A Nonparametric Surrogate Model for Stochastic Crosstalk Analysis Including Confidence Bounds

A Nonparametric Surrogate Model for Stochastic Crosstalk Analysis Including Confidence Bounds

Paolo Manfredi and Riccardo Trinchero
EMC Group, Department of Electronics and Telecommunications, Politecnico di Torino
Corso Duca degli Abruzzi 24, 10129 Torino, Italy
E-mail: {paolo.manfredi,riccardo.trinchero}@polito.it

***Abstract*—This paper introduces a probabilistic nonparametric surrogate model based on Gaussian process regression to perform uncertainty quantification tasks with the inclusion of confidence bounds on the predicted statistics. The performance of the proposed method is compared against two state-of-the-art techniques, namely the parametric sparse polynomial chaos expansion and the nonparametric least-square support vector machine regression.**

***Index Terms*—Crosstalk, Gaussian process regression, machine learning, surrogate modeling, uncertainty quantification.**

I. Introduction

The increasing miniaturization in large-scale integration circuits is pushing the demand for uncertainty quantification (UQ) tools. In the signal integrity (SI) community, a great deal of attention was devoted in recent years to the generalized polynomial chaos expansion (PCE) [1]–[3] and to the related stochastic collocation methods [4], [5].

The PCE is a parametric surrogate model, meaning that the form of the model is determined a priori based on the number of uncertain parameters as well as the selected expansion order and truncation scheme [6]. The model coefficients are then computed by means of spectral methods, interpolation, or least-square regression. The PCE model is attractive because it is designed in a statistical sense: proper orthogonal basis functions are selected according to the probability distribution of the uncertain parameters, thereby favoring global statistical accuracy over local accuracy. Moreover, some statistical moments, as well as sensitivity indices, are obtained analytically from the PCE coefficients [7]. The parametric feature inherently implies that the model does not scale favorably with the number of uncertain parameters, and that a large number of samples is required to estimate the model coefficients. The latter drawback is overcome by the adoption of sparse PCE models [6], at the expense of a reduced efficiency in the calculation of the coefficients.

On the other hand, nonparametric machine learning (ML) methods exhibit a complexity that is mostly determined by the number of available "training data", thus becoming promising alternatives for problems with many independent parameters [8], [9]. Examples with applications to UQ in SI problems were reported for artificial neural networks [10] as well as for support vector machine (SVM) and least-square support vector machine (LS-SVM) regression [11]. In this scenario, ML surrogates are used for the fast prediction of random samples in a Monte Carlo (MC) analysis. The combination with principal component analysis (PCA) allows to effectively deal with multi-output problems [12].

A common feature of both the aforementioned parametric and nonparametric methods is the lack of information about the model accuracy. In this regard, an interesting alternative is provided by Gaussian process regression (GPR) [13]. Indeed, while most methods provide a deterministic model, GPR outputs a stochastic process. Hence, the model prediction is not a deterministic value, but rather a Gaussian random variable with a certain mean and standard deviation. In an UQ scenario, this information can be exploited to provide confidence bounds for the statistics of interest.

This paper discusses an application of GPR to UQ with emphasis on SI analysis. The features and performance of the proposed method are compared against two aforementioned surrogate modeling approaches, namely the parametric sparse PCE and the nonparametric LS-SVM regression.

II. Problem Definition

We consider a generic system depending on a set of d uncertain parameters $\boldsymbol{x} = (x_1, \ldots, x_d)^\mathsf{T}$, i.e.,

$$\boldsymbol{y} = \mathcal{M}(\boldsymbol{x}), \tag{1}$$

where $\mathcal{M} : \mathbb{R}^d \rightarrow \mathbb{R}^M$ is a function that maps a given configuration of the parameters $\boldsymbol{x}$ to the corresponding outputs of interest $\boldsymbol{y} = (y_1, \ldots, y_M)^\mathsf{T}$.

The goal of UQ is to calculate pertinent statistics of $\boldsymbol{y}$, such as moments and probability distributions. In a MC analysis, one would draw some random realizations of the uncertain parameters $\{\boldsymbol{x}_i^*\}_{i=1}^N$, use (1) to calculate the corresponding outputs $\{\boldsymbol{y}_i^*\}_{i=1}^N$, with $\boldsymbol{y}_i^* = \mathcal{M}(\boldsymbol{x}_i^*)$, and use standard numerical approaches to calculate pertinent statistics. Owing to the slow convergence of the estimates, this approach becomes inefficient when (1) is expensive to compute. For this reason, more effective surrogate modeling techniques were proposed for UQ.

