IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 29, 2020, accepted November 10, 2020, date of publication November 16, 2020,
date of current version November 27, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3038314

An Effective Manifold Learning Approach to Parametrize Data for Generative Modeling of Biosignals

LORENZO MANONI, CLAUDIO TURCHETTI, (Life Member, IEEE), AND LAURA FALASCHETTI, (Member, IEEE)
Dipartimento di Ingegneria dell'Informazione (DII), Università Politecnica delle Marche, 60131 Ancona, Italy
Corresponding author: Claudio Turchetti (c.turchetti@univpm.it)

This work was supported by the Università Politecnica delle Marche.

ABSTRACT Modeling data generated by physiological systems is a crucial step in many problems such as classification, signal reconstruction and data augmentation. However finding appropriate models from high-dimensional data sampled from biosignals is in general unpracticable due to the problem known as the ''curse of dimensionality''. Dimensionality reduction, that is representing data in some lower-dimensional space, is the commonly adopted technique to handle these data. In this context *manifold learning* has drawn great interests as a promising nonlinear dimensionality reduction method. Neverthless the main drawback of methods based on manifold learning is that they learn data implicitly, that is with no explicit model of data belonging to the manifold. The aim of this article is to develop a manifold learning approach to parametrize data for generative modeling of biosignals, by deriving an explicit function that represents the local parametrization of the manifold. The approach involves two main stages, *i)* estimation of the intrinsic dimension of data, that is the dimension of the manifold, and *ii)* estimation of the function representing the local parametrization of the manifold. Experimental results both on synthetic and real-world data shown the effectiveness of the presented approach. The source code of the algorithm for unsupervised learning of data is available at https://codeocean.com/capsule/6692152/tree/v3.

INDEX TERMS Biosignal generative modeling, intrinsic dimension, latent variables, manifold learning, nonlinear dynamical systems, regression.

I. INTRODUCTION

High dimensional data generated by physiological systems are common in many application fields, in which observations are signals achieved in the form of time series. Biological signals such as ECG, EEG and speech signals are well known examples of signals that generate data of high dimensionality.

In many related problems of classification, signal reconstruction and so on, the goal is to learn a model that adequately describes the behaviour of the system that generates the observed data [1]. Data augmentation, that synthetically increase the amount of training data to increase learning accuracy is another application that requires accurate signal generative models [2].

The associate editor coordinating the review of this manuscript and approving it for publication was Sara Dadras.

A common approach to derive biosignal generative model is the so called mathematical-based approach which rely on physical insights into the specific problem for choosing the appropriate parametric form [3], [4]. Experimental evidence has proven that physiological systems that generates biological signals belong to the wide class of nonlinear dynamical systems (NLDS) [5], that are described by nonlinear, time-dependent equations. However, since the dynamics of the systems that generate this kind of signals are too complex or unknown, just in a few cases they can be described by analytical equations.

An effective method to face this problem is to use unsupervised learning techniques in order to learn a model from unlabeled data. In this case one has a set of N observations and the goal is to directly derived a model of data without the help of a supervisor or teacher providing a degree-of-error for each observation [6]. Nevertheless analysing real world

signals and finding appropriate models from high-dimensional data is in general unpracticable because many data analysis techniques fail due to the problem known as the *"curse of dimensionality"*. As these techniques perform well for low dimensional data, understanding the potential intrinsic low-dimensional structures of high-dimensional data is an essential preprocessing step in many data analysis processes. In general to handle those data in a proper way a usually adopted approach is to represent data in some lower-dimensional space. Linear transforms, such as principal component analysis, factor analysis, linear discriminant analysis [7] and nonlinear transforms such as kernel principal component analysis [8] have been widely used for dimensionality reduction. In recent years, many new mathematical models have been proposed to capture more complex low-dimensional structures than a single subspace [9]. Dimensionality reduction methods based on local geometric structure [10] and graph learning [11] have also been widely used in the recent literature, particularly for hyperspectral imagery.

In the context of dimensionality reduction approaches, *manifold learning* (ML) has recently attracted extensive attention due its rigorous geometric interpretation, nonlinear nature and computational feasibility [12]–[15]. The main assumption in manifold learning is that the observable high dimensional data are embedded in a nonlinear manifold of lower dimension. In recent years several different manifold learning approaches, i.e. locally linear embedding (LLE), isometric mapping (IM), locally multidimensional scaling (LMDS), maximum variance unfolding (MVU), local tangent space alignment (LTSA), Laplacian eigenmaps (LE), Riemannian manifold learning (RML), diffusion maps (DM) and Hessian eigenmaps (HE), have been proposed [16]–[26]. However, a main drawback of the manifold learning methods is that they learn the low-dimensional representations of the high-dimensional data implicitly. This means that no explicit mapping representing the local parametrization of the manifold can be obtained after the training process. As a consequence during testing stage the learning procedure has to be repeatedly implemented including both the training and the testing samples as inputs, making this approach unsuitable for signal generation.

In order to apply manifold learning techniques to data generated by nonlinear dynamical systems, an explicit parametric model of such systems must be derived. Said in another way a low-dimensional *manifold* $\mathcal{M}$ embedded in the high-dimensional space of data and characterized by a nonlinear map ϕ from low-dimensional parameter space to high-dimensional space has to be determined. Thus the the manifold learning or nonlinear dimensionality reduction (NLDR) problem is to recover the nonlinear mapping relationship ϕ from data to the reduced feature map.

Several machine learning-based approaches have been proposed [27] so far for signal generation, however none of these takes advantages of the manifold concept. Here we will prove that under wide general assumptions, data generated by a dynamical nonlinear system lie on a nonlinear manifold ϕ between data and some feature variables, also called *latent variables*. The dimension of such variables in low-dimensional space is called 'intrinsic dimension' (ID) of data. This measure essentially may be interpreted as the minimum number of parameters required to describe data [28]. The estimation of ID is particularly crucial in the unsupervised learning of nonlinear time series as it allows data they generate can be accurately modelled. Several methods have been suggested for ID estimation which can be classified in two main classes [29]: local methods [29]–[32] and global methods [33]–[38]. However many of these methods are empirical or not specifically suitable for ID estimation of time series data. Recently a useful result based on the geometry of local parametrization of manifolds that establishes a rigorous criterion to determine the ID of time series data, has been derived [39].

Once the latent variables have been discovered, the initial unsupervised learning problem reduces to a supervised problem. The supervised learning of nonlinear time series falls into the wider problem of nonlinear systems identification. Over the years a large variety of different approaches has been proposed in the literature to face this problem [40]. One of the most popular is the Lee-Schetzen method that identifies the Volterra kernels of nonlinear systems stimulated by random inputs with assigned statistic [41], [42]. Simplified Volterra-based models which combine a static nonlinearity and a linear dynamical system (Hammerstein-Wiener systems) have been profitably used to overcome calculation of multidimensional Volterra kernels [43]–[45]. Because of nonlinear signal processing and learning capability, artificial neural networks (ANN's) have become a powerful tool for nonlinear system identification [46], [47]. Recently machine learning techniques such as support vector machine (SVM) are progressing rapidly, and overcomes the neural networks' shortcomings, that is local minimizing and inadequacy to statistical problems [48], [49].

Machine learning techniques reduce nonlinear system identification to solving a regression problem, thus polynomials play a central role in this context due to their property of approximating a function with arbitrary accuracy. Among a great variety of polynomials Bernstein polynomials [50] have the property that the coefficients of polynomials are given by the function to be approximated evaluated at points in a fixed grid. This useful property avoid the need of an algorithm to determine the unknown coefficients, as in other techniques occur. Recently an effective machine learning technique for regression of input-output relationships based on a set of new functions named Particle-Bernstein Polynomials (PBP) has been suggested for nonlinear system identification [51], that circumvents the problem of a time-consuming learning stage.

In this article we propose an effective manifold learning approach to the generative modeling of biosignals. The main assumption of this approach is that data are generated by a NLDS to be identified.

The aim of this article is twofold:

i) to derive a parametric model of a NLDS, thus transforming the unsupervised time series identification to a supervised problem;

ii) to develop a manifold learning approach to the identification of the model derived in *i)*.

Concerning the first point, it will be shown that under wide assumptions an NLDS generates data that are parametrized by set of variables, that represent the so called latent variables. Besides, once data are filtered to discard irregular components, data can be represented by a graph of a smooth function G, thus proving that to a NLDS corresponds a manifold $\mathcal{M}$.

