Progress In Electromagnetics Research, PIER 53, 167–188, 2005

KERNELS EVALUATION OF SVM-BASED ESTIMATORS FOR INVERSE SCATTERING PROBLEMS

E. Bermani, A. Boni, A. Kerhet, and A. Massa

Department of Information and Communication Technology
University of Trento
Via Sommarive, 14, 38050 Trento, Italy

Abstract—Buried object detection by means of microwave-based sensing techniques is faced in biomedical imaging, mine detection, and many other practical tasks. Whereas conventional methods used for such a problem consist in solving nonlinear integral equations, this article considers a recently proposed learning by examples approach [1] based on Support Vector Machines, the techniques that proved to be theoretically justified and effective in real world domains. The article considers the approach performance for two different kernel functions: Gaussian and polynomial. The obtained results demonstrate that using polynomial kernels along with slightly sophisticated model selection criterion allow to outperform the Gaussian kernels. Simulations have been carried out for synthetic data generated by Finite Element code and a PML technique; noisy environments are considered as well. The results obtained by means of polynomial and Gaussian kernels are presented and discussed.

1. INTRODUCTION

In contrast to forward problem, where one seeks a consequence of a cause, inverse problem requires to restore a cause for an observed consequence. In particular, the inverse scattering problem requires the determination of unknown dielectric properties of scatterers from the scattered field information. The problem's handling is impeded by its ill-posedness, that is, a small error of measured data can bring to significant errors of estimated parameters. Such a problem arises in various areas, such as biomedical imaging, geophysics, remote sensing, and non-destructive evaluation, when inner properties of a body are deduced from its exterior measurements.

The problem is normally formulated in terms of an integral equation, which is iteratively solved by means of generally nonlinear minimization techniques. High computational cost of this approach could lead to its impracticability when real-time performance is required.

However, there are circumstances when one has (sometimes restricted) amount of *a-priori* information about the problem in the form of cause-consequence pairs. Furthermore, one does not always need to recover exhaustive electromagnetic properties of an object under analysis (relative permittivity and conductivity as functions of spatial coordinates); sometimes only an estimate of some object properties (e.g. scatterers presence or absence) is required. Recovering exhaustive properties in this case seems to be redundant. In other words, *when solving a given problem, try to avoid solving a more general problem as an intermediate step* [2].

Such circumstances give opportunity to solve the problem using *learning by examples* approaches, in particular such popular techniques as Artificial Neural Networks (ANNs) and Support Vector Machines (SVMs). The advantages of the latter are: 1) one has to solve a constrained quadratic optimization problem (instead of multiextremal

minimization for ANNs), 2) SVMs are based on Statistical Leaning Theory that gives the possibility to control the model's complexity and, hence, to control its generalization ability [2].

This article considers recently proposed SVM-based approach to buried objects detection problem [1]. The approach performance is estimated and compared for two different SVM configurations: Gaussian kernel SVM and polynomial kernel SVM. The obtained results demonstrate that using polynomial kernel along with slightly sophisticated model selection criterion can deliver higher accuracy to the problem under consideration than the Gaussian kernel.

The initial data for training, model selection, and testing have been synthetically obtained by means of Finite Element code and a PML technique. Environments with a number of *signal-to-noise ratios* (SNRs) are considered.

The paper is organized as follows: Section 2 describes the geometry of the problem under consideration and presents its mathematical and statistical learning statements. Section 3 is devoted to brief introduction to the SVM regression technique. Section 4 discusses the problem of model selection. Section 5 deals with the description of simulation steps; the results are presented and discussed as well. In Section 6 conclusions from the obtained results are drawn, and future work directions are given.

2. INVERSE SCATTERING PROBLEM POSING

This Section briefly describes the geometry, the physics, and the statistical learning formulation of the problem under consideration (a two-dimensional half-space, see Fig. 1).

A homogeneous circular cylindrical scatterer with center coordinates (x_{act}, y_{act}) and radius ρ is buried into the homogeneous soil inside the square region R_C (chained line). Hence, the domain under consideration for the cylinder centers D_C is the square located inside R_C at a distance of ρ from its borders (dashed line). The coordinate origin is associated with the center of D_C.

Multiple transmitters/receivers with the coordinates (x_{tr}, y_{tr}), $tr = 1, \ldots, TR$ and (x_{rs}, y_{rs}), $rs = 1, \ldots, RS$ are located at the height h above the air-soil interface. The soil's and the scatterer's dielectric properties are given by complex constants $\tau_S = (\varepsilon_S - 1) - j\frac{\sigma_S}{2\pi f \varepsilon_0}$ and $\tau_B = (\varepsilon_B - 1) - j\frac{\sigma_B}{2\pi f \varepsilon_0}$ respectively.

The transmitter with coordinates (x_{tr}, y_{tr}) radiates monochromatic electromagnetic field with free-space wavelength λ in microwave

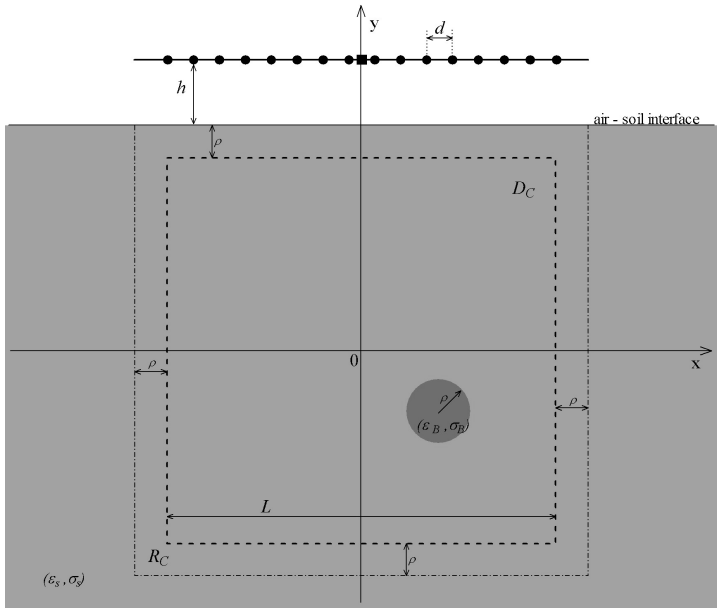

Figure 1. The geometry of the problem.