III. State-of-the-Art Surrogate Models

In this section, we briefly review two specific and effective implementations of PCE and SVM methods, namely the sparse PCE and the LS-SVM regression, and we highlight their main features. We refer to [12] for further details. For the ease of notation, we consider a system (1) with scalar output. We also

assume that a set of observations $\{(\boldsymbol{x}_l^\dagger, y_l^\dagger)\}_{l=1}^L$ be available for training the surrogate models, with $y_l^\dagger = \mathcal{M}(\boldsymbol{x}_l^\dagger)$, $\forall l = 1, \ldots, L$.

A. Sparse PCE

A PCE surrogate takes the form

$$y \approx \hat{\mathcal{M}}_{\mathrm{PCE}}(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}} c_{\boldsymbol{\alpha}} \varphi_{\boldsymbol{\alpha}}(\boldsymbol{x}), \tag{2}$$

where $\varphi_{\boldsymbol{\alpha}}$ are orthogonal polynomials, $\boldsymbol{\alpha}$ are multi-indices indicating the degree in each dimension, and the coefficients $c_{\boldsymbol{\alpha}}$ are typically computed by means of spectral projection, least-square regression, or interpolation. An important feature of PCE surrogates is that they are specifically tailored for UQ by taking into account the distribution of the input uncertain parameters, both in the definition of the basis functions and in the calculation of the model coefficients.

In standard implementations, the size of set $\mathcal{A}$ grows exponentially with d, thus demanding for a very large number of samples for the training, thereby making it prohibitive. This drawback is mitigated by sparse PCEs, in which a subset of non-negligible coefficients is adaptively identified with a limited amount of data [6]. Nevertheless, the starting point is always the full-blown PCE, which makes this approach still parametric.

B. LS-SVM Regression

The LS-SVM model takes the form

$$y \approx \hat{\mathcal{M}}_{\mathrm{LS-SVM}}(\boldsymbol{x}) = \sum_{l=1}^{L} a_l K(\boldsymbol{x}, \boldsymbol{x}_l^\dagger; \boldsymbol{\theta}) + b, \tag{3}$$

where $K(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{\theta})$ is a kernel function. The model coefficients a_l and b, as well as the kernel hyperparameters $\boldsymbol{\theta}$, are fitted in the process of model training.

As opposed to the PCE surrogate (2), the complexity of the LS-SVM model (3) is determined by the number of training samples L. Moreover, the LS-SVM surrogate is not specifically designed for UQ, but it is rather used for the fast prediction of samples in a MC analysis [11].

C. Multiple Outputs

Very often the output of system (1) is not scalar. An example is when the UQ is to be performed on an entire transient or frequency-domain response of a circuit, possibly at multiple ports. The sparse PCE and LS-SVM methods can be individually applied to each output component. However, this approach becomes unfeasible for very-large size outputs. A more effective strategy is to compress output variables using PCA, and train an individual model for the principal components only [12]. Since the number of principal components is typically from two to three orders of magnitude smaller than the original output components, this approach turns out to be much more efficient. A model for the original output is then recovered by inverse transformation.

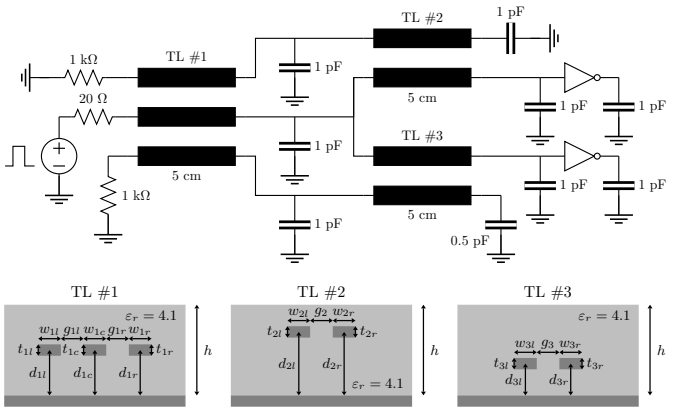

Fig. 1. Network for the application example, reproduced from [14].

