P is idempotent $(P^2 = P)$.

The Expectation-Maximization (EM) algorithm [17],[18] performs maximum likelihood estimation for problems in which some of the variables are unobserved and estimates the parameters of a model iteratively, starting from some initial guess. Each iteration consists of an expectation (E) step which finds the distribution for the unobserved variables and a maximization (M) step which re-estimates the parameters of the model to be those with the maximum likelihood for the observed and missing data combined.

In the E-step, the expectation of the conditional probability of the missing data (22) is taken and substituted, in the M-step, into the joint probability of the combined data, or its equivalent cost function (21), which is then optimized with respect to the model parameters σ^2 and ς^2. These two steps are guaranteed to increase the marginal probability of the observed data and when iterated convergence to a local maximum.

Detailed analysis [16] results in a pair of re-estimation formula for the parameters σ^2 and ς^2,

$$\sigma^2 = \frac{e^T e + \gamma \sigma^2}{N}, \quad (25)$$

$$\varsigma^2 = \frac{w^T w + (M - \gamma)\varsigma^2}{M}, \quad (26)$$

where $e = y - Hw$ and $\gamma = M - \lambda \mathrm{trace}\mathrm{A}^{-1}$. Initial guesses are substituted into the right hand sides which produce new guesses. The process is repeated until a local minimum of (24) is reached.

C. Optimizing the Parameters of RBFs

Another key parameter is the size of the RBFs. This section describes a simple scheme to find an overall scale size for the RBFs in a network. This method chooses the best overall *size* for the RBFs from a number of trial values which is rendered tractable by the efficient optimization of λ.

In a linear model as (3) with fixed basis functions h_j and weights $\omega_j, j = 1, \ldots, M$, the model complexity can be controlled by the addition of a penalty term to the sum of squared errors over the training set, $(x_i, y_i), i = 1, \ldots, N$. When the combined error,

$$E = \sum_{i=1}^{N} (y_i - f(x_i))^2 + \lambda \sum_{j=1}^{M} \omega_j^2, \tag{27}$$

is optimized, large components in the weight vector w are inhibited. This kind of penalty is known as *ridge regression* or *weight-decay* and the parameter λ, which controls the amount of penalty, is known as the *regularization* parameter. While the nominal number of free parameters is M (the weights), the effective number is less, due to the penalty term, and is given by $\gamma = M - \lambda \mathrm{trace}\mathrm{A}^{-1}$, where $A = H^T H + \lambda I_m$, and H is the design matrix with elements $H_{ij} = h_j(x_i)$. The expression for γ is monotonic in λ so model complexity can be decreased (or increased) by raising (or lowering) the value of λ.

The parameter λ has a Bayesian interpretation. It is the ratio of σ^2, the noise corrupting the training set outputs, to ς^2, the *a priori* variance of the weights. If the value of λ is known then the optimal weight vector is, $w = A^{-1} H^T y$. However, neither σ^2 nor ς^2 may be available in a practical situation so it is usually necessary to establish an effective value for λ in parallel with optimizing the weights. This may be done with model selection criterion (MSC) such as GCV or MML and in particular with one or more re-estimation formula [16] as,

$$\lambda = \frac{\eta}{N - \gamma} \frac{e^T e}{w^T A^{-1} w}, \tag{28}$$

where $e = y - Hw$ and $\eta = \mathrm{trace}\left(\mathrm{A}^{-1} - \lambda \mathrm{A}^{-2}\right)$.

An initial guess for λ is used to evaluate the right hand side of the above formula which produces a new guess. The resulting sequence of re-estimated values converge to a local minimum. Each iteration requires the inversion of $M \times M$ matrix A and therefore costs of order M^3 floating points operation.

The optimization of λ by iteration of the re-estimation formula is burdened by the necessity of having to compute an expensive matrix inverse every iteration. However, by a reformulation of the individual terms of the equation using the eigenvalues and eigenvectors of HH^T it is possible to perform most of the work during the first iteration and reuse the results for subsequent ones. Thus the amount of computation required to complete an optimization which takes q steps to converge is reduced by almost a factor of $1/q$. Unfortunately, the technique only works for a single global regularization parameter, not for multiple parameters applying to different groups of weights or to individual weights.