With reference to the second point the proposed approach follows the scheme reported in Fig. 1. Assuming data are sampled from a manifold, the first stage is addressed to determine the intrinsic dimension of data, that is the dimension of the manifold. In such a way a local parametrization ϕ of the manifold, through a partition $\left(y'', G(y'')\right)^T$ of the data vector y, that represents the graph of a function G, is derived. A crucial step in determining the local parametrization is the estimation of ID, that is the dimension of the parametrization. Although several methods have been suggested for this purpose, a very effective method based on the Jacobian of the parametrization is used in this article. To improve the accuracy an optimal version of Nadaraya-Watson derivative estimator is developed. To this end a rigorous analysis to guarantee the best accuracy for the Jacobian estimation, is derived. By partitioning data in accordance with this parametrization, the observed data can be considered as input-output values of a function G. Thus the identification of the time series reduces to the supervised learning of the function G. To face this problem an effective machine learning technique for regression based on the set of PBPs has been adopted, that does not depend on unknown parameters to be determined.

The rest of the paper is organized as follows. In Section II we summarize related work. Section III deals with the parametric model of a dynamical system. Section IV is addressed to the supervised learning of the function that represents the local parametrization of the manifold, using Particle-Bernstein Polynomials as basis functions. Experimental results are presented in Section V.

II. RELATED WORK

A. HMM

Among machine learning-based approaches for biosignal generative modeling, the Hidden Markov Model (HMM) [52] is a popular method for modeling sequential data. HMM is defined by two probability distributions: the transition probability between hidden state to the next state and the observation distribution between the observed values and hidden states. One of the main limitations with HMMs is that they require 2^N hidden states in order to model N bits of information about the past history.

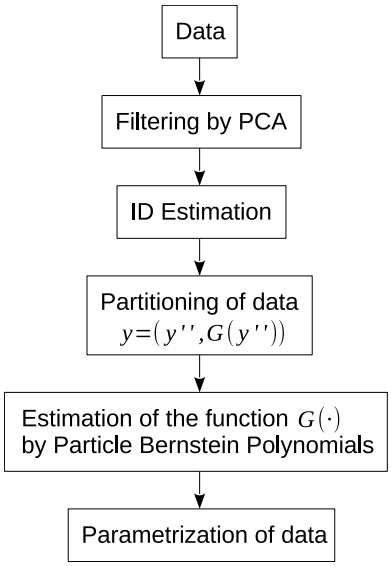


FIGURE 1. Schematic diagram of the algorithm for unsupervised learning of data generated by dynamical systems.

B. AUTOENCODER

Autoencoders have been successively applied for the reconstruction and analysis of biomedical signals [53]. An autoencoder consists of two parts. The encoder maps the input to a latent space, and the decoder maps the latent representation of the data. Unfortunately this model is not able to represent a causal model, since both the encoder and decoder are static maps, thus making autoencoders unsuitable for the generative problem.

C. RNN

A model that has been used for modeling sequential data is the Recurrent Neural Network (RNN). Generally, an RNN is obtained from a feedforward network by connecting the neuron outputs to their inputs, and modeling the short-term time-dependency by the hidden-to-hidden connections. There are two issues associated with RNN models: *i)* the number of time steps ahead has to be predetermined for most RNNs, *ii)* they fail to capture long temporal dependency for the input sequence. A popular extension to address these drawbacks, is to use a special RNN architecture named Long-Short-Term Memory Neural Network (LSTM) [54]. An LSTM neural network is composed of one input layer, one recurrent hidden layer and one output layer. The main lack of LSTM neural networks is that they are described by nonlinear composition functions depending on a large number of unknown parameters, thus requiring a time-consuming training stage.

D. GAN

Recently effective approaches based on the class of generative adversarial networks (GANs) have been proposed [55], [56]. A GAN [57] consists of two networks: the generator (G) and the discriminator (D). The generator is trained to produce data from noise samples. The discriminator is trained to distinguish training data from data generated by the generator. In the architecture proposed in [56] for biosignal generation,

both G and D are LSTM networks. More specifically G consists of a deep LSTM layer and a fully connected layer, while D consists of a deep LSTM layer, a fully connected layer, and an average pooling layer. Comparing with other techniques, this approach has shown to be more accurate, however some limitations exist. First, it requires a substantial long time to train the model that depends on 2.4×10^3 hyperparameters to be determined. Moreover, without a knowledge of the ID, the number of parameters used for training is not in general the minimum required to describe data. Second, the method is not able to vary the fundamental frequency of the generated signal, since the generator produces data from noise samples. This is a severe limitation that reduces the capability of the proposed approach to generate a large variety of signals.

III. OUR METHOD

A. MOTIVATION

Let us consider an n-dimensional random vector $z(t)$, $t = 1, \ldots, n$ satisfying the condition $E\{z(t)\} = 0$ that represents a biosignal of length n. We assume a set of data $Z = \{z_j,\ j = 1, \ldots, N\}$, i.e. the observations of the random n-vector z, can be collected from measurements.

Given data Z, our first goal is discover a low-dimensional vector $x_d \in \mathbb{R}^d$ of variables, the *latent variables*, such that data are parametrized by x_d. These latent variables x_d can be view as the hidden intrinsic state variables of the dynamical system, that mostly affect the dynamics of the system. If these variables exist with $d \ll n$, they allow data can be described in a more compact form without incurring in the *curse of dimensionality* problem. Mathematically this problem is equivalent to determine a d-dimensional manifold $\mathcal{M}$ embedded in $\mathbb{R}^n$ $(d < n)$, characterized by a nonlinear map

$$z = \alpha(x_d), \quad x_d \in U \subset \mathbb{R}^d,\ z \in \mathbb{R}^n \tag{1}$$

from low-dimensional space $U \in \mathbb{R}^d$ to high-dimensional space $\mathbb{R}^n$. In simple words, manifolds is used in mathematics to describe a parametrized surface $\mathcal{M}$ (see Fig. 2) such that to data points $\{z_1, \ldots, z_N\}$ sampled from $\mathcal{M} \subset \mathbb{R}^n$ corresponds the set $\{x_d^{(1)}, \ldots, x_d^{(N)}\}$ from $U \subset \mathbb{R}^d$ given by

$$z_i = \alpha(x_d^{(i)}), \quad i = 1, \ldots, N,\ z_i \in \mathbb{R}^n,\ x_d^{(i)} \in \mathbb{R}^d. \tag{2}$$

Here d is the so called intrinsic dimension (ID), that represents the minimum number of parameters required to describe data.

The effectiveness of manifold approach for generative modeling of biosignal is related to the dimensionality reduction of the model. Several benefits are obtained by this reduction: *i)* it is well known that high dimensionality often degrades the classification performance in pattern recognition, thus dimensionality reduction can boost the classification accuracy; *ii)* in the unsupervised learning the estimation of the probability density function $P_r(z) = P_r\,(z(1), \ldots, z(n))$ can be made simpler, as for large value of n requires a huge amount a training data; *iii)* in a low dimensionality space a substantial reduction of time required to train a model is obtained.

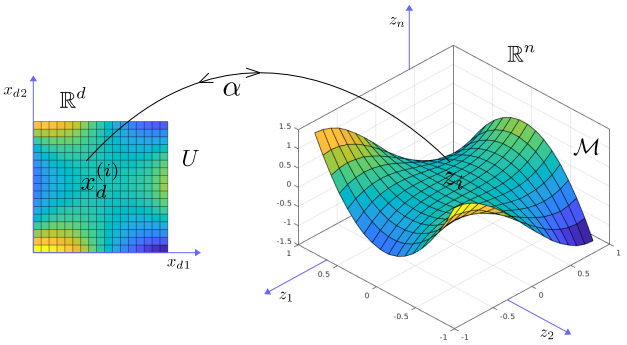


FIGURE 2. An example of parametrized surface or manifold $\mathcal{M}$.

B. NOVELTY

Three main aspects are relevant in order to capture in a manifold the low dimensional structure of data for biosignal generation: nonlinearity, explicit modeling, intrinsic dimension (ID) estimation.

Nonlinearity is essential to capture the nonlinear geometric structure of data. For biosignal generation an explicit model of the signal to be generated, as a function of some latent variables, is required. The ID of dataset has to be discovered in order to reduce the feature space and the cost of modeling computation.

Several algorithms of ML have been proposed such as LLE, IM, LMDS, MVU, LTSA, LE, RML, DM, HE. However none of these approaches for ML is able to satisfy all the aforementioned key requirements.