IV. Proposed Probabilistic GPR Model

The GPR returns a stochastic process rather than a deterministic function, i.e.,

$$y \approx \hat{\mathcal{M}}_{\mathrm{GPR}}(\boldsymbol{x}) \sim \mathcal{GP}(m(\boldsymbol{x}), c(\boldsymbol{x}, \boldsymbol{x}')) \tag{4}$$

where $m(\boldsymbol{x})$ and $c(\boldsymbol{x}, \boldsymbol{x}')$ denote the mean and covariance functions, respectively. Hence, the model prediction for a given value $\boldsymbol{x}^*$ is Gaussian random variable with mean $m(\boldsymbol{x}^*)$ and standard deviation $\sqrt{c(\boldsymbol{x}^*, \boldsymbol{x}^*)}$. The latter provides an indication of the model uncertainty. The process mean and covariance are found by choosing a prior Gaussian process with a given kernel function $k(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{\theta})$ and conditioning it to interpolate the training samples [13].

When sampling the model (4) in a MC scenario using a finite number of samples $\{\boldsymbol{x}_i^*\}_{i=1}^N$, the result is a set of correlated Gaussian random variables with mean

$$\boldsymbol{\mu}_y = \boldsymbol{K}_* \boldsymbol{K}^{-1} \boldsymbol{y}^\dagger \tag{5}$$

and covariance matrix

$$\boldsymbol{\Sigma}_y = \boldsymbol{K}_{**} - \boldsymbol{K}_* \boldsymbol{K}^{-1} \boldsymbol{K}_*^{\mathsf{T}}, \tag{6}$$

where $\boldsymbol{y}^\dagger = (y_1^\dagger, \ldots, y_L^\dagger)^{\mathsf{T}}$ is a vector collecting the observations, whereas $\boldsymbol{K}$, $\boldsymbol{K}_*$, and $\boldsymbol{K}_{**}$ are matrices with entries

$$[\boldsymbol{K}]_{lm} = k(\boldsymbol{x}_l^\dagger, \boldsymbol{x}_m^\dagger) \tag{7a}$$

$$[\boldsymbol{K}_*]_{il} = k(\boldsymbol{x}_i^*, \boldsymbol{x}_l^\dagger) \tag{7b}$$

$$[\boldsymbol{K}_{**}]_{ij} = k(\boldsymbol{x}_i^*, \boldsymbol{x}_j^*) \tag{7c}$$

respectively, with $l, m = 1, \ldots, L$ and $i, j = 1, \ldots, N$.

A random draw from $\mathcal{N}(\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y)$ provides a realization of the MC samples. By considering an ensemble of predictions, statistical information can be estimated with the inclusion of confidence bounds. Also in this case, the method can be combined with PCA to deal with multiple-output systems.

V. Numerical Results

We apply the discussed methods to the interconnect of Fig. 1, which is taken from [14]. The quantity of interest is the transient crosstalk at the 0.5-pF termination. For a comparison on the performance in terms of computational time, we consider four different test cases with an increasing

number of uncertain parameters, as summarized in Table I. The first test case has $d = 11$ uncertain parameters, namely all the trace widths and gaps of the transmission lines. The second test case adds eight uncertain parameters, i.e., the vertical positions of the traces as well as the substrate thickness. The third test case further includes the trace thicknesses. Finally, the network capacitors are added to the forth test case, which has therefore a total of $d = 35$ uncertain parameters.

The original model (1) is in this case represented by HSPICE simulations of the network of Fig. 1. For the MC analysis, $N = 1000$ samples are considered, drawn according to a Latin hypercube sampling scheme. This step takes roughly 20 min for each test case. As a rule of thumb, $L = 10 \cdot d$ samples, distributed according to a Sobol's low-discrepancy sequence, are used for training the surrogate models. For a fair comparison, the surrogate models are then used to predict the same samples as considered in the reference MC analysis.

TABLE I
DESCRIPTION OF THE TEST CASES.