range. The electric field collected at the point (x_{rs}, y_{rs}) is

$$\begin{aligned} E^{tot}(x_{rs}, y_{rs}|x_{tr}, y_{tr}) \;=\; & E^{inc}(x_{rs}, y_{rs}|x_{tr}, y_{tr}) \\ & + k^2 \int_{R_C} E_S(x, y|x_{tr}, y_{tr}) \\ & \times G_S(x_{rs}, y_{rs}; x, y) \cdot \tau(x, y) dx\, dy, \end{aligned} \tag{1}$$

where

$$\tau(x, y) = \begin{cases} \tau_B & \text{if } \sqrt{(x - x_{act})^2 + (y - y_{act})^2} \leq \rho \\ \tau_S & \text{for the rest of } R_C\,. \end{cases} \tag{2}$$

Here $E^{inc}(x_{rs}, y_{rs}|x_{tr}, y_{tr})$ is the electric field collected at the point (x_{rs}, y_{rs}) in case of absence of the scatterer; $E_S(x, y|x_{tr}, y_{tr})$ is the electric field inside R_C in case of the scatterer's presence; $G_S(x_{rs}, y_{rs}; x, y)$ is the Sommerfeld-Green function for the half-space geometry (for details see [1, 3] and references therein).

The values of geometric and physics parameters assumed in this article are summarized in Table 1.

Inverse scattering problem in this case consists in recovering the

Table 1. Values of the parameters.

Parameter	Value	Parameter	Value
λ	0.6 m	d	$\lambda/15$
L	λ	ε_S	8.0
ρ	$\lambda/12$	σ_S	0.025 S/m
h	$\lambda/6$	ε_B	5.0
TR	1, in the centre	σ_B	0.0 S/m
RS	16, equally spaced		

location of the scatterer's center on the basis of known values of

$$\begin{array}{l} E^{tot}(x_{rs}, y_{rs}|x_{tr}, y_{tr}), \\ rs = 1, \ldots, RS \\ tr = 1, \ldots, TR. \end{array} \tag{3}$$

In terms of statistical learning, (3) forms a vector of *features (inputs)*, while horizon and depth coordinates form a vector of *outcomes (outputs)*. The learning process consists in building a prediction model on the basis of the set of available observations *(examples)*. Example means a known input-output pair, and the set of such pairs used for building a prediction model is called *training set* Γ_{train}.

Thus, for the inverse scattering problem stated above, feature vector $\boldsymbol{\chi}$ consists of $N = 2{\cdot}TR{\cdot}RS$ scalar features (this follows from (3) after taking into consideration the fact that every $E^{tot}(x_{rs}, y_{rs}|x_{tr}, y_{tr})$ consists of real and imaginary parts). The output vector $\boldsymbol{\upsilon}$ is a 2-vector: $\boldsymbol{\upsilon} = (\upsilon^x,\ \upsilon^y)$, where υ^x and υ^y denote horizon and depth coordinates respectively. Let us denote the number of examples in Γ_{train} by l. In this case

$$\Gamma_{train} = \{(\boldsymbol{\chi}_i, \boldsymbol{\upsilon}_i),\ i = 1, \ldots, l\}\,. \tag{4}$$

3. SVM REGRESSION FORMULATION

Support Vector Machines (SVMs) [2, 4, 5] are learning by examples techniques introduced by V. Vapnik. Their advantage over other approaches like ANNs is due to such relevant aspects as 1) reduction of problem to solving constrained quadratic optimization problem

(CQP) and 2) the solid Vapnik's Statistical Learning Theory basement that results in employment of Structural Risk Minimization (SRM) principle and Vapnik-Chervonenkis complexity measure [2]. Since SVMs are kernel methods, they represent input/output relation in form of linear combination of basis functions (*kernels*). This Section briefly introduces SVM regression approach.

SVM regression implies scalar outputs, therefore the inverse scattering problem stated in Section 2 has been decomposed on recovering horison and depth coordinates. This means reformulation of (4) in terms of two training sets

$$\begin{array}{l} \Gamma^{\upsilon}_{train} = \{(\boldsymbol{\chi}_i, \upsilon_i),\ i = 1, \ldots, l\}\,, \\ \upsilon \in \{\upsilon^x,\ \upsilon^y\} \end{array} \tag{5}$$

and training two independent SVMs.

Let us suppose to have a non-linear transformation $\boldsymbol{\Phi} : \mathbb{R}^N \to \mathbb{F}$, which maps the inputs $\boldsymbol{\chi}$ into a new high-dimensional space $\mathbb{F}$. SVM regression approach searches for the linear function in this new space, which reflects input/output relation in the best way:

$$\hat{\upsilon} = \boldsymbol{w} \cdot \boldsymbol{\Phi}(\boldsymbol{\chi}) + b\,. \tag{6}$$

Such an input transformation is necessary in order to better interpolate strong non-linearity of the problem under consideration.

The discrepancy between original and predicted outputs is evaluated by means of ε-*insensitive* loss function [2]

$$\left|\upsilon - \hat{\upsilon}\left(\boldsymbol{\chi}\right)\right|_{\varepsilon} = \max\left\{0,\ \left|\upsilon - \hat{\upsilon}\left(\boldsymbol{\chi}\right)\right| - \varepsilon\right\}\,. \tag{7}$$

The most intuitive way to fit the model to the available $\Gamma^{\upsilon}_{train}$ (to define optimal values of $\boldsymbol{w}$ and b, which we denote by $\boldsymbol{w}^{opt}$ and b^{opt}) is *Empirical Risk Minimization* principle (ERM):

$$\left(\boldsymbol{w}^{opt}, b^{opt}\right) = \arg\min_{\boldsymbol{w},b} \sum_{i=1}^{l} \left|\upsilon_i - \hat{\upsilon}_i\left(\boldsymbol{\chi}\right)\right|_{\varepsilon}\,. \tag{8}$$

However, this principle does not take into consideration model's complexity, which has straightforward relation to model's generalization capacity [2]. On the contrary, SVM approach consists in minimizing a trade-off (tuned by the parameter C) between the model's complexity and the error on $\Gamma^{\upsilon}_{train}$:

$$\left(\boldsymbol{w}^{opt}, b^{opt}\right) = \arg\min_{\boldsymbol{w},b} \left[\frac{1}{2}\left\|\boldsymbol{w}\right\|^2 + C\sum_{i=1}^{l} \left|\upsilon_i - \hat{\upsilon}_i\left(\boldsymbol{\chi}\right)\right|_{\varepsilon}\right]\,. \tag{9}$$