Our approach has been designed to address all the three fundamental aspects of ML for biosignal generation in a unified framework and represents an advancement with respect to the state-of-the-art. Indeed, to the best of our knowledge, these three aspects have never been combined in the way done in this article. The main steps of our approach are:

- We assume biosignals are caused by nonlinear dynamical systems (NLDS), that is generated by a nonlinear input-output transformation. This is a very general assumption including a large variety of biosignals.
- On the basis of the previous assumption it will be proven that the nonlinear input-output transformation can be parametrized as

$$z = \alpha(x_d) + V(x_d)\eta + \epsilon \tag{3}$$

  where x_d is the vector of latent variables, $\alpha(\cdot)$, $V(\cdot)$ are nonlinear functions and η, ϵ are noise vectors.
- The term $y = \alpha(x_d)$ is a mapping from low-dimensional space $x_d \in \mathbb{R}^d$ to the high-dimensional space $y \in \mathbb{R}^n$, thus representing a parametrized manifold $\mathcal{M}$. This proves that, once noise is discarded, biosignals lie on a manifold.
- Since x_d is hidden, i.e. not known, it will be shown that a local parametrization ϕ of the manifold, through a partition $\left(y'', G(y'')\right)^T$ of the vector y can be derived. This is the explicit nonlinear generative model depending on

latent, but known, variables $y'' \in \mathbb{R}^d$, where d =ID. The model is completely defined once d and $G(\cdot)$ have been estimated.
- To derive ID a robust technique based on the estimation of the Jacobian $J(\phi)$ of the local parametrization will be used. In particular to face this problem a Nadaraya-Watson derivative estimator approach will be adopted. To improve the accuracy an optimal version of this estimator will be developed. This result represents an improvement of the technique presented in [39], and a true advancement compared to the state-of-the-art of ID estimation.
- To estimate the function $G(\cdot)$, an effective machine learning technique for regression of input-output relationship based on a set of new functions named Particle-Bernstein-Polynomials (PBP) will be used.

The framework so derived satisfies the three main aspects, mentioned before, which are relevant to capture in a manifold the low dimensional structure of biosignals.

C. PARAMETRIC MODEL OF A DYNAMICAL SYSTEM

In many physiological systems the signal $z(t)$ can be considered as caused by a nonlinear transformation h of an input random signal $e(t)$, $t = 1, \ldots, n$, so that the following input-output relationship

$$z(t) = h\left(e(t), e(t-1), \ldots, e(1)\right), \quad t = 1, \ldots, n \tag{4}$$

holds as depicted in Fig. 3. In this scheme $e(t)$ represents the excitation (hidden or non observable), while h is the physical system that generates the random process $z(t)$ under observation. Without lack of generality we assume that to a given input $e(t)$ corresponds a unique output $z(t)$, i.e. h is one-to-one.

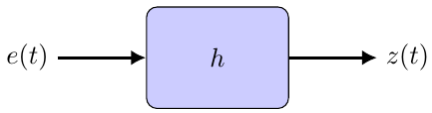


FIGURE 3. Input-output relationship.

D. LATENT VARIABLES IN DYNAMICAL SYSTEMS

One of the most general models for nonlinear transformations, that encompasses a wide class of real world dynamical nonlinear systems, is given by the following series expansion

$$\begin{aligned} z(t) = &\sum_{i=0}^{t-1} a_i\, e(t-i) + \sum_{i=0}^{t-1}\sum_{j=0}^{t-1} a_{ij}\, e(t-i)e(t-j) \\ &+ \sum_{i=0}^{t-1}\sum_{j=0}^{t-1}\sum_{k=0}^{t-1} a_{ijk}\, e(t-i)e(t-j)e(t-k) + \ldots, \\ &\qquad\qquad t = 1, \ldots, n. \end{aligned} \tag{5}$$

This series can be formally derived by assuming h in (4) is sufficiently well-behaved so that it can be expanded in Taylor series about some fixed point, the point $0 = (0, 0, 0, \ldots)$ without lack of generality. Thus, we can write

$$\begin{aligned} h(0) &= 0, \\ a_i &= \left(\frac{\partial h}{\partial e(t-i)}\right)_0, \\ a_{ij} &= \left(\frac{\partial h}{\partial e(t-i)\,\partial e(t-j)}\right)_0, \\ a_{ijk} &= \left(\frac{\partial h}{\partial e(t-i)\,\partial e(t-j)\,\partial e(t-k)}\right)_0, \quad \text{etc.} \end{aligned} \tag{6}$$

The expansion in (5) is known as Volterra series and it provides one of most general representations for nonlinear dynamic systems [5], [41], [58]–[60].

It is well known form the linear theory of stochastic processes (s.p.), that every s.p. $e(t)$ such that $E\{e(t)\} = 0$, can be represented as the sum of the two mutually orthogonal process,

$$e(t) = x(t) + \eta(t) \tag{7}$$

where $x(t)$ is a 'linear deterministic' process, that is a s.p. that is completely determined by a linear function of its past values, while $\eta(t)$ is a 'purely non deterministic' process, an s.p. that at time t is determined by a random shock or innovation which is unrelated to the shocks at other times. Say in another way, $x(t)$ is a process with memory while $\eta(t)$ has no memory since the random variable defined by $\eta(t)$ at a given t is independent of the random variables defined by $\eta(t)$ at all other t. The results is known as Wold decomposition theorem [61]–[63], for stationary processes and it was extended by Cramér and Leadbetter [61] to nonstationary processes.

The theorem can be states as follows: *Every stochastic process $e(t)$ such that $E\{e(t)\} = 0$ and $E\{|e(t)|^2\} < \infty$ for all t, can be represented as the sum (7), where $\eta(t)$ is purely nondeterministic, while $x(t)$ is linear deterministic.*

A proof of this proposition is reported in [61]. On the basis of this result, we can assume $x(t)$ is completely determined by d lagged values $x(t-1), \ldots, x(t-d)$ so that it results $x(t) = L\left(x(t-1), x(t-2), \ldots, x(t-d), t\right), t > d$ where $L(\cdot, t)$ represents a linear combination of the values $x(t-1), \ldots x(t-d)$. It is worth to notice that with the general representation (7) in mind, all the components of the signal $e(t)$ that nonlinearly depend on the past values are included in term $\eta(t)$. Highlighting the linear part is a fundamental assumption in the linear theory of random processes, nevertheless (7) is a general decomposition that is true for every process.

Substituting (7) in (5) we have

$$\begin{aligned} z(t) = &\sum_{i=0}^{t-1} a_i\, x(t-i) + \sum_{i=0}^{t-i} a_i\, \eta(t-i) \\ &+ \sum_{i,j=0}^{t-1} a_{ij}\, x(t-i)\, x(t-j) + \sum_{i,j=0}^{t-1} a_{ij}\, x(t-i)\, \eta(t-j) \\ &+ \sum_{i,j=0}^{t-1} a_{ij}\, \eta(t-i)\, \eta(t-j) + \ldots, \quad t = 1, \ldots, n. \end{aligned} \tag{8}$$

Rearranging the terms, (8) can be rewritten as a summation of three functions

$$\begin{aligned} z(t) &= h_x\left(x(t), x(t-1), \ldots, t\right) \\ &\quad + h_v\left(x(t), x(t-1), \ldots, \eta(t), \eta(t-1), \ldots, t\right) \\ &\quad + h_n\left(\eta(t), \eta(t-1), \ldots, t\right) \\ &= y(t) + v(t) + \epsilon(t), \quad t = 1, \ldots, n. \end{aligned} \tag{9}$$

The first term of (9)

$$y(t) = h_x\left(x(t), x(t-1), \ldots, t\right) \tag{10}$$

only depends on the deterministic process $x(t)$, so that it can be classified as 'nonlinear deterministic' process. The second term

$$v(t) = h_v\left(x(t), x(t-1), \ldots, \eta(t), \eta(t-1), \ldots, t\right) \tag{11}$$

is a 'non deterministic' process as it depends both on the innovations $\eta(t), \eta(t-1), \ldots$ and the process $x(t)$. The third term

$$\begin{aligned} \epsilon(t) &= h_n\left(\eta(t), \eta(t-1), \ldots, t\right) \\ &= \sum_{i=0}^{t-1} a_i\, \eta(t-1) + \sum_{i,j=0}^{t-1} a_{i,j}\, \eta(t-i)\, \eta(t-j)\, t \ldots \end{aligned} \tag{12}$$

is a 'purely non deterministic' since only depends on $\eta(t)$.