Test case	d	(Additional) uncertain parameters	Nominal value
#1	11	w_{1l}, w_{1c}, w_{1r}	150 μm
		w_{2l}, w_{2r}	130 μm
		w_{3l}, w_{3r}	170 μm
		g_{1l}, g_{1r}, g_2, g_3	150 μm
#2	19	d_{1l}, d_{1c}, d_{1r}	100 μm
		d_{2l}, d_{2r}	140 μm
		d_{3l}, d_{3r}	70 μm
		h	200 μm
#3	26	t_{1l}, t_{1c}, t_{1r}	30 μm
		t_{2l}, t_{2r}	20 μm
		t_{3l}, t_{3r}	40 μm
#4	35	C_{b1}, C_{b2}, C_{b3} C_{c1}, C_{c2}, C_{c3}	1 pF
		C_{c4}	0.5 pF

To calculate the sparse PCE models, we use the UQLab toolbox [15]. We select an expansion with total-degree truncation and a maximum order of four, and the least-angle regression algorithm for the calculation of the coefficients. For the LS-SVM model, we use the LS-SVMlab toolbox [16] with a radial basis function kernel. Finally, to train the GPR model, we make use of the pertinent routine available in the MATLAB® Statistics and Machine Learning Toolbox™ [17] with a squared exponential kernel. All simulations are performed on a Lenovo ThinkPad X1 Yoga laptop with an Intel(R) Core(TM) i7-7500U processor, CPU running at 2.7 GHz, and 16 GB of RAM.

Figure 2 shows the results pertaining to the transient crosstalk for Test Case #4. The solid gray lines in the top panel are a subset of the MC responses. The solid blue line is the mean of the MC samples, whereas the dashed green line is the GPR estimate of the mean. Confidence bounds are also plotted, yet they are too tight to be distinguished, denoting a very high accuracy of the estimate. The bottom panels refer to the variance instead, plotted on a reduced time window. The

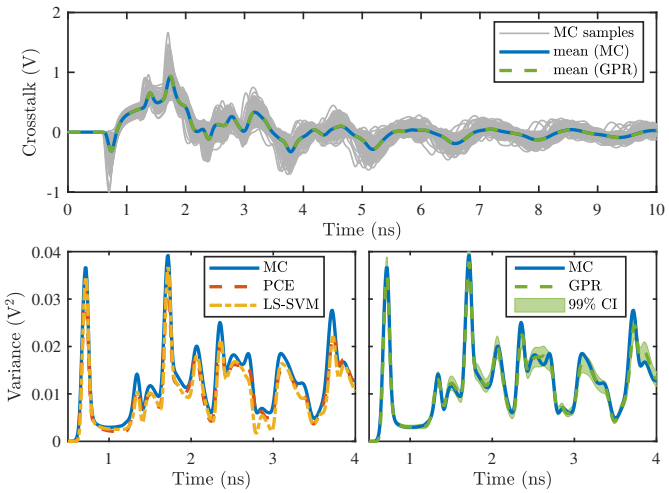

Fig. 2. Stochastic transient crosstalk for Test Case #4. Top panel: subset of MC samples (solid gray lines), mean of the MC samples (solid blue line), GPR estimate of the mean (dashed green line). Bottom panels: crosstalk variance computed from the MC samples (solid blue lines) and with the PCE, LS-SVM, and GPR surrogate models (dashed red, dash-dotted yellow, and dashed green lines, respectively). The shaded green area is the 99% CI of the GPR estimate.

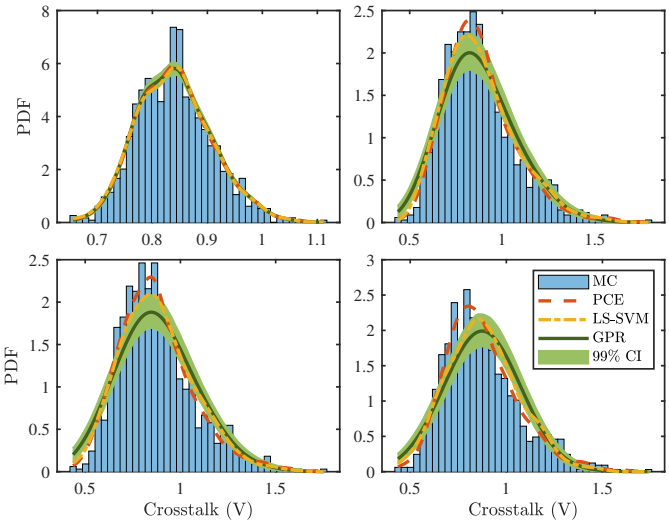

Fig. 3. PDF of the crosstalk at 1.71 ns for the four test cases. Blue histograms: reference MC distribution; dashed red, dash-dotted yellow, and solid green lines: distributions obtained from the sparse PCE, LS-SVM, and GPR models, respectively; shaded green area: 99% CI of the GPR distribution.