The expression (9) can be rewritten as follows:

$$(\boldsymbol{w}^{opt}, b^{opt}) = \arg \min_{\boldsymbol{w},b,\xi_i,\xi_i^*} \left[\frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*)\right]$$
$$\text{subject to } \begin{cases} \upsilon_i - \boldsymbol{w}\cdot\boldsymbol{\Phi}(\boldsymbol{\chi}_i) - b & \leq & \varepsilon + \xi_i \\ \boldsymbol{w}\cdot\boldsymbol{\Phi}(\boldsymbol{\chi}_i) + b - \upsilon_i & \leq & \varepsilon + \xi_i^* \\ \xi_i,\ \xi_i^* & \geq & 0\,. \end{cases} \tag{10}$$

The above CQP, which is also called the *Primal*, is solved by using the Lagrange multipliers theory [6] in order to obtain the corresponding *Dual*:

$$\max_{\alpha_i,\alpha_i^*} -\frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)\boldsymbol{\Phi}(\boldsymbol{\chi}_i)\cdot\boldsymbol{\Phi}(\boldsymbol{\chi}_j) - \varepsilon\sum_{i=1}^{l}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{l}\upsilon_i(\alpha_i - \alpha_i^*)$$
$$\text{subject to } \begin{cases} 0 \leq \alpha_i, \alpha_i^* \leq C \\ \sum_{i=1}^{l}(\alpha_i - \alpha_i^*) = 0\,. \end{cases} \tag{11}$$

Each of *dual variables* α_i, α_i^* is a Lagrange multiplier associated with the corresponding constraint.

Then $\boldsymbol{w}^{opt}$ is calculated as a linear combination of transformed input vectors from $\Gamma_{train}^{\upsilon}$:

$$\boldsymbol{w}^{opt} = \sum_{i=1}^{l}(\alpha_i^{opt} - \alpha_i^{*\,opt})\boldsymbol{\Phi}(\boldsymbol{\chi}_i)\,, \tag{12}$$

α_i^{opt} and $\alpha_i^{*\,opt}$ being the optimal α_i and α_i^* for (11). Thus, the dual formulation allows to write $\hat{\upsilon}$ in terms of dual variables:

$$\hat{\upsilon}(\boldsymbol{\chi}) = \sum_{i=1}^{l}(\alpha_i^{opt} - \alpha_i^{*\,opt})\boldsymbol{\Phi}(\boldsymbol{\chi}_i)\cdot\boldsymbol{\Phi}(\boldsymbol{\chi}) + b^{opt}\,. \tag{13}$$

As it follows from (11, 13), transformed input vectors appear only in the form of dot product. Thus, introducing the function

$$k(\boldsymbol{\chi}_i, \boldsymbol{\chi}_j) = \boldsymbol{\Phi}(\boldsymbol{\chi}_i)\cdot\boldsymbol{\Phi}(\boldsymbol{\chi}_j), \tag{14}$$

one avoids the explicit handling $\mathbf{\Phi}$ (so-called *kernel trick*). The theory of kernels, that is, the conditions under which equation (14) holds, is known since the beginning of the last century thanks to the Mercer's theorem [2], and has been applied to pattern recognition tasks since the '60s [7], but only recently its connection with learning machines has been well formalized [5]. Since the seminal works on kernel functions, many kernels that satisfy the Mercer's theorem have been found; we recall the linear, the Gaussian and the polynomial kernels:

$$\begin{array}{rcl} k(\boldsymbol{\chi}_i, \boldsymbol{\chi}_j) & = & \boldsymbol{\chi}_i \cdot \boldsymbol{\chi}_j \\ k(\boldsymbol{\chi}_i, \boldsymbol{\chi}_j) & = & \exp\left(-\gamma \|\boldsymbol{\chi}_i - \boldsymbol{\chi}_j\|\right) \\ k(\boldsymbol{\chi}_i, \boldsymbol{\chi}_j) & = & \left(\delta + \gamma \cdot \boldsymbol{\chi}_i \cdot \boldsymbol{\chi}_j\right)^p . \end{array} \tag{15}$$

As far as the kernel satisfies the Mercer's theorem, the CQP (11) can be efficiently solved [8, 9].

As a final remark, let us note that the parameter b can be computed by exploiting the *Karush–Khun–Tucker* (KKT) conditions [5]. In particular, according to KKT, at the solution point the product between dual variables and constraints must vanish:

$$\begin{array}{rcl} \alpha_i^{opt} \left(\varepsilon + \xi_i - \upsilon_i + \boldsymbol{w}^{opt} \cdot \mathbf{\Phi}(\boldsymbol{\chi}_i) + b^{opt}\right) & = & 0 \\ \alpha_i^{*\,opt} \left(\varepsilon + \xi_i^* + \upsilon_i - \boldsymbol{w}^{opt} \cdot \mathbf{\Phi}(\boldsymbol{\chi}_i) - b^{opt}\right) & = & 0 \,. \end{array} \tag{16}$$

This allows one to write (see [5] for details):

$$\begin{array}{l} b^{opt} = \upsilon_i - \boldsymbol{w}^{opt} \cdot \mathbf{\Phi}(\boldsymbol{\chi}_i) - \varepsilon, \quad \alpha_i^{opt} \in (0, C) \\ b^{opt} = \upsilon_i - \boldsymbol{w}^{opt} \cdot \mathbf{\Phi}(\boldsymbol{\chi}_i) + \varepsilon, \quad \alpha_i^{*\,opt} \in (0, C) \,. \end{array} \tag{17}$$

The approach described so far, the ε-based SVM for regression (ε-SVMR), is the algorithm to be used when the desired accuracy of the estimation is known a priori. However, in the case of the inverse scattering problem discussed here one needs the estimate to be as accurate as possible, without having to fix a priori a given level of accuracy. To this aim, one could use a modification of ε-SVMR, called ν-based SVM for regression (ν–SVMR) [10, 5]. The main concept of ν-SVMR can be summarized in the following way: for each $\boldsymbol{\chi}_i$ we accept an error ε; all errors above ε are stored in slack variables ξ_i or ξ_i^*, which are inserted in the global cost function and penalized by the constant C; the value of ε, in its turn, is traded-off against the model's complexity and slack variables by a parameter $\nu \geq 0$. The final Primal