For Gaussian radial functions of fixed width r and transfer functions of the hidden units as (2), there is no re-estimation formula for r, as there is for λ, even in this simple case where the same scale is used for each RBF and each component of the input. To properly optimize the value of r would thus require the use of a nonlinear optimization algorithm and would have to incorporate the optimization of λ (since the optimal value of λ changes as r changes). An alternative is to test a number of trial values for r. For each value r an optimal λ is calculated by using (28) and the model selection score noted. When all the values have been checked, the one associated with the lowest score wins.

Suppose the eigenvalues μ_i and eigenvectors $u_i, i = 1, \ldots, N$ of HH^T and that the projections of y onto the eigenvectors are $\tilde{y}_i = y^T u_i$. Then, as shown in [16], the four terms involved in the re-estimation formula (28) are,

$$\eta = \sum_{i=1}^{N} \frac{\lambda}{\mu_i + \lambda}, \tag{29}$$

$$N - \gamma = \sum_{i=1}^{N} \frac{\mu_i}{(\mu_i + \lambda)^2}, \tag{30}$$

$$e^T e = \sum_{i=1}^{N} \frac{\mu_i \tilde{y}_i^2}{(\mu_i + \lambda)^3}, \tag{31}$$

$$w^T A^{-1} w = \sum_{i=1}^{N} \frac{\lambda^2 \tilde{y}_i^2}{(\mu_i + \lambda)^2}, \tag{32}$$

If λ is re-estimated by computing (29-32), instead of explicitly calculating the inverse in (28), then the computational cost of each iteration is only of order N, instead of M^3. The overhead of initially calculating the eigensystem, which is of order N^3, has to be taken into account but is only performed once. For problems in which N is not much bigger than M this represents a significant saving in computation time and makes it feasible to optimize multiple guesses for the initial value of λ to decrease the chances of getting caught in a local minimum.

III. Application System

The study case system is the model of the Greek mainland system projected to 2005. It comprises steam turbines, which produce the base load and almost 70% of the annual energy, hydro turbines which are used for peak load and frequency regulation and two new combined cycle units. The system is interconnected with the rest Balkan system with two 400 kV synchronous interconnections. There is also a new DC interconnection with Italy which connects the system directly to the European one. The peak load is about 9000 MW (summer 2002), while the off-peak about 3500 MW. The model includes the generation and transmission system up to 20 kV buses. The total number of buses, including the buses of the external system, is 1023, the number of branches (lines and transformers) 1296 and the generation units, some of which represent a group of generators, 115 in peak load and 86 in off-peak load.

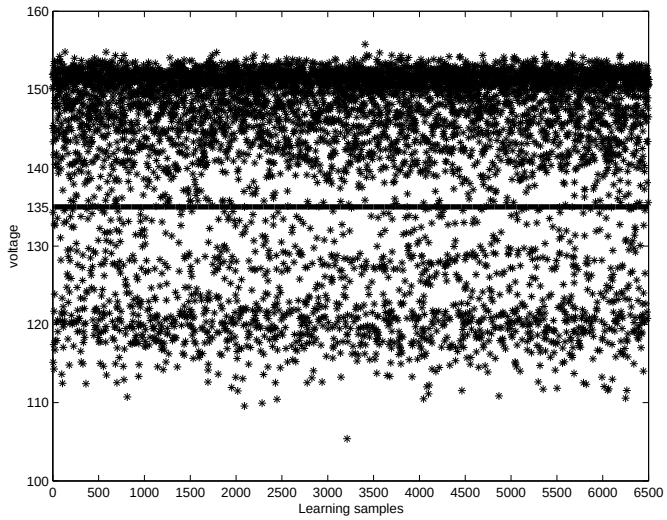

Fig. 2. Area voltage of data set.

Wind power to be installed in 2005 in the mainland system is estimated to about 1170 MW. In some areas the transmission system is not able to absorb the wind power, which is the case for the north-east of the system (area of Thrace). As a result in case of large wind power penetration it is possible that security problems appear due to low voltages caused from the large power production in the area. In this case, it is possibly required to reduce the production of one combined cycle unit in order to reduce the total production of the area.

IV. Creation of the Knowledge Base

The knowledge base is obtained from a large number of off-line dynamic simulations that define a data set. This data set is randomly partitioned in two sub-sets, a learning set and a testing set. The learning set is required to extract the knowledge needed to derive classification. It consists of a large number of operating points, covering all possible states of the under study system, in order to ensure its representativity. Each operation point is characterized by a vector of pre-disturbance steady-state variables, called attributes, that can be either directly measured (power, voltages, etc.) or indirectly calculated quantities (wind penetration, spinning reserve, etc.). The quality of the selected attributes and the representativity of the learning set are very important for the successful implementation of the classification.