From properties of linear deterministic processes previously discussed, we can write

$$x(t) = L\left(x(t-1), x(t-2), \ldots, x(t-d), t\right), \quad t > d \tag{13}$$

where $L(\cdot, t)$ represents a linear combination of the values $x(t-1), \ldots, x(t-d)$. Since for $t = d+1$ it results $x(d+1) = L\left(x(d), \ldots, x(1), d+1\right)$ it is straightforward to show that in general we have

$$x(t) = L'\left(x(d), \ldots, x(1), t\right), \quad t > d \tag{14}$$

meaning that for $t > d$ $x(t)$ is a linear combination of initial values $x(1), \ldots, x(d)$ alone. Using this result in (10) yields

$$y(t) = h_x\left(x(t), x(t-1), \ldots, t\right) = h'_x\left(x(d), \ldots, x(1), t\right) \tag{15}$$

where

$$\begin{aligned} &h_x\left(x(t), x(t-1), \ldots, t\right) \\ &\quad = \sum_{i=0}^{t-i} a_i\, x(t-i) \\ &\qquad + \sum_{i,j=0}^{t-1} a_{i,j}\, x(t-i)\, x(t-j) \\ &\qquad + \sum_{i,j,k}^{t-1} a_{i,j,k}\, x(t-i)\, x(t-j)\, x(t-k) + \ldots \end{aligned} \tag{16}$$

and

$$\begin{aligned} y(t) = h'_x\left(x(d), \ldots, x(1), t\right) &= \sum_{i=1}^{d} \alpha_i^{(t)}\, x^{(i)} \\ &\quad + \sum_{i,j=1}^{d} a_{ij}^{(t)}\, x(i)x(j) \\ &\quad + \sum_{i,j,k=1}^{d} \alpha_{ijk}^{(t)}\, x(i)x(j)x(k) + \ldots, t > d. \end{aligned} \tag{17}$$

(17) can be rewritten as

$$\begin{aligned} y(1) &= h'_x\left(x(d), \ldots, x(1), 1\right) \\ &\vdots \\ y(n) &= h'_x\left(x(d), \ldots, x(1), n\right), \end{aligned} \tag{18}$$

thus defining the vectors $x_d = (x(1), \ldots, x(d))^T$, $y = (y(1), \ldots, y(n))^T$ and the nonlinear function $\alpha(x_d) = (\alpha(1), \ldots, \alpha(n))^T$ such that

$$\begin{aligned} \alpha(1) &= h'_x\left(x_d, 1\right) \\ &\vdots \\ \alpha(n) &= h'_x\left(x_d, n\right), \end{aligned} \tag{19}$$

we have

$$y = \alpha(x_d), \quad x_d \in \mathbb{R}^d,\ y \in \mathbb{R}^n. \tag{20}$$

On the other hand the term $v(t)$ in (11) is given by

$$\begin{aligned} v(t) &= \sum_{i,j=0}^{t-1} a_{ij}\, x(t-i)\eta(t-j) \\ &\quad + \sum_{i,j,k=0}^{t-1} a_{ijk}\, x(t-i)x(t-k)\eta(t-j) + \ldots \\ &= \sum_{j=0}^{t-1} \left(\sum_{i=0}^{t-1} a_{ij}\, x(t-i) \right. \\ &\quad \left. + \sum_{i,k=0}^{t-1} a_{ijk}\, x(t-i)x(t-k) + \ldots \right) \eta(t-j) \\ &= \sum_{j=0}^{t-1} g_j\left(x(t-1), x(t-2), \ldots, t\right) \eta(t-j) \end{aligned} \tag{21}$$

and by virtue of (14) we have

$$\begin{aligned} v(t) &= \sum_{j=0}^{t-1} g'_j\left(x(d), \ldots, x(1), t\right) n(t-j) \\ &= \sum_{j=0}^{t-1} g'_j(x_d, t) n(t-j). \end{aligned} \tag{22}$$

Then defining the vectors $\eta = (\eta(1), \ldots, \eta(n))^T$, $v = (v(1), \ldots, v(n))^T$ and the $n \times n$ matrix $V(x_d) = \{g'_j(x_d, i), i, j = 1, \ldots, n\}$, (22) can be rewritten as

$$v = V(x_d)\eta \tag{23}$$

and finally, using (20) and (23), (9) becomes

$$z = \alpha(x_d) + V(x_d)\eta + \epsilon \tag{24}$$

where $\epsilon = (\epsilon(1), \ldots, \epsilon(n))^T$, $z = (z(1), \ldots, z(n))^T$ and $\alpha(\cdot)$ is a smooth nonlinear function. The above equation corresponds to the input-output transformation given by (5), thus it is able to describe a wide class of nonlinear dynamical systems. The above equation represents a parametrization of the general input-output transformation (5), through the latent variables x_d. The model expressed by (24) is quite general since encompasses a wide variety of nonlinear dynamical systems. In (20) $\alpha(x_d)$ is a mapping $\alpha : U \to \mathbb{R}^n$ where $U \subset \mathbb{R}^d$ with $d < n$, thus it can be interpreted as a parametrized manifold $\mathcal{M}$ of dimension d embedded into the n-dimensional Euclidean space $\mathbb{R}^n$. The dimension of the vector x_d, the vector of latent variables is the intrinsic dimension (ID) of data, that is the minimum numbers of parameters required to describe data. The term $V(x_d)\eta$ is the response to the driving input, or excitation signal, η. Being η a purely non deterministic process, or a noise, this component gives rise to an irregular behaviour. This signal corresponds to an artefact or, for example in speech signal, to the unvoiced speech. Although this term can be dominant in some circumstances, the estimation of this signal is out the scope of this article as it would require advanced statistical techniques. Thus in this article we only focus on systems whose dynamics is mainly determined by the first term, thus assuming $V(x_d) = 0$. This assumption is equivalent to consider data have a smooth behaviour, so that data points are confined to a region around a regular surface, and they are modeled by

$$z = \alpha(x_d) + \epsilon \tag{25}$$

where ϵ is a noise.

The term $y = \alpha(x_d)$ in (25) is a mapping from some representation space, called the latent space, to the space of the data y, then we may express this more formally as

$$\alpha : x_d \in \mathbb{R}^d \to y \in \mathbb{R}^n \tag{26}$$

where x_d is a sample from the latent space. Thus (25) can be interpreted as generative model, meaning that once the statistical distribution of latent variables x_d is known, the model is able to capture the statistical distribution of training data Z.

Following this point of view, suppose data is a set of points $z_1, \ldots, z_N$ sampled from (21), thus we are interested in recovering the mapping $\alpha(\cdot)$ that represents the parametrized manifold $\mathcal{M}$. To this end any filtering technique can be adopted to remove the ϵ component, however principal component analysis (PCA) represents an effective technique based on geometrical considerations. In this scheme, assuming $\mathcal{B}$ is an orthonormal basis thus z can be decomposed as

$$z = \mathcal{B}k = (\mathcal{B}_y, \mathcal{B}_\epsilon)\begin{pmatrix} k_y \\ k_\epsilon \end{pmatrix} = \mathcal{B}_y k_y + \mathcal{B}_\epsilon k_\epsilon \tag{27}$$

where $\epsilon = \mathcal{B}_\epsilon k_\epsilon$ and the term $\mathcal{B}_y k_y$ corresponds to the principal components, with $k_y = \mathcal{B}_y^T z$. Comparing (21) and (27) we have

$$y = \mathcal{B}_y \mathcal{B}_y^T z \tag{28}$$

and data with noise removed are

$$Y = \{y_i,\ i = 1, \ldots N \mid y_j = \mathcal{B}_y \mathcal{B}_y^T z_i\}. \tag{29}$$

E. DATA AS GRAPH OF A FUNCTION

Following previous assumptions, as data Y are modeled by (20), thus the vector x_d completely describes data. However these variables are hidden, as they cannot be directly observed at the output of the system. Here we want to show that within the assumption previously established, a parametrized model in terms of output variables can be derived.

Given the first d time instants $t = 1, 2, \ldots, d$ we have

$$\begin{aligned} y(1) &= h'_x(x_d, 1) = y''(1) \\ &\vdots \quad\quad \vdots \quad\quad \vdots \\ y(d) &= h'_x(x_d, d) = y''(d) \end{aligned} \tag{30}$$

or in compact form

$$y'' = F(x_d) \tag{31}$$

where $y'' = \left(y''(1), \ldots, y''(d)\right)$ is a row vector. From (20) it also results

$$y = \left(y''(x_d), y'(x_d)\right)^T \tag{32}$$

being y'', y' row vectors. Having assumed the input-output relationship h is one-to-one, thus F is invertible in a generic point x of U

$$x_d = F^{-1}(y''). \tag{33}$$

Then we have

$$y = \left(y'', y'(F^{-1}(y''))\right)^T \tag{34}$$

or

$$y = \left(y'', G(y'')\right)^T \tag{35}$$

where

$$G(\cdot) = y'(F^{-1}(\cdot)). \tag{36}$$

From (35) it follows that y describes a manifold $\mathcal{M}$ or a parametric surface $\phi(y'')$ from U to $\mathbb{R}^n$ defined by the graph of $G(\cdot)$, so that (20) can be rewritten as

$$y = \phi(y'') = \left(y'', G(y'')\right)^T,\ \phi : U \to \mathbb{R}^n. \tag{37}$$

Similarly to (20), (37) is a mapping from the latent space to the space of data y

$$\phi :\ y'' \in \mathbb{R}^d \to y \in \mathbb{R}^n, \tag{38}$$

thus using (37) in (25) a generative model that depends on latent variables y'' is obtained. Since y is zero mean and

assuming, without lack of generality, the local parametrization is around the point $y_0'' = 0$ then from differentiability of ϕ around y_0'' it results

$$y = J(\phi)y''^T + R(y'') \tag{39}$$

where $J(\phi)$ is the Jacobian of ϕ and $R(y'') = o(\|y''\|)$ is the residual such that $o(\|y''\|) \to 0$ as $\|y''\| \to 0$, $\|\cdot\|$ is the norm of a vector and $o(\cdot)$ is such that $\lim_{t\to 0} o(t)/t = 0$. In (39) the first term is linear deterministic as it linearly depends on the past values y'', while the second term is nonlinear deterministic as it depends on the past values y'' but in a nonlinear manner.