CQP problem is

$$(\boldsymbol{w}^{opt}, b^{opt}) = \arg \min_{\boldsymbol{w},b,\xi_i,\xi_i^*,\varepsilon} \left[\frac{1}{2} \|\boldsymbol{w}\|^2 + C \left(l\nu\varepsilon + \sum_{i=1}^{l} (\xi_i + \xi_i^*) \right) \right]$$
$$\text{subject to} \begin{cases} \upsilon_i - \boldsymbol{w} \cdot \boldsymbol{\Phi}(\boldsymbol{\chi}_i) - b \leq \varepsilon + \xi_i \\ \boldsymbol{w} \cdot \boldsymbol{\Phi}(\boldsymbol{\chi}_i) + b - \upsilon_i \leq \varepsilon + \xi_i^* \\ \xi_i,\ \xi_i^* \geq 0 \\ \varepsilon \geq 0\,. \end{cases} \tag{18}$$

After several mathematical steps, the following dual CQP is obtained:

$$\max_{\alpha_i,\alpha_i^*} -\frac{1}{2} \sum_{i,j=1}^{l} (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) k(\boldsymbol{\chi}_i, \boldsymbol{\chi}_j) + \sum_{i=1}^{l} \upsilon_i (\alpha_i - \alpha_i^*)$$
$$\text{subject to} \begin{cases} 0 \leq \alpha_i, \alpha_i^* \leq C \\ \sum_{i=1}^{l} (\alpha_i - \alpha_i^*) = 0 \\ \sum_{i=1}^{l} (\alpha_i + \alpha_i^*) \leq Cl\nu\,. \end{cases} \tag{19}$$

Thanks to the presence of ν, ν-SVMR automatically computes ε. It has been shown that ν has several important properties; among others, the most important is that $\nu \in [0, 1]$ is an upper bound on the fraction of training points lying outside the ε-tube.

LIBSVM software [11] has been applied to implement SVM technique. ν-SVM regression based on Gaussian and polynomial kernel functions (15) have been considered.

4. PARAMETERS, HYPERPARAMETERS, AND MODEL SELECTION

One should notice that the procedure described in Section 3 finds optimal decision function (6) while values of the parameters γ, δ, p, C, ν are supposed to be already predefined (we will refer to these parameters as to *hyperparameters*). This leads to an additional problem: to find such values of hyperparameters, which would afford SVM with possibly lower *generalization error (actual risk)* [2]. To this end one minimizes the function

$$G(h_1, h_2, \ldots, h_n) \tag{20}$$

that evaluates the generalization error of SVM with hyperparameters $(h_1, h_2, \dots h_n)$, n being the number of the hyperparameters.

There is a number of approaches used for evaluating generalization error [12, 13]. The one used in this article is the so-called *validation set* approach: the generalization error for the SVM that corresponds to the hyperparameters $(h_1, h_2, \dots h_n)$ is evaluated by means of *mean square error* (MSE)

$$MSE(h_1, h_2, \dots, h_n) \tag{21}$$

reached on the *validation set* Γ^{υ}_{val}. The structure of this set is the same as the structure of $\Gamma^{\upsilon}_{train}$ (5):

$$\begin{array}{l} \Gamma^{\upsilon}_{val} = \left\{(\boldsymbol{\chi}_i, \upsilon_i),\ i = 1, \dots, l^{val}\right\}, \\ \upsilon \in \{\upsilon^x,\ \upsilon^y\}. \end{array} \tag{22}$$

However, calculating (21) in some point $(h_1, h_2, \dots, h_n)$ of hyperparameters' space means SVM training (i.e. solving CQP) and testing (on validation set), i.e. has high computational cost. Nevertheless, one can define a reasonable set of values

$$SET_i, \qquad i = 1, 2, \dots, n \tag{23}$$

for every hyperparameter and minimize (21) on cartesian product of these sets. The values of hyperparameters obtained in such a way will be referred to as *suboptimal values of hyperparameters*.

5. SIMULATION

This Section describes the simulation steps (Sections 5.2), presents the obtained results for Gaussian and polynomial kernels (Sections 5.3, 5.4 and 5.5), which are then discussed in Section 5.6.

5.1. Datasets

So far, we have already introduced training set $\Gamma^{\upsilon}_{train}$ (5) and validation set Γ^{υ}_{val} (22). The third set used in simulation is *test set* Γ^{υ}_{test}. Its aim is to provide the data for calculating the prediction error of the model found by model selection procedure (Section 4). Γ^{υ}_{test} has the same structure as $\Gamma^{\upsilon}_{train}$ and Γ^{υ}_{val}:

$$\begin{array}{l} \Gamma^{\upsilon}_{test} = \left\{(\boldsymbol{\chi}_i, \upsilon_i),\ i = 1, \dots, l^{test}\right\}, \\ \upsilon \in \{\upsilon^x,\ \upsilon^y\}. \end{array} \tag{24}$$

Noise distortion of the scattered signals received by antennas has been considered and modeled as well (additive Gaussian noise). Thus, *ultima analysi* for each of horizon and depth recovery problems 7 triplets of datasets have been generated:

$$\begin{array}{rcl} (\Gamma_{train}^{\upsilon,SNR},\ \Gamma_{val}^{\upsilon,SNR},\ \Gamma_{test}^{\upsilon,SNR}) & & \\ \upsilon & \in & \{\upsilon^x,\ \upsilon^y\} \\ SNR & \in & \{5\,\text{dB},\ 10\,\text{dB},\ 20\,\text{dB},\ 35\,\text{dB}, \\ & & 50\,\text{dB},\ 100\,\text{dB},\ \text{noiseless}\}. \end{array} \tag{25}$$

To this end, Finite Element code and a PML technique have been applied for the problem stated in Section 2.

The cylinder's center positions used to form $\Gamma_{train}^{\upsilon,SNR}$, $\Gamma_{val}^{\upsilon,SNR}$, and $\Gamma_{test}^{\upsilon,SNR}$ are indicated in Fig. 2. $\Gamma_{train}^{\upsilon,SNR}$ consists of $l = 676$ input-output pairs, whereas each of $\Gamma_{val}^{\upsilon,SNR}$ and $\Gamma_{test}^{\upsilon,SNR}$ consists of $l^{val} = l^{test} = 625$ pairs.