For the creation of the data set a number of 6500 initial operating points are obtained by varying randomly the load for each bus in the area, the wind power produced by each park and the production of the combined cycle unit. Loads and wind powers are varying between 0.7 and 1.0 of their nominal values, while the combined cycle power is varying between its technical minimum and its technical maximum. For the study the system is considered to be in off-peak operation where the wind power effects are more severe. The state of the rest system does not change as it does not influence significantly the situation in the Thrace area. The simulated disturbance is a 15% increase of wind power in a time period of 1 minute. As the combined cycle unit can not react quickly this power is exported from the region. This increase of exported power can lead the voltage to dangerously low levels due to the low transfer capacity of the transmission lines in the area. At the end of the simulation the voltage level in the buses of the area is recorded. It is desired that all voltages are over a security threshold which has be taken 0.9 of nominal voltage, otherwise the system operation is considered unsatisfactory. The 0.9 p.u. voltage threshold has been set taking into account the under-voltage protections of the wind turbines which are usually set to 0.8-0.85 p.u., so voltages below 0.9 can easily lead to loss of large amount of wind power production even for a small disturbance.

TABLE I
Two-class partition for RBF.

Knowledge Base	Learning Set	Test Set
6500	3500	3000
Two-class Partition for RBF		
class A (secure)	3100	2336
class B (insecure)	400	664

Training is performed by random selection of the learning set, while the test set (I) is the rest of the data set. In this way, the learning rate is checked and the capability of the method to classify correctly unforeseen states can be evaluated on a more objective basis.

V. Results

In testing the design matrix H_t of data that is used for prediction is found first. In sequence the output of RBFN, $Y_{pred} = H_t\omega$ is computed and this is compared to real output.

Initially 9 attributes were selected as candidate attributes. Engineering judgement was used to select the attributes directly relevant to the voltage variations of the system. There are 2 attributes for the wind parks productions, which have been grouped according to their geographical location, 5 attributes for the active and reactive power of the total area and the sub-areas, one attribute for the production of the combined cycle unit and one attribute which is the total production minus the total load. This is the power that is exported from the area and is a very significant index for the security assessment.

By applying Principal Component Analysis (PCA) [19] the 9 attributes of learning set are reduced to 4 columns and the performance of the method is significantly increased.

The type of RBF is Gaussian with exp equal to 2 and bias equal to 1. The limits are between 0 and 10 for the distance scale r and for λ between 0 and 1. The scale found equal to 0.5 and λ equal to 0.0125.

The false alarms rate is defined as the ratio of the secure OPs that are classified as insecure to the secure OPs, while the missed alarms rate is defined as the ratio of the insecure OPs that are classified as secure to the total number of insecure OPs.

Table II shows the results of the two-class classification of the 3000 new test samples by RBF. From the evaluation values we find if the test sample belongs to secure or insecure state, accordingly to the specified criteria.

The maximum absolute error of the above comparison was 22.15 KV. The mse for the testing set is equal to 29.45 and the

TABLE II
EVALUATION OF TWO-CLASS CLASSIFICATION PERFORMANCE.

Classification of Test Set by RBF		
For LS=3500	class A	class B
Samples of class A(2336)	2176	160
Samples of class B(664)	34	634
Classification Performance Evaluation		
Success Rate	93.53% (2806/3000)	
False Alarms	6.85% (160/2336)	
Missed Alarms	5.12% (34/664)	

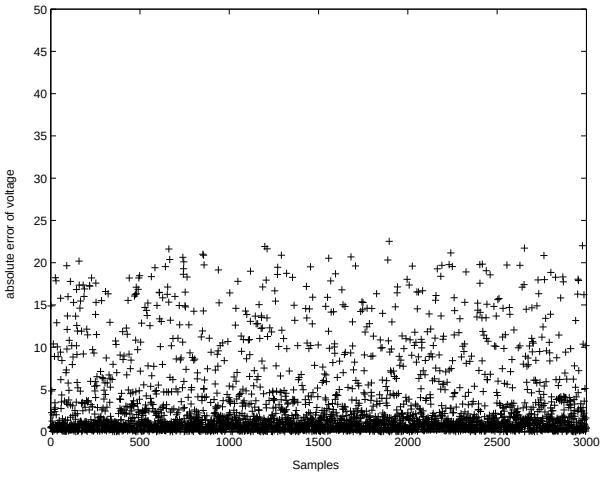

Fig. 3. Comparison of the estimated output variable to corresponding real values for the voltage value for RBF with regularization.