The result so obtained proves that, once data are filtered to discard irregular components, data generated by the generic input-output nonlinear transformation (4) lie on the manifold $\mathcal{M}$ given by (37).

F. ID ESTIMATION

The intrinsic dimension (ID) of y is the dimension d of vector $y'' \in \mathbb{R}^d$ that, together with the function $G(\cdot)$, completely define the deterministic component y of Z. The main issue in defining such a model is to estimate the ID (d) and the nonlinear function G.

The estimation of ID is crucial for this purpose since once ID is known, a partition of y can be derived so that the unsupervised learning of y reduces to the regression of the function $G(\cdot)$ in (37). To give a more operational definition of intrinsic dimension, we proceed as follows.

Assuming ϕ is differentiable at point $y_0'' \in U$ then for every y'' around y_0'' we have

$$\phi(y'') = \phi(y_0'') + J(\phi)(y'' - y_0'')^T + o\left(\|y'' - y_0''\|\right)^T \tag{40}$$

where $J(\phi)$ is the Jacobian matrix of ϕ. It is straightforward to show that

$$J(\phi) = J\left(\phi(y'')\right) = \begin{pmatrix} I_{dd} \\ J(G^T) \end{pmatrix} \tag{41}$$

where I_{dd} is a $(d \times d)$ diagonal identity matrix, so that by virtue of condition $q \gg d$ it results

$$rank\, J(\phi) = d \tag{42}$$

meaning that the intrinsic dimension d is the dimension of the tangent subspace $J(\phi)$.

Although the rank of $J(\phi)$ univocally defines the intrinsic dimension of data, this approach is not practicable as the Jacobian matrix of ϕ can not be derived from data in general since the function ϕ and the dimension d are not known.

The following proposition gives a more practicable method to estimate d [39].

Proposition 1: Let $y \in \mathbb{R}^n$ a s.p. with ID $= d$, that is such that all the observations belong to a manifold M defined by the graph (37) of the function G. For any partition $p > d$ of y given by

$$y = \psi(u) = (u, \beta(u))^T, \; u \in \mathbb{R}^p \tag{43}$$

the singular values of Jacobian $J\psi(t)$ are such that

$$\lambda = (\lambda_1, \ldots, \lambda_d, \lambda_{d+1}, \ldots, \lambda_p) \tag{44}$$

with $\lambda_{d+1} = \ldots = \lambda_p = 1$.

In [39] a kernel approximation for $\beta(u)$ is used to derive the Jacobian analytically by differentiating (43). Although this approach has shown to be effective in the estimation of the ID, it is very sensitive to the superimposed noise. Thus in order to reduce the effect of noise a more robust technique has been developed for the estimation of Jacobian $J(\psi)$.

NONPARAMETRIC ESTIMATION OF JACOBIAN $J(\psi)$

Applying a generic partition p to data, as given by (43), the ID estimation reduces to the computation of Jacobian $J(\beta^T)$ since by differentiation (43) we have

$$J(\psi) = \begin{pmatrix} I_{pp} \\ J(\beta^T) \end{pmatrix} \tag{45}$$

where I_{pp} is a $(p \times p)$ diagonal identity matrix and $J(\beta^T)$ is the $(n - p) \times p$ Jacobian matrix of $\beta^T(u)$. Thus in this way the ID estimation is equivalent to the estimation of Jacobian $J(\beta^T)$ from data. However since $\beta(u)$ is not available in analytic form, a numerical technique has to be adopted. To face this problem a Nadaraya-Watson derivative estimator approach [64] is used.

Let us consider a noise-perturbed scalar model:

$$w_k = f(u_k) + \epsilon_k \quad k = 1 \ldots N \tag{46}$$

where $u_k \in \mathbb{R}^p$ is a random vector, $f(\cdot)$ is a scalar function and ϵ_k is a noise with a variance $\sigma^2 = E(\epsilon_k^2)$, and uncorrelated with u. With reference to a given kernel function $K(\cdot)$, the Nadaraya-Watson regression is given by:

$$\widehat{f}(u) = \frac{\widehat{h}(u)}{\widehat{g}(u)} \tag{47}$$

where $\widehat{g}(u)$ denotes the estimate of the pdf $g(u)$ of the random vector u, and $\widehat{h}(u)$ is the estimate of $h(u) = f(u)g(u)$.

Nadaraya-Watson regression is a non parametric technique that estimates the quantities $g(u)$ and $h(u)$ with the following relationships:

$$\widehat{g}(u) = \frac{1}{N}\sum_{k=1}^{N} c_k(u), \quad \widehat{h}(u) = \frac{1}{N}\sum_{k=1}^{N} \left(b_k(u) + \epsilon_k c_k(u)\right) \tag{48}$$

$$c_k(u) = \frac{1}{D_k} K\left(H_k^{-1}(u_k - u)\right) \tag{49}$$

$$b_k(u) = \frac{f(u_k)}{D_k} K\left(H_k^{-1}(u_k - u)\right) \tag{50}$$

where H_k is the kernel bandwidth.

At first we consider the general case with a different kernel bandwidth for each variable:

$$H_k = \text{diag}(h_1^k \ldots h_p^k), \quad D_k = \det(H_k) = \prod_{j=1}^{p} h_j^k. \tag{51}$$

The choice of the bandwidth H_k is crucial for the quality of Nadaraya-Watson estimator, thus in the following we will derive an optimal value for the bandwidth H_k, that guarantees the best accuracy for the Jacobian estimation.

The gradients of $\widehat{g}(u)$ and $\widehat{h}(u)$ are given by:

$$\frac{\partial \widehat{g}(u)}{\partial u} = \frac{1}{N}\sum_{k=1}^{N} d_k(u)\frac{\partial \widehat{h}(u)}{\partial u} = \frac{1}{N}\sum_{k=1}^{N}\left(a_k(u) + \epsilon_k d_k(u)\right) \tag{52}$$

where

$$d_k(u) = H_k^{-2}\frac{u_k - u}{D_k} K\left(H_k^{-1}(u_k - u)\right) \tag{53}$$

$$a_k(u) = f(u_k) H_k^{-2}\frac{u_k - u}{D_k} K\left(H_k^{-1}(u_k - u)\right). \tag{54}$$

The final derivative estimator therefore becomes:

$$\frac{\partial \widehat{f}}{\partial u} = \frac{1}{\widehat{g}(u)}\frac{\partial \widehat{h}(u)}{\partial u} - \frac{\widehat{f}(u)}{\widehat{g}(u)}\frac{\partial \widehat{g}(u)}{\partial u} \tag{55}$$

The convergence properties of Nadaraya-Watson estimator of the gradient $\frac{\partial f}{\partial u}$ with respect to H_k have been studied in [64] for the 1-dimensional case, here a generalization of such analysis to the p-dimensional case was performed.

The mean-square error $\mathcal{E}^2(u) = E(||\nabla\widehat{f} - \nabla f||_2^2)$ can be written as the sum of two main terms:

$$\mathcal{E}^2(u) = \mathcal{E}_V^2(u) + \mathcal{E}_B^2(u) \tag{56}$$

$$\mathcal{E}_V^2(u) = E\left(||\nabla\widehat{f} - E(\nabla\widehat{f})||_2^2\right) \tag{57}$$

$$\mathcal{E}_B^2(u) = ||E(\nabla\widehat{f}) - \nabla f||_2^2 \tag{58}$$

where ∇ denotes the partial derivative $\frac{\partial}{\partial u}$. Strictly speaking $\mathcal{E}_V^2(u)$ represents the variance contribution to the error while $\mathcal{E}_B^2(u)$ is the bias of the estimator. The asymptotic behaviour, i.e. as the bandwidth $H_k \to 0$, of those two terms in the p-dimensional case is given by:

$$\mathcal{E}_V^2(u) = \frac{\sigma^2\xi_2 p}{g(u)N^2}\sum_{k=1}^{N}\frac{\left\langle \mathrm{diag}\left(H_k^{-2}\right)\right\rangle}{D_k} \tag{59}$$

$$\mathcal{E}_B^2(u) = \frac{1}{N^2}\left\|\sum_{k=1}^{N} B_k(u)\right\|_2^2. \tag{60}$$

where ξ_2 is a constant, $\langle\cdot\rangle$ denotes the average of the elements of a vector, diag$(\cdot)$ denotes the diagonal of a matrix. $B_k(u)$ is a quadratic form of the bandwidth H_k, i.e. it can be expressed as follows:

$$B_k(u) = O(H_k^2 Q_k(u)) \tag{61}$$

where $Q_k(u)$ is an appropriate function which does not depend on H_k and $a = O(b)$ means that a/b is bounded. A derivation of (59) and (60) is reported in Appendix.

In order to derive relationships more intuitive than (59) and (60), we assume a constant scalar bandwidth $H_k \equiv H = \widetilde{h}I$. With this simplification the expressions for $\mathcal{E}_{V,B}^2(u)$ reduce to:

$$\mathcal{E}_V^2(u) = \frac{\sigma^2\xi_2}{Ng(u)\widetilde{h}^{p+2}} \tag{62}$$

$$\mathcal{E}_B^2(u) = \widetilde{h}^4||Q(u)||_2^2 \tag{63}$$

where the dependency on index k has been eliminated. In general is difficult to estimate directly the function $Q(u)$ since it depends on first and second-order derivatives of $f(u)$ and $g(u)$, nevertheless an analysis for the estimation of $Q(u)$ can be found in Appendix. Some useful considerations can be derived from (62) and (63) as follows.

The bias term $\mathcal{E}_B^2(u)$ does not depend on the number of regression points N and is an increasing function of the bandwidth $\widetilde{h}$. The variance term $\mathcal{E}_V^2(u)$, which is responsible for the fast fluctuations of the estimate, is proportional to the noise power σ^2 and decreases with the number of the regression points N as well as their density $g(u)$. Clearly $\mathcal{E}_V^2(u) \sim \frac{1}{h^{p+2}}$ which is high rapidly increasing function as $\widetilde{h} \to 0$. Since the two terms $\mathcal{E}_V^2$ and $\mathcal{E}_B^2$ show an opposite trend with respect to $\widetilde{h}$ thus $\mathcal{E}^2(u)$ has a global minimum which immediately leads to the optimal choice for $\widetilde{h}$:

$$h^{\mathrm{opt}}(u) = \left(\frac{(p+2)\sigma^2\xi_2}{Ng(u)||Q(u)||_2^2}\right)^{\frac{1}{p+6}} \tag{64}$$

The detailed estimation error analysis and the practical implementation of (64) can be found in Appendix.

As a simple example to validate the technique discussed in this section, we refer to the following 2-D noisy model:

$$\begin{aligned} w &= f(x_1, x_2) + \epsilon \\ f(x_1, x_2) &= \exp(-\tau x_1) + A_0\sin(\omega x_2) \end{aligned} \tag{65}$$

where $x \sim \mathcal{N}(0,1)$, $\epsilon \sim \mathcal{N}(0,\sigma^2)$, $\sigma = 0.05$, $A_0 = 1/2$, $\omega = 0.3$.

Fig. 4 compares the true and the estimated gradient achieved with the proposed technique. The estimation error $\mathcal{E}^2(u)$ is reported in Fig. 5 as a function of $\widetilde{h}$ for different points u. As you can see all the graphs have a minimum point which balances the trade-off between variance and bias error, as predicted by (62) and (63).

IV. SUPERVISED LEARNING BASED ON PARTICLE BERNSTEIN POLYNOMIALS

A. BERNSTEIN POLYNOMIALS

Assuming the ID of data set has been determined with the previous approach, the unsupervised learning of s.p. y that generates data reduces to the estimation of the input-output function $G(\cdot)$ in (35). In such a way the initial problem of unsupervised learning reduces to a supervised learning as the input y'' of the system (37) is known. In this context polynomials are useful basis functions to represent input-output relationships of the kind given by (36) [65], [66]. Such important result is founded on the well known Weierstrass

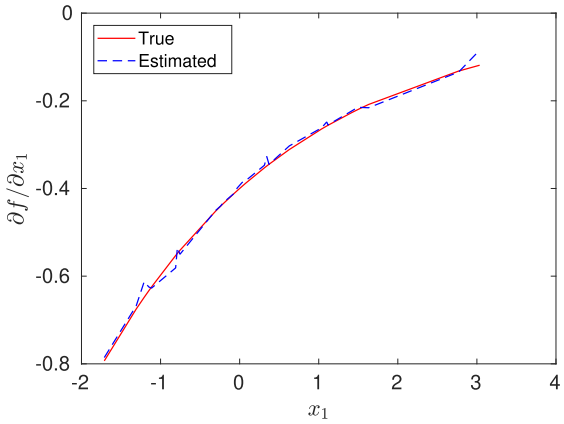


FIGURE 4. Comparison of the true and estimated gradient $\frac{\partial f}{\partial x_1}$.

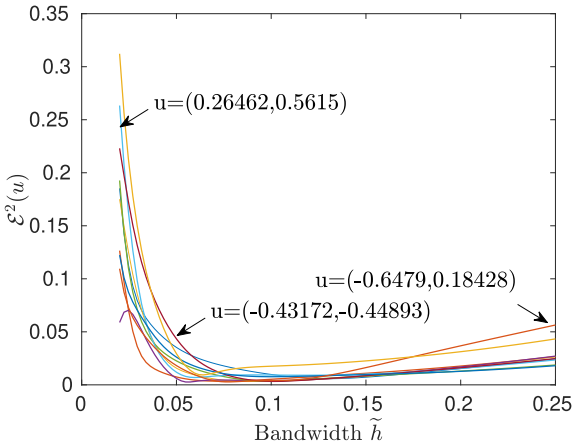


FIGURE 5. The estimation error $\mathcal{E}^2(u)$ as a function of $\tilde{h}$ for different points u.

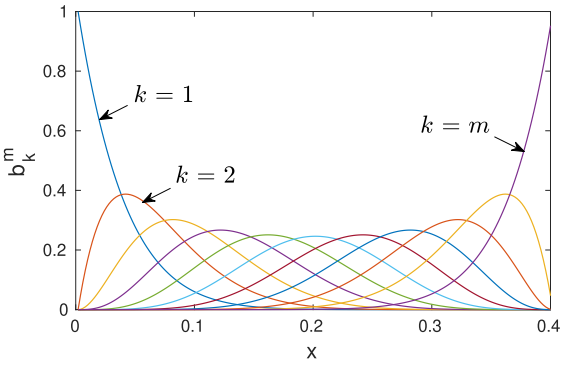


FIGURE 6. The functions $b_k^m(x)$ for $m = 20$ and $k = 1, \ldots, m$.

approximation theorem which states that every continuous function f defined in a finite interval can be approximated by a polynomial with arbitrary accuracy in the uniform norm. In the class of such basis functions, Bernstein polynomials have been widely used to solve both theoretical and application problems [50], [67].

The univariate Bernstein polynomials of degree m over the interval $[0, 1]$ are defined by:

$$b_k^m(x) = \binom{m}{k} x^k (1-x)^{m-k}, \quad k = 0, 1, \ldots, m \tag{66}$$

where $x \in [0, 1]$ and $\binom{cm}{k} = \frac{m!}{(m-k)!\,k!}$. They form a complete basis for the space of polynomials of degree less than or equal to m, as they span the space of polynomials and they are linearly independent. A complete treatment of Bernstein polynomials is reported in [50] where a number of other properties can be found. It can be easily proven that each term $x^k(1-x)^{m-k}$ is maximum at $x = k/m$, and this behaviour is shown in Fig. 6 for $m = 20$ and some values of k.

The multivariate version of Bernstein polynomials can be easily derived from (66) giving

$$b_k^m(x) = \binom{m}{k_1} \ldots \binom{m}{k_d} x_1^{k_1}(1-x_1)^{m-k_1} \ldots x_d^{k_d}(1-x_d)^{m-k_d} \tag{67}$$

where $x = (x_1, \ldots, x_d)$, $k = (k_1, \ldots, k_d)$.

One of the main properties of Bernstein polynomials is that given a continuous function $f(x)$ in $[0, 1]^d$ the sequence $B_m(x)$ defined as

$$B_m(x_1, \ldots, x_d) = \sum_{k_1=0}^{m} \ldots \sum_{k_d=0}^{m} f\left(\frac{k_1}{m}, \ldots, \frac{k_d}{m}\right) \times b_k^m(x) \tag{68}$$

converges uniformly to f as $m \to \infty$. As you can see this representation only requires the knowledge of the function at the points

$$x = (k_1/m, \ldots, k_d/m), \quad k_1 = 0, \ldots, m, \; k_d = 0, \ldots, m \tag{69}$$

without the need of an algorithm to determine the unknown coefficients, as in other techniques occurs. The set of points in (69) defines a grid in the space $\mathbb{R}^d$ of the input variable formed by $(m+1)^d$ points. The grid must be sufficiently dense to get a good approximation of f, thus requiring an increasing value of m for better accuracy. However as the dimensionality d of input space increases, the computational cost of (68) increases dramatically. In order to overcome this limitation a new class of polynomials will be derived in the following.

B. PARTICLE-BERNSTEIN POLYNOMIALS

In Bernstein polynomials both the variables m and k are integer, as the binomial coefficients are defined for integer values alone. We can remove the constraint of a fixed grid by assuming k is real and denoting this value with ξ so that a new set of functions, called particle-Bernstein polynomials (PBPs) [51], can be defined as follows

$$C_\xi^m(x) = \alpha_\xi^m x^\xi (1-x)^{m-\xi} = \alpha_\xi^m k_\xi^m(x), \quad \xi \in \mathbb{R}^1, \;\; \xi \in [0, m] \tag{70}$$

with the coefficients α_ξ^m chosen in such a way the integral constraint

$$\int_0^1 C_\xi^m(x)\,dx = 1 \tag{71}$$

holds, that is

$$\alpha_\xi^m = \frac{\Gamma(m+2)}{\Gamma(\xi+1)\,\Gamma(-\xi+m+1)} \tag{72}$$

In the multivariate case (70) becomes

$$\begin{aligned} C_\xi^m(x) &= \alpha_\xi^m\, x_1^{\xi_1}(1-x_1)^{m-\xi_1}\ldots x_d^{\xi_d}(1-x_d)^{m-\xi_d} \\ &= \alpha_\xi^m\, k_\xi^m(x) \end{aligned} \tag{73}$$

with $\xi = (\xi_1, \ldots, \xi_d)$, $x = (x_1, \ldots, x_d)$ and

$$\alpha_\xi^m = \prod_{t=1}^{d} \frac{\Gamma(m+2)}{\Gamma(\xi_t+1)\,\Gamma(-\xi_t+m+1)} \tag{74}$$

$$k_\xi^m = \prod_{t=1}^{d} x_t^{\xi_t}(1-x_t)^{m-\xi_t}. \tag{75}$$

Fig. 7 shows several examples of functions defined by (70) for $m = 20$ and different values of ξ. As you can see all the functions have the same area and attain their maximum at $x = \xi/m$. This property can be used to approximate a function $f(x)$ around a given point. In fact supposing the function $C_\xi^m(x)$ is mostly concentrated around its maximum, and this is true for $m \gg 1$, the following approximation for f

$$f\,(\xi/m) \cong \int_0^1 f(x)\, C_\xi^m(x)\,dx \tag{76}$$

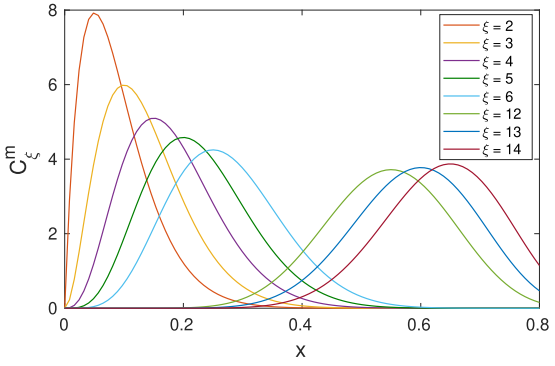


FIGURE 7. The function $C_\xi^m(x)$ for $m = 20$ and several values of ξ.

holds, where $f\,(\xi/m)$ is the value of $f(\cdot)$ at the maximum of $C_\xi^m(x)$. Thus the integral

$$f_m(\xi/m) = \int_0^1 f(x)\, C_\xi^m(x)\,dx \tag{77}$$

represents an approximating function of $f(x)$ around the point $x = \xi/m$. For a given set of values $\{x^{(j)},\ j = 1, 2, \ldots, N\}$ the integral in (77) can be approximated as

$$f_m(\xi/m) \cong \frac{1}{N}\sum_{j=1}^{N} f(x^{(j)}) C_\xi^m(x^{(j)}) \tag{78}$$

Similarly, using the same approximating concept, it results

$$\int_0^1 C_\xi^m(x)dx \cong \frac{1}{N}\sum_{j=1}^{N} C_\xi^m(x^{(j)}) \cong 1 \tag{79}$$

thus combining (76), (78) and (79) we have

$$f_m(\xi/m) \cong \frac{f_m(\xi/m)}{1} \cong \frac{\sum_{j=1}^{N} f(x^{(j)}) C_\xi^m(x^{(j)})}{\sum_{j=1}^{N} C_\xi^m(x^{(j)})}. \tag{80}$$

(80) can be used to estimate the function $f(x)$ at testing point ξ/m given the training set $\{f\left(x^{(j)}\right),\ j = 1, \ldots, N\}$. As you can see the estimate of the function f at a given point does not require the knowledge of the function in a fixed grid, as in Bernstein polynomials occurs, instead only the function at a set of N points randomly chosen is required for this purpose. In addition, once the order m is chosen, (80) does not depend on unknown parameter to be determined, thus avoiding the need for a time-consuming training stage.

V. EXPERIMENTAL RESULTS

The method to parametrize data generated by dynamical systems previously discussed, has been validated by several experiments. The experiments were conducted both on data generated by synthetic nonlinear dynamic systems (first and second experiments) and data generated by real systems (third, fourth and fifth experiments).

A. UNSUPERVISED LEARNING OF DATA GENERATED BY A FORCED NONLINEAR OSCILLATOR

In the first example data are obtained as the output values y of the following input-output nonlinear discrete system

$$y(t+1) = \frac{\Delta t^2\left[e(t) - \beta y^3(t)\right] - y(t-1) + \delta y(t)}{(1+k\Delta t)} \qquad t = 1, \ldots, n \tag{81}$$

forced by the input

$$e(t) = \gamma\, x(2)\cos\left(x(1)\cdot\Delta t(t-1)\right) \tag{82}$$

where $\delta = k\Delta t - \alpha\Delta t^2 + 2$, being Δt, k, α, β, γ constant parameters, and $x = (x(1), x(2))$ is a random vector. (81) is the discrete-time version of the well known Duffing equation with $\Delta t = 0.2$, $k = 0.3$, $\alpha = -4$, $\beta = 1$, $\gamma = 2$, and x uniformly distributed in the interval $[0, 1]^2$. The initial conditions have been chosen to be $y(0) = y(-1) = 1$. Some realizations of the stochastic nonlinear system defined by (81) are reported in Fig. 8 The experiment has been conducted with $N = 10^3$, assuming $n = 50$ and for different partitions of $y = (u, u')^T$, $u \in \mathbb{R}^p$ with $p = 1, \ldots, n-1$. Table 1 shows the singular values of Jacobian matrix $J(\psi)$ as estimated with the approach of Section III-F with p ranging from 1 to 12. As you can see for all the values of p a number $p - 2$ of singular values equal to 1 occur, thus resulting in a value of intrinsic dimension $d = 2$. As a consequence we assume each observation y is partitioned as $y = (y'', y')^T$, such that the input-output relationship (37) holds, with $y'' \in \mathbb{R}^d$. In such

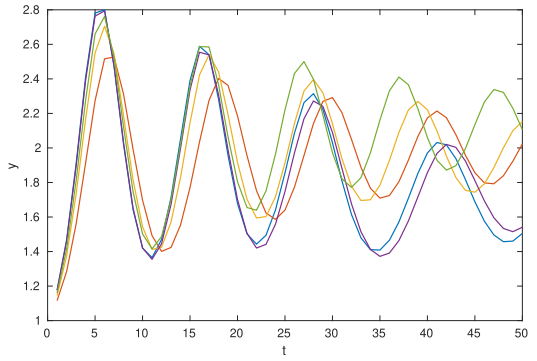


FIGURE 8. Some randomly chosen frames generated by the nonlinear system defined by (81).

TABLE 1. Singular values in experiment 1.

p	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	λ_8	λ_9	λ_{10}	λ_{11}	λ_{12}
2	61.6595	6.2928	0	0	0	0	0	0	0	0	0	0
3	13.2458	1.7985	1.0000	0	0	0	0	0	0	0	0	0
4	2.5477	1.5152	1.0000	1.0000	0	0	0	0	0	0	0	0
5	1.6344	1.1899	1.0000	1.0000	1.0000	0	0	0	0	0	0	0
6	1.5323	1.0775	1.0000	1.0000	1.0000	1.0000	0	0	0	0	0	0
7	1.5483	1.0466	1.0000	1.0000	1.0000	1.0000	1.0000	0	0	0	0	0
8	1.4843	1.0292	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0	0	0	0
9	1.3838	1.0186	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0	0	0
10	1.3204	1.0152	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0	0
11	1.3083	1.0194	1.0002	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0
12	1.3202	1.0343	1.0003	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

a way the unsupervised learning reduces to the regression of the function $G(\cdot)$ given the data y_j, $j = 1, \ldots, N$. To solve efficiently this problem the regression method based on Particle Bernstein Polynomials of Section IV-B has been applied. Fig. 9 reports as blue stars the results achieved with this approach for two different observations, and for comparison the behaviour of data as continuous lines. As you can see the learning approach is able to model data with good accuracy.

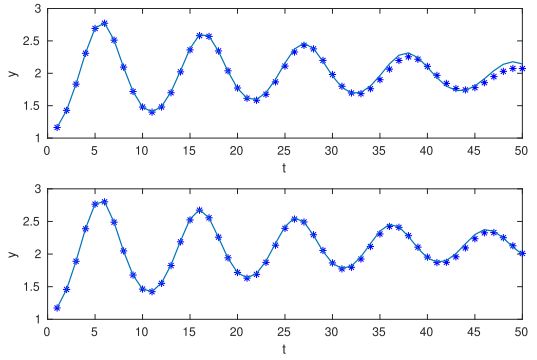


FIGURE 9. Comparison of the results achieved by the approximation (80) (blue stars) with data (continuous line) in the experiment 1.

B. UNSUPERVISED LEARNING OF DATA GENERATED BY NARENDRA's EQUATIONS

In the second experiment data were generated according to equation proposed by Narendra and Parthasarathy [46].

$$y(t) = \frac{y(t-1)y(t-2)(y(t-1)-2.5)}{1+y^2(t-1)+y^2(t-2)} + e(t-1) \qquad (83)$$

$$e(t) = ax(2)\sin(x(1)\omega t) + b\cos(\omega_1 x(3)) \quad t = 1, \ldots, n \qquad (84)$$

where $x = (x(1), x(2), x(3))$ is a uniformly distributed in $[-1, 1]^3$ random vector, $a = 2$, $b = 1.2$, $\omega = 1$, $\omega_1 = 1/3$.

The initial conditions have been set to $y(0) = y(-1) = 1$.

Experiments were performed by using $n = 20$ and $N = 10^6$ data points.

The same partition $y = (u, u')^T$, $u \in \mathbb{R}^p$ as in Experiment 1 was chosen with $p = 3, \ldots, 12$.

Table 2 provides Jacobian eigenvalues showing that the correct value $d = 3$ of the intrinsic dimension is estimated by the method.

TABLE 2. Singular values in Narendra experiment.

p	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	λ_8	λ_9	λ_10	λ_{11}	λ_{12}
3	133.5698	39.1274	6.3974	0	0	0	0	0	0	0	0	0
4	14.1856	8.8066	5.7307	1.0000	0	0	0	0	0	0	0	0
5	11.6586	3.6292	2.4125	1.0000	1.0000	0	0	0	0	0	0	0
6	9.8911	2.2837	2.1186	1.0000	1.0000	1.0000	0	0	0	0	0	0
7	7.8889	2.0632	1.5652	1.0000	1.0000	1.0000	1.0000	0	0	0	0	0
8	5.6862	1.8486	1.3928	1.0000	1.0000	1.0000	1.0000	1.0000	0	0	0	0
9	3.7562	1.6179	1.2492	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0	0	0
10	2.6113	1.5336	1.1504	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0	0
11	2.5292	1.4308	1.0843	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0
12	1.9266	1.4208	1.0590	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

C. UNSUPERVISED LEARNING OF SPEECH DATA

As a third example speech data have been used to validate the unsupervised learning approach previously described. An example of signals used in this experiment is shown in Fig. 10, which depicts a portion of the signal corresponding to the vowel 'a' pronounced by an Italian speaker. As you can see the signal is almost periodic, thus every period of length n can be considered as generated by a nonlinear system of the kind given by (4) with random initial conditions.

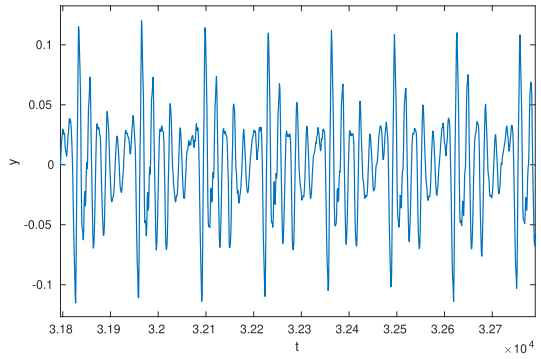


FIGURE 10. A frame of the speech signal corresponding to the vowel 'a' pronounced by an Italian speaker.

The suggested approach for the ID estimation proposed in Section III-F was applied to a set of $N = 33567$ signals of length $n = 140$ extracted from a collection of speech vowels sampled at 16000 Hz, pronounced by different speakers. The same partition $y = (u, u')^T$, $u \in \mathbb{R}^p$ as in synthetic data experiments was used with p ranging from 16 to 26.

TABLE 3. Singular values in experiment 3.

p	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	λ_8	λ_9	λ_{10}	λ_{11}	λ_{12}	λ_{13}	λ_{14}	λ_{15}	λ_{16}	λ_{17}	λ_{18}	λ_{19}	λ_{20}	λ_{21}	λ_{22}	λ_{23}	λ_{24}	λ_{25}	λ_{26}
16	5.0265	3.8932	2.4098	2.1121	1.4039	1.2308	1.2055	1.0425	1.0363	1.0138	1.0090	1.0043	1.0023	1.0018	1.0009	1.0005	0	0	0	0	0	0	0	0	0	0
17	5.2800	4.3764	2.5983	2.2906	1.4021	1.2540	1.1460	1.0464	1.0360	1.0289	1.0084	1.0060	1.0029	1.0017	1.0011	1.0008	1.0003	0	0	0	0	0	0	0	0	0
18	5.2192	4.7599	2.6268	2.2617	1.4672	1.2752	1.1011	1.0746	1.0397	1.0334	1.0207	1.0076	1.0056	1.0025	1.0017	1.0013	1.0008	1.0001	0	0	0	0	0	0	0	0
19	5.2595	4.5508	2.5780	2.5128	1.4871	1.3752	1.1091	1.1044	1.0402	1.0389	1.0255	1.0151	1.0056	1.0039	1.0035	1.0016	1.0011	1.0005	1.0001	0	0	0	0	0	0	0
20	5.5691	4.3462	2.8007	2.5817	1.5834	1.5598	1.1193	1.0944	1.0513	1.0407	1.0314	1.0190	1.0074	1.0060	1.0037	1.0028	1.0011	1.0006	1.0002	1.0001	0	0	0	0	0	0
21	6.0329	4.2077	3.0647	2.4333	1.8421	1.6343	1.1395	1.1148	1.0681	1.0444	1.0399	1.0168	1.0084	1.0063	1.0054	1.0038	1.0014	1.0009	1.0004	1.0002	1.0001	0	0	0	0	0
22	6.3710	4.0137	3.3214	2.3520	1.9466	1.5424	1.1858	1.0981	1.0885	1.0587	1.0361	1.0261	1.0181	1.0073	1.0055	1.0034	1.0020	1.0012	1.0004	1.0004	1.0001	1.0000	0	0	0	0
23	6.8395	3.9992	3.5059	2.3989	2.0078	1.3913	1.2162	1.1512	1.1179	1.0689	1.0422	1.0340	1.0159	1.0110	1.0048	1.0036	1.0020	1.0012	1.0005	1.0002	1.0002	1.0001	1.0000	0	0	0
24	7.4942	3.9982	3.7169	2.5576	2.1287	1.4306	1.2896	1.2027	1.1081	1.0930	1.0421	1.0356	1.0156	1.0085	1.0052	1.0041	1.0020	1.0013	1.0009	1.0005	1.0002	1.0001	1.0000	1.0000	0	0
25	7.9459	4.6569	3.7241	2.9325	2.1928	1.6229	1.2794	1.1728	1.1362	1.0622	1.0431	1.0378	1.0232	1.0165	1.0074	1.0032	1.0019	1.0014	1.0007	1.0006	1.0002	1.0002	1.0001	1.0000	1.0000	0
26	8.3517	6.1497	4.0213	3.1035	2.3893	1.6200	1.2544	1.1699	1.1357	1.0659	1.0426	1.0334	1.0151	1.0113	1.0072	1.0040	1.0016	1.0010	1.0008	1.0005	1.0004	1.0002	1.0001	1.0001	1.0000	1.0000

Table 3 shows the singular values of Jacobian matrix $J(\psi)$ in a single dataset point u.

ID was locally estimated by thresholding the residual energy of singular values vector $\widetilde{\lambda} = \lambda - 1$ (see Fig. 11), using the following criterion

$$\widehat{ID}(u) = \min d : \quad ||\widetilde{\lambda}_{d+1:p}||_1 \leq \theta ||\widetilde{\lambda}||_1 \tag{85}$$

where θ is a threshold parameter.