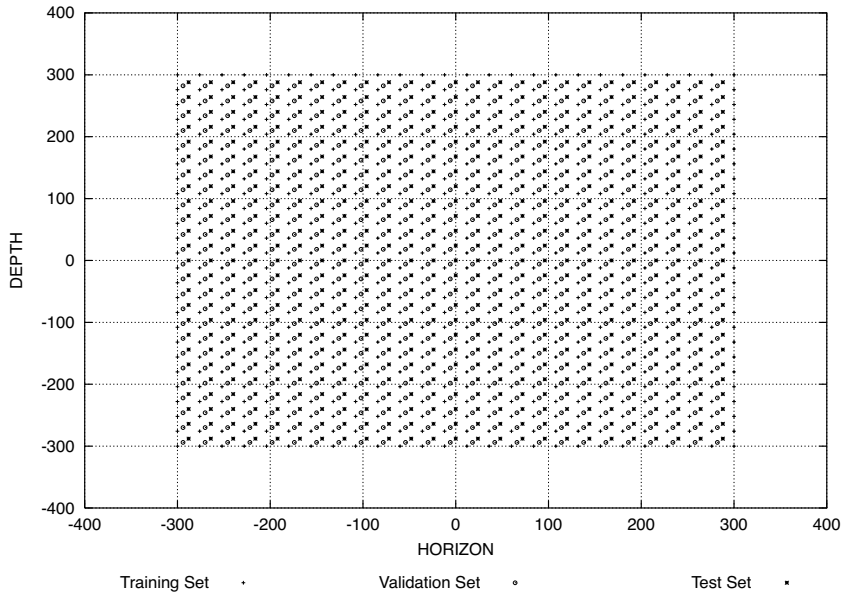

Figure 2. Training, validation and test sets' domains.

5.2. Simulation Steps

The simulation can be represented as the consequence of the following phases:

5.2.1. Normalization

Γ^{v}_{train} is linearly scaled, i.e. input vector's coordinates are projected into $[-1, +1]$ interval. The same is done with output values. The obtained scaling coefficients are then used for the normalization of Γ^{v}_{val} and Γ^{v}_{test}. This phase is useful for numerical reasons, and strongly recommended [14].

5.2.2. Searching Suboptimal Values for Hyperparameters and Testing

One determines the set of values for each hyperparameter (23). Hence, each element of cartesian product $SET_1 \times \cdots \times SET_n$ defines hyperparameters for certain SVM. Every such SVM is trained on normalized Γ^{v}_{train} and is then validated on normalized Γ^{v}_{val}. The hyperparameters of the SVM that delivers the minimal MSE on Γ^{v}_{val} define the suboptimal values of hyperparameters. This phase can be repeated iteratively by redefinition of (23), see Section 5.4.

The winner is then tested on Γ^{v}_{test}.

5.2.3. Denormalization

The predicted values are denormalized (descaled) by means of the coefficients obtained on the normalization step.

The consequence of these phases is carried out for each of 14 triplets (24).

5.2.4. Error Calculation

For each SNR value real and predicted coordinate values are used for calculating *local average error* according to the next definition [3]:

$$\begin{aligned} \zeta^{u}_{x} &= \frac{\left| x^{u}_{act} - \dfrac{1}{V(u)} \displaystyle\sum_{v(u)=1}^{V(u)} x^{v(u)}_{rec} \right|}{d_{max}} \quad u = 1, \ldots, U \\ \zeta^{v}_{y} &= \frac{\left| y^{v}_{act} - \dfrac{1}{U(v)} \displaystyle\sum_{u(v)=1}^{U(v)} y^{u(v)}_{rec} \right|}{d_{max}} \quad v = 1, \ldots, V \end{aligned} \tag{26}$$

Here u and v define respectively horizontal and vertical position on the grid formed by the cylinder's center coordinates of the test set (Fig. 2). $v(u)$ represents possible vertical positions for horizontal position

defined by u; $V(u)$ is the number of such positions. Similarly $u(v)$ represents possible horizontal positions for vertical position defined by v; $U(v)$ is the number of such positions (for the given test set $V(u) = U(v) = 25$). x^u_{act} and y^v_{act} are actual values of horizon and depth coordinates for the position on the grid defined by u and v respectively. $x^{v(u)}_{rec}$ is the recovered horizon value for the position on the grid defined by $(u, v(u))$. Similarly, $y^{u(v)}_{rec}$ is the recovered depth value for the position on the grid defined by $(u(v), v)$. $d_{max} = L_S$, see Section 2. This phase is independently done for each SNR value.

Table 2. Values of hyperparameters for Sections 5.3, 5.4, and 5.5.

HYPREPARAMETER	SET OF VALUES
Gaussian kernel	
ν	0.4 0.6 0.8
C	10^{-1} 1 10 10^2
γ	0.2 0.4 0.6 0.8 1
polynomial kernel, γ and δ are fixed	
p	1 2 3 4 5 6 7 8
ν	0.01 0.02 0.03 0.04 0.05 0.06 0.1
	0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9
C	10^{-3} 10^{-2} 10^{-1} 1 10
	10^2 10^3 10^4 10^5 10^6
δ	default (0)
γ	default $(1/k)$
polynomial kernel, δ is fixed	
p	2 3 4 5
ν	0.01 0.03 0.1 0.3 0.6 0.7
C	10^{-1} 10^2 10^3 10^5
δ	default (0)
γ	0.005 0.05 0.1

5.3. Gaussian Kernel

Sets of hyperparameter values used in case of Gaussian kernel and the obtained suboptimal values are cited in Table 2 (top part) and Table 3 respectively. The obtained local average errors (26) for horizon and depth recovery problems (5 dB and 50 dB SNR) are presented on Fig. 4.

Table 3. Suboptimal values of hyperparameters: Gaussian kernel.

SNR	Horizon SVM			Depth SVM		
[dB]	γ	ν	C	γ	ν	C
5	0.2	0.8	10^{-1}	0.2	0.4	10^{-1}
10	0.8	0.4	10	1	0.8	10^{-1}
20	0.2	0.8	10^{-1}	1	0.8	10^{2}
35	1	0.6	10^{2}	1	0.8	10^{2}
50	0.2	0.8	10^{-1}	1	0.8	10^{2}
100	1	0.6	10^{2}	1	0.8	10^{2}
noiseless	1	0.6	10^{2}	1	0.8	10^{2}

5.4. Polynomial Kernel, γ and δ Fixed by Default Values

The following approach has been considered for this simulation. At the beginning the sets (23) for ν and C have been chosen equal to ones for the case of Gaussian kernel SVM (see the top part of the Table 2), the set for p has been chosen as follows: $SET_p = \{1, 2, 3, 4, 5\}$, γ and δ have been fixed by default LIBSVM values: $\gamma = 1/N$, $\delta = 0$. On the basis of obtained MSE values graphs

$$MSE(h_i^*) = MSE(h_1^{*sub}, \ldots, h_{i-1}^{*sub}, h_i^*, h_{i+1}^{*sub}, \ldots, h_m^{*sub}), \quad i = 1, \ldots, m \tag{27}$$

have been plotted for both horizon and depth recovery cases. Here

- m is the number of hyperparameters that are not fixed;
- $h_i^* \in SET_i$;
- h_i^{*sub} is the found subopotimal value of the hyperparameter h_i.