result of the MC analysis (solid blue line) is compared in the left panel against the variance obtained with the sparse PCE and LS-SVM surrogates (dashed red and dash-dotted yellow lines, respectively), and in the right panel against the estimate of the GPR model (dashed green line). The shaded green area indicates the 99% confidence interval (CI) of the estimate. It is observed that the MC variance is mostly enclosed by the CI. The results provided by the three surrogate models are comparable, yet the GPR model is more informative. Indeed, the information on the CI bridges the discrepancy that is observed between the prediction and the reference.

Furthermore, Fig. 3 shows, for all the four test cases, the probability density function (PDF) at 1.71 ns, i.e., the time around which the maximum crosstalk occurs. The distribution

TABLE II
PERFORMANCE OF SURROGATE MODELS.

	Test Case #1			Test Case #2			Test Case #3			Test Case #4		
Training samples	128.1 s ($L = 110$)			267.3 s ($L = 190$)			366.6 s ($L = 260$)			469.3 s ($L = 350$)		
# principal components	13			35			38			43		
Step \ Method	PCE	LS-SVM	GPR	PCE	LS-SVM	GPR	PCE	LS-SVM	GPR	PCE	LS-SVM	GPR
Model training	7.4 s	14.4 s	5.3 s	37.1 s	67.3 s	4.7 s	78.0 s	71.5 s	7.5 s	336.9 s	136.6 s	7.8 s
Model evaluation	0.1 s	0.2 s	163.1 s	0.3 s	0.4 s	370.7 s	0.2 s	0.3 s	276.7 s	0.3 s	0.5 s	250.1 s
Total	7.5 s	14.6 s	168.4 s	37.4 s	67.7 s	375.4 s	78.2 s	71.8 s	284.2 s	337.2 s	137.1 s	257.9 s

of the MC samples is shown by the blue histograms. The predictions obtained from the PCE model (dashed red lines) are closer to the MC results. This can be explained by the fact that the PCE is specifically designed in statistical terms. The LS-SVM and GPR surrogates (dash-dotted yellow and solid green lines, respectively) provide comparable results that slightly differ from the reference MC distributions. The LS-SVM result is enclosed by the 99% CI of the GPR estimate (shaded green area).

Finally, Table II provides the main figures concerning the efficiency of the surrogate models. The second row reports the number of principal components, which corresponds to the number of individual surrogate models to be trained. It should be noted that the original training data consist of L responses evaluated at 1001 time points. It is interesting to note that the time required to train the GPR model is more or less constant with the number of uncertain parameters, and it is by far the lowest. The training cost of the LS-SVM and sparse PCE models exhibits a linear and exponential growth, respectively. On the other hand, the model evaluation is negligible for both the sparse PCE (which features an analytical calculation of the mean and the variance) and the LS-SVM, whereas it is relatively high for the GPR model. This is due to the additional overhead required by the manipulation of the posterior covariance matrices in the calculation of the prediction confidence, and it is mainly determined by the number N of random samples considered. The inversion of the prior covariance matrix in (6) is instead negligible because of the limited number of training samples that is used. If only the mean prediction (5) is of interest, the evaluation time becomes comparable to the other two techniques. Owing to its better scaling with the number of uncertain parameters, the GPR method becomes more efficient for large d.

VI. CONCLUSIONS

This paper introduced a probabilistic nonparametric model, based on GPR, for UQ with the inclusion of confidence bounds. The performance of the proposed method was compared against two state-of-the-art techniques, namely the parametric sparse PCE and the nonparametric LS-SVM regression, in the simulation of crosstalk in an interconnect network. It was found that the GPR exhibits the lowest training cost and scales better with the number of uncertain parameters. The accuracy is comparable with the LS-SVM and slightly worse than the PCE method. However, this loss is compensated by the additional information on the confidence bounds.