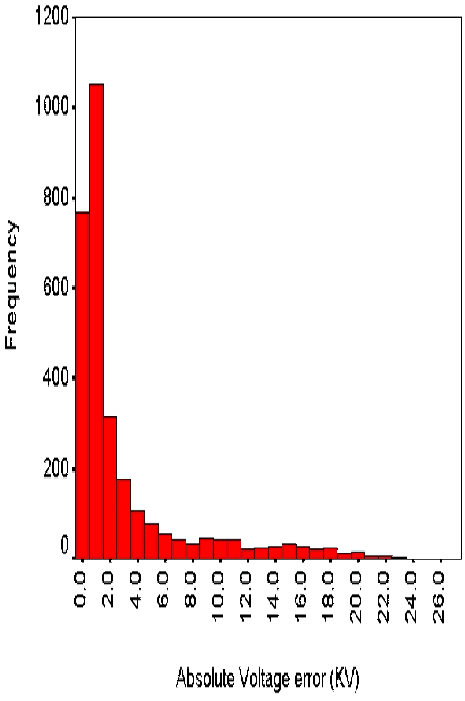

Fig. 4. Frequency of abslolute voltage error for RBF with regularization.

TABLE III
EVALUATION OF TWO-CLASS CLASSIFICATION PERFORMANCE.

Classification of Test Set by RBF with $\lambda = 0$		
For LS=3500	class A	class B
Samples of class A(2336)	2197	139
Samples of class B(664)	624	40
Classification Performance Evaluation		
Success Rate	94.03% (2821/3000)	
False Alarms	5.95% (139/2336)	
Missed Alarms	6.02% (40/664)	

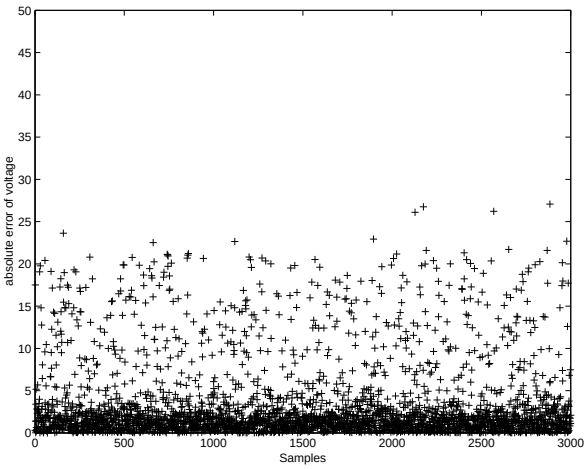

Fig. 5. Comparison of the estimated output variable to corresponding real values for the voltage value for RBF with $\lambda = 0$.

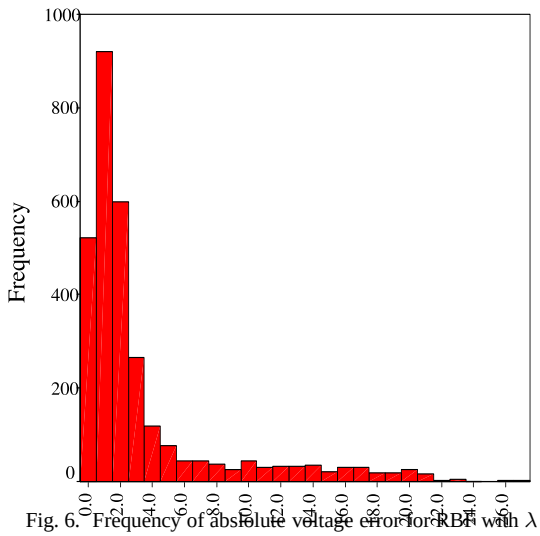

Fig. 6. Frequency of absolute voltage error for RBF with $\lambda = 0$.

sumsquareerror is equal to 572.43.

For the same conditions and for $\lambda = 0$ (standart ridge regression) the obtained results are shown in the Table III and Fig. 5. The mse for the testing set is equal to 33.75 and the absolute maximum error is equal to 27.06. The sumsquareerror is equal to 654.78.