These graphs reflect the behavior of (21); in fact they are formed by the points that belong to cross-sections of surface defined by (21). This gives the opportunity to correct SET_i: to remove some elements from

a region of h_i where MSE is large or on the contrary to add some new elements from a region of h_i where MSE has appeared to be small. Some examples of the graphs defined by (27) are presented on Fig. 3.

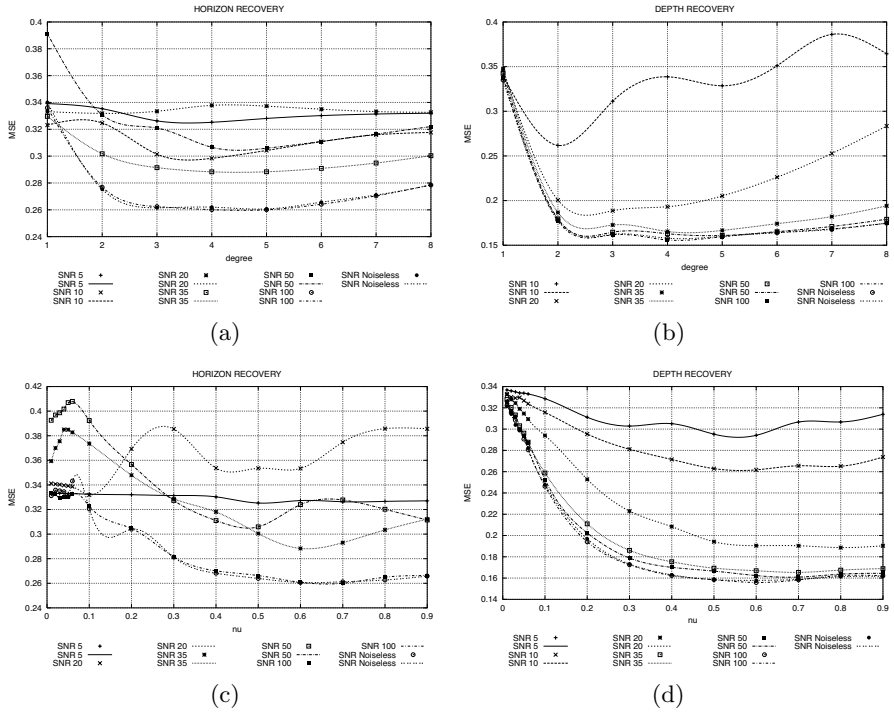

Figure 3. Mean Square Error on validation set: polynomial kernel, γ and δ are fixed.

Such a correction allows to restart searching suboptimal values of hyperparameters on the corrected sets (23). Thus, searching suboptimal values of hyperparameters have been carried out iteratively. The sets (23) have been corrected 2 times. The finally formed sets are cited in Table 2 (mid part). The overall number of different SVMs that participated in model selection is 1200. Obtained suboptimal values are cited in Table 4. Finally Fig. 4 (a-d) demonstrate the obtained local average error (26) for horizon and depth recovery problems for the cases of 5 dB and 50 dB SNR values.

This simulation has taken approximately 5 days (733 MHz Intel Pentium III CPU, 128 MBytes RAM). The major part of time (approximately 4 days) fell to SVM training phase. In its turn, majority of training phase time fell to SNR = 5 dB (approximately 2 days). It has been also noticed that within the bounds of every

Table 4. Suboptimal values of hyperparameters: polynomial kernel, γ and δ are fixed.

SNR	Horizon SVM				Depth SVM			
[dB]	p	ν	C	MSE	p	ν	C	MSE
5	4	0.5	10^{-1}	0.33	2	0.6	10^6	0.29
10	4	0.03	10^3	0.30	2	0.6	10^6	0.26
20	2	0.1	10^2	0.33	3	0.8	10^2	0.19
35	4	0.6	10^6	0.29	4	0.7	10^6	0.17
50	5	0.5	10^6	0.31	5	0.7	10^6	0.16
100	5	0.7	10^6	0.26	4	0.6	10^6	0.16
noiseless	5	0.6	10^6	0.26	4	0.6	10^6	0.16

particular SNR value the main time expenses fell to training SVMs with large values of C ($C = 10^5$, $C = 10^6$). Table 5 describes dependence of training and test phase times on values of SNR and C. $C = 10^5$, $C = 10^6$ are not cited because of requiring relatively great time expenses.

Table 5. Dependence of (horizon+depth) training and test phase times on SNR ($p = 3$, $\nu = 0.6$, $C = 1000$) and on C ($p = 3$, $\nu = 0.6$, SNR $= 5$ dB).

SNR (dB)	training time (sec)	test time (sec)	C	training time (sec)	test time (sec)
5	5.89	1.68	10^{-1}	2.24	1.65
10	3.25	1.74	1	2.37	1.79
20	3.26	1.81	10^2	2.67	1.59
35	2.69	1.82	10^3	5.89	1.68
			10^4	29.43	1.61

5.5. Polynomial Kernel, Various Values of γ

This simulation concerns removing limitation on fixing γ. To approximately estimate amount of time needed for its execution, two

horizon+depth SVM pairs ($p = 5$, $\nu = 0.7$, $C = 10^6$, $\delta = default$) have been trained on $\Gamma_{train}^{v^x,\ 100\ dB}$ and $\Gamma_{train}^{v^y,\ 100\ dB}$: with default γ (1/32) and $\gamma = 0.9$. Training has taken 17 seconds and 17 minutes respectively. Several other values of γ have just reconfirmed the trend: the greater the γ the more time expensive training phase. Finally, it has been decided to use three different γ values: 0.005 (less than default γ), 0.05 and 0.1 (greater than default γ). Sets of values for p, ν and C have been formed on the basis of the suboptimal values noted in Table 4 with some changes, see Table 2 (bottom part). Hence, the overall number of SVMs that participated in model selection is 288. $C = 10^6$ has not been considered because of high computational expenses. Suboptimal values of hyperparameters are collected in Table 6.

Table 6. Suboptimal values of hyperparameters:polynomial SVM, δ is fixed.

SNR	Horizon SVM				
[dB]	p	ν	C	γ	MSE
5	4	0.7	10^2	0.005	0.33
10	3	0.03	10^2	0.1	0.28
20	5	0.7	10^5	0.1	0.30
35	4	0.6	10^5	0.1	0.27
50	4	0.3	10^5	0.1	0.24
100	4	0.3	10^5	0.1	0.22
	Depth SVM				
5	3	0.6	10^5	0.1	0.27
10	4	0.6	10^5	0.1	0.25
20	4	0.3	10^5	0.1	0.15
35	3	0.7	10^5	0.1	0.15
50	4	0.7	10^5	0.1	0.13
100	4	0.6	10^5	0.1	0.14

Execution of training phase of this simulation has taken 7 days (733 MHz Intel Pentium III CPU, 128 MBytes RAM) plus 2 days (1700 MHz Intel Pentium IV CPU, 256 MBytes RAM). All the consequent simulation phases (model selection, testing, etc.) have taken less than 1 day (1700 MHz Intel Pentium IV CPU, 256 MBytes RAM).

Thus given simulation has consumed more time than the previous one despite the fact of decreasing the number of combinations by approximately 4 times (288 instead 1200). Fig. 4 (e–h) demonstrate obtained local average error (26) for horizon and depth recovery problems for the cases of 5 dB and 50 dB SNR values.

5.6. Discussions

Training phases for simulations described in Sections 5.4 and 5.5 have taken more than 4 and 7 days respectively. Therefore, the considered approach can not be recommended for problems where learning machine is supposed to be retrained often. On the other hand, test phase takes less than 2 seconds for $l^{test} = 625$ samples, that is less than $3.2 \cdot 10^{-3}$ seconds per sample. This means possibility of scatterer detection on the run.

Fig. 3 demonstrates that generalization performance of SVM in case of horizon recovery does not significantly depend on degree of polynomial kernel starting from $p = 2$ (Fig. 3(a)). The same relates to dependence on ν except SNR=5 dB (Fig. 3(c)), however slight trend to decreasing MSE when ν increases up to 0.5 can be traced for high SNR values. For depth recovery (Fig. 3(b) and 3(d)) trends are more traceable. Namely, optimal degree values are almost always 2, 3 or 4. Then MSE significantly decreases for high SNR values when ν increases up to 0.5.

Dependence on C is the same for both depth and horizon recovery (except 5 dB, 10 dB and 20 dB horizon recovery): MSE decreases when C increases. Probable explanation of this fact is high similarity of $\Gamma^{\upsilon}_{train}$ and Γ^{υ}_{val} (Fig. 2). Thus, for every point from Γ^{υ}_{val} close point from $\Gamma^{\upsilon}_{train}$ exists. This means that as long as SNR value is small, SVM during model selection phase is tested on almost the same data that has been used for training. Consequently, it seems difficult to evaluate generalization performance in this situation. This remains solving the ERM problem (8) obtained from (9) by assuming exactly $C = \infty$. As a conclusion, redefining of datasets in more disorderly way is expected to be reasonable.

According to the obtained results polynomial kernels have demonstrated better performance. However it is worth to notice that in case of Gaussian kernels iterative procedure of searching suboptimal values for hyperparameters has not been performed. This fact hamper in rigorous comparison of two kernels performances. Nevertheless, for low SNR values results are quite similar for both kernels.

Results obtained in Section 5.5 generally exceed the ones for Section 5.4.

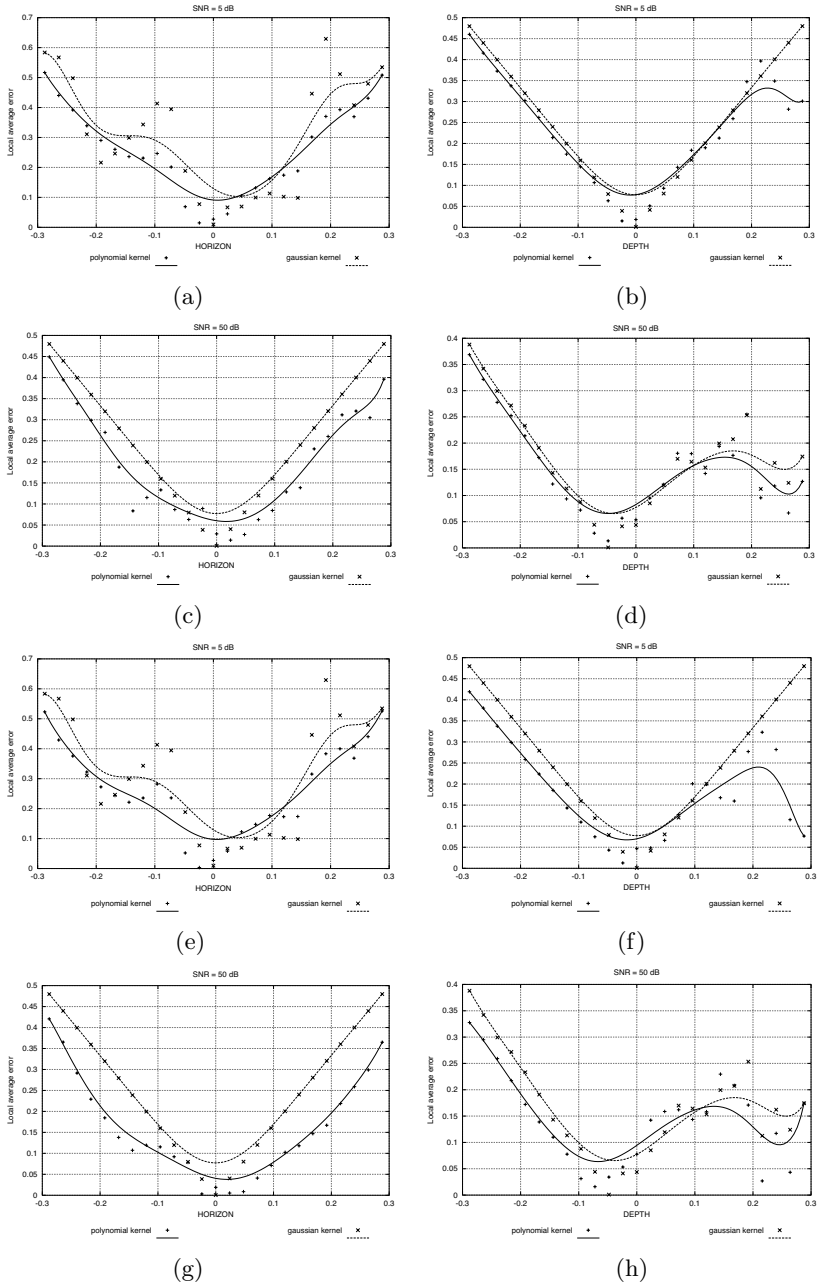

Figure 4. Local average errors: γ and δ are fixed (a–d); δ is fixed (e–h).

6. CONCLUSIONS

In this article buried object detection problem has been reformulated as regression estimation, and solved by means of *learning by examples* methodology, namely by means of ν-SVM regression technique. Simulation has been carried out on synthetic data generated by Finite Element code and a PML technique; noisy environments have been considered as well. Two different types of kernel functions have been utilized.

It has been shown that using polynomial kernel along with iterative model selection phase allows to outperform Gaussian kernel.

Though time required to SVM training can be tremendous, test phase takes less than 2 seconds for $l^{test} = 625$ samples, that is less than $3.2 \cdot 10^{-3}$ seconds per sample. This implies possibility of scatterer detection on the run.

The obtained results distinctly demonstrate the significant increase of horizon's local average error as the horizontal distance from the transmitter increases. Thus, the main direction the future work will press towards is considering the polynomial (and possibly some other) kernel performance for the model with multiple transmitters [15]. One of the other directions is model selection techniques. So far, validation set method have been used for model selection, which is reasonable when one is in data-reach situation (like synthetic data). However, this method is unsuitable on real world domain, where only limited amount of data is available. Considering other model selection methods (e.g. *k-fold cross-validation* or *maximum discrepancy* criterion [13]), which are better meet the specificity of limited amount of data, will allow to more realistically evaluate the proposed approach.

REFERENCES

1. Bermani, E., A. Boni, S. Caorsi, and A. Massa, "An innovative real-time technique for buried object detection," *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 41, No. 4, 927–931, 2003.
2. Vapnik, V. N., *The Nature of Statistical Learning Theory*, Statistics for Engineering and Information Science, 2nd edition, Springer Verlag, 1999.
3. Caorsi, S., D. Anguita, E. Bermani, A. Boni, and M. Donelli, "A comparative study of nn and svm-based electromagnetic inverse scattering approaches to on-line detection of buried objects," *ACES Journal*, Vol. 18, No. 2, 2003.

4. Cristianini, N. and J. Shawe-Taylor, *An Introduction to Support Vector Machines*, Cambridge University Press, 2000.
5. Schölkopf, B. and A. J. Smola, *Learning with Kernels*, MIT Press, Cambridge, MA, 2002.
6. Bertsekas, D. P., *Constrained Optimization and Lagrange Multipliers*, Academic Press, New York, 1982.
7. Aizerman, M. A., E. M. Braverman, and L. I. Rozonoer, "Theoretical foundations of the potential function method in pattern recognition learning," *Automation and Remote Control*, Vol. 25, 821–837, 1964.
8. Platt, J., "Fast training of support vector machines using sequential minimal optimization," *Advances in Kernel Methods — Support Vector Learning*, B. Scölkopf, C. Burges, and A. Smola (Eds.), MIT Press, 1999.
9. Lin, C.-J., "Asymptotic convergence of an SMO algorithm without any assumptions," *IEEE Trans. on Neural Networks*, Vol. 13, No. 1, 248–250, Jan. 2002.
10. Smola, A., B. Schölkopf, R. Williamson, and P. Bartlett, "New support vector algorithms," *Neural Computation*, Vol. 12, No. 5, 1207–1245, May 2000.
11. Chang, C.-C. and Ch.-J. Lin, *Libsvm: A Library for Support Vector Machines*, May 2003.
12. Hastie, T., R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning. Data Mining, Inference, and Prediction*, Springer, New York, 2001.
13. Anguita, D., S. Ridella, F. Rivieccio, and R. Zunino, "Hyperparameter design criteria for support vector classifiers," *Neurocomputing*, Vol. 55, 109–134, Sept. 2003.
14. Hsu, Ch.-W., Ch.-Ch. Chang, and Ch.-J. Lin, "A practical guide to support vector classification," Department of Computer Science and Information Engineering, National Taiwan University, July 2003.
15. Bermani, E., A. Boni, S. Caorsi, M. Donelli, and A. Massa, "A multi-source strategy based on a learning-by-examples technique for buried object detection," *PIER Journal*, Vol. 48, 185–200, 2004.

Emanuela Bermani received the laurea degree in Electronic Engineering and the Ph.D. degree from the University of Pavia, Italy in 1997 and 2000, respectively. She is currently a Research Assistant in the Department of Information and Communication Technologies. Her main interests are in electromagnetic scattering, inverse problems in electromagnetics and numerical methods in electromagnetics.

Andrea Boni was born in Genova, Italy, in 1969 and graduated in Electronic Engineering in 1996. He received a Ph.D. degree in Electronic and Computer Science in 2000. After working as research consultant at DIBE, University of Genova, he joined the Department of Information and Communication Technologies, University of Trento, Italy, where he teaches digital electronics. His main scientific interests are on the study and development of digital circuits for advanced information processing, with particular attention to programmable logic devices, digital signal theory and analysis, statistical signal processing, statistical learning theory and support vector machines. The application of such interests focuses on identification and control of non-linear systems, pattern recognition, time series forecasting, signals processing.

Aliaksei Kerhet graduated from Radio-Physics Faculty of Belarusian State University, Minsk, Belarus, in 2002. Currently he is a Ph.D. student in International Graduate School in Information and Communication Technologies, Department of Information and Communication Technology, University of Trento, Italy. His main research interests are machine learning, applications of Support Vector Machines, to buried object detection, model selection techniques.

Andrea Massa received the laurea degree in Electronic Engineering from the University of Genoa, Genoa, Italy, in 1992 and Ph.D. degree in Electronics and Computer Science from the same university in 1996. From 1997 to 1999 he was an Assistant Professor of Electromagnetic Fields at the Department of Biophysical and Electronic Engineering (University of Genoa). Since 2000, he has been an Associate Professor at the University of Trento. At present, Prof. Massa is the director of the ELEDIALab at the University of Trento. He is a member of the PIERS Technical Committee and of the Inter-University Research Center for Interactions Between Electromagnetic Fields and Biological Systems (ICEmB). His research work since 1992 has been principally on electromagnetic direct and inverse scattering, microwave imaging, optimization techniques, wave propagation in presence of nonlinear media, wireless communication and applications of electromagnetic fields to telecommunications, medicine and biology.