REFERENCES

[1] J. B. Preibisch, T. Reuschel, K. Scharff, and C. Schuster, "Impact of continuous time linear equalizer variability on eye opening of high-speed links," in *Proc. 2016 IEEE 20th Workshop on Signal and Power Integrity (SPI)*, Turin, Italy, May 2016, pp. 1-4.

[2] P. Manfredi and R. Trinchero, "A hierarchical approach to the stochastic analysis of transmission lines via polynomial chaos," in *Proc. 2019 IEEE 23rd Workshop on Signal and Power Integrity (SPI)*, Chambéry, France, May 2019, pp. 1-4.

[3] S. Guglani and S. Roy, "Development of improved predictor for expedited training of polynomial chaos metamodels of multi-walled carbon nanotube interconnects," in *Proc. 2020 IEEE 24th Workshop on Signal and Power Integrity (SPI)*, Cologne, France, May 2020, pp. 1-4.

[4] X. Chen, X. Ma, A. Rong, J. E. Schutt-Ainé, and A. C. Cangellaris, "Monte Carlo estimation of sparse grid interpolation residual for stochastic collocation of high-order systems with uncertainties," in *Proc. 2017 IEEE 21st Workshop on Signal and Power Integrity (SPI)*, Baveno, Italy, May 2017, pp. 1-4.

[5] Y. Tao, K. Guo, F. Ferranti, B. Nouri, M. Nakhla, and R. Achar, "Time-domain variability analysis of large circuits with stochastic linear terminations," in *Proc. 2017 IEEE 21st Workshop on Signal and Power Integrity (SPI)*, Baveno, Italy, May 2017, pp. 1-4.

[6] G. Blatman and B. Sudret, "Adaptive sparse polynomial chaos expansion based on least angle regression," *J. Comput. Phys.*, vol. 230, no. 6, pp. 2345–2367, Mar. 2011.

[7] B. Sudret, "Global sensitivity analysis using polynomial chaos expansions," *Rel. Eng. Syst. Saf.*, vol. 93, no. 7, pp. 964—979, Jul. 2008.

[8] R. Trinchero, P. Manfredi, I. S. Stievano, and F. G. Canavero, "Machine learning for the performance assessment of high-speed links," *IEEE Trans. Electromagn. Compat.*, vol. 60, no. 6, pp. 1627–1634, Dec. 2018.

[9] M. Swaminathan, H. M. Torun, H. Yu, J. A. Hejase, and W. D. Becker, "Demystifying machine learning for signal and power integrity problems in packaging," *IEEE Trans. Compon. Packag. Manuf. Technol.*, vol. 10, no. 8, pp. 1276–1295, Aug. 2020.

[10] X. Ma, M. Raginsky, and A. C. Cangellaris, "A machine learning methodology for inferring network S-parameters in the presence of variability," in *Proc. 2018 IEEE 22nd Workshop on Signal and Power Integrity (SPI)*, Brest, France, May 2018, pp. 1-4.

[11] R. Trinchero, M. Larbi, H. M. Torun, F. G. Canavero, and M. Swaminathan, "Machine learning and uncertainty quantification for surrogate models of integrated devices with a large number of parameters," *IEEE Access*, vol. 7, pp. 4056–4066, 2019.

[12] P. Manfredi and R. Trinchero, "A data compression strategy for the efficient uncertainty quantification of time-domain circuit responses," *IEEE Access*, vol. 8, pp. 92019–92027, 2020.

[13] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2006.

[14] M. Ahadi and S. Roy, "Sparse linear regression (SPLINER) approach for efficient multidimensional uncertainty quantification of high-speed circuits," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst*, vol. 35, no. 10, pp. 1640–1652, Oct. 2016.

[15] S. Marelli and B. Sudret, "UQLab: A framework for uncertainty quantification in Matlab," in *Proc. 2nd Int. Conf. on Vulnerability, Risk Analysis and Management*, Liverpool, United Kingdom, Apr. 2014, pp. 2554–2563.

[16] LS-SVMlab, version 1.8; Department of Electrical Engineering (ESAT), Katholieke Universiteit Leuven: Leuven, Belgium, 2011. Available online: http://www.esat.kuleuven.be/sista/lssvmlab/

[17] *Statistics and Machine Learning Toolbox*, version 11.7, The MathWorks, Inc., Natick, MA, USA.