For the RBFs with regularization the results of the comparison of the estimated output variable to corresponding real values is represented in Fig. 3 and the frequency of the absolute voltage error in Fig. 4.

From Fig. 3 and Fig. 5, it is observed, that the absolute maximum error for RBF with regularization is 18% lower than the maximum error which is computed from the application of RBF as standart ridge regression $(\lambda = 0)$.

VI. Conclusions

Results from the application of the method on operating point series from the Greek Mainland system show the accuracy and versatility of the method, which are well compared to other methods, which also use the learning data as batch.

The interest is focused on single-layer networks with functions, which are fixed in position and size. Thus the expensive nonlinear gradient descent algorithms (such as the conjugate gradient and variable metric methods) that are employed in explicitly nonlinear networks, is avoided. Also using linear algebra makes analysis easier and computations quicker.

Also, the performance for RBF with regularization is better than the one from the application of RBF as standart ridge regression, as expected from the theory of regularization.

References

[1] L. Wehenkel, "Automatic learning techniques in power systems," *Kluwer Academic, Boston*, 1998.

[2] M. El-Sharkawi and D. Neibur, "Artifi cial neural networks with applications to power systems," *IEEE-PES Special Publ. 96TP-112-0*, 1996.

[3] CIGRE, "New trends and requirements for dynamic security assessment," *TF 38.02.13*, 1997.

[4] D.J. Sobajic, "Fast frequency assessments for memory-augmented static contingency analysis," *Proc. Of ESAP'93, Melbourne, Australia*, pp. 332–337, Jan. 1993.

[5] L. Wehenkel and M. Pavella, "Decision tree approach to power system security assessment," *Int. J. Electrical Power and Energy Systems*, vol. 15, no. 1, Feb. 1993.

[6] D.J. Sobajic and Y.H. Pao, "Artifi cial neural-net based dynamic security assessment for electric power systems," *IEEE Transactions on Power Systems*, vol. 4, pp. 220–228, 1989.

[7] Vladimir Vapnik, *The nature of statistical learning theory*, Springer-Verlag, New York, 1995.

[8] A. J. C. Sharkey, Ed., *Combining Artificial Neural Nets*, Perspectives in neural computation. Springer, 1999.

[9] A. E. Gavoyiannis, D. G.Vogiatzis, and N. D. Hatziargyriou, "Dynamic security classifi cation using support vector machines," *International Conference on Intelligent System Application to Power Systems*, 2001.

[10] L.S. Moulin, A.P. Alves da Silva, M.A. El-Sharkawi, and R.J. Marks, "Neural networks and support vector machines applied to power systems transient stability analysis," *Engineering Intelligent System Journal, Victoria, AUSTRALIA*, 2001.

[11] A. E. Gavoyiannis, D. G.Vogiatzis, D. P. Georgiadis, and N. D. Hatziargyriou, "Combined support vector classifi ers using fuzzy clustering for dynamic security assessment," *IEEE 2001 PES Summer meeting,Vancouver,Canada*, 2001.

[12] C. M. Bishop, *Neural Networks for Pattern Recognition*, Clarendon Press, Oxford New York, 1995.

[13] A. G. Dutton, G. Kariniotakis, J. A. Halliday, and E. Nogaret, "Load and wind power forecasting methods for the optimal management of isolated power systems with high wind penetration," *Wind Engineering*, vol. 23, no. 2, June 1999.

[14] G. Kariniotakis, M. Matos, and V. Miranda, "Assessment of the benefi ts from advanced load and wind power forecasting n autonomous power systems," *EWEC 99, Nice France*, Mar. 1999.

[15] D.K.Ranaweera, N.F. Hubele, and A.D. Papalexopoulos, "Application of Radial Basis Function neural network model for short-term load forecasting," *IEE Proceedings-Generation, Transmission, Distribution*, vol. 1, pp. 45–50, 1995.

[16] Mark J. L. Orr, "Recent advances in radial basis function networks," Tech. Rep., Institute for Adaptive and Neural Computation Division of Informatics, Edinburgh University, Scotland, 1999.

[17] A. E. Gavoyiannis, N. A. Vlassis, and N. D. Hatziargyriou, "Probabilistic neural networks for power fbw analysis," *ACAI'99, Machine Learning and Applications, Chania, Crete, GREECE*, pp. 44–51, July 1999.

[18] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society B*, vol. 39, pp. 1–38, 1977.

[19] S. Haykin, *Neural Networks*, Prentice Hall, 1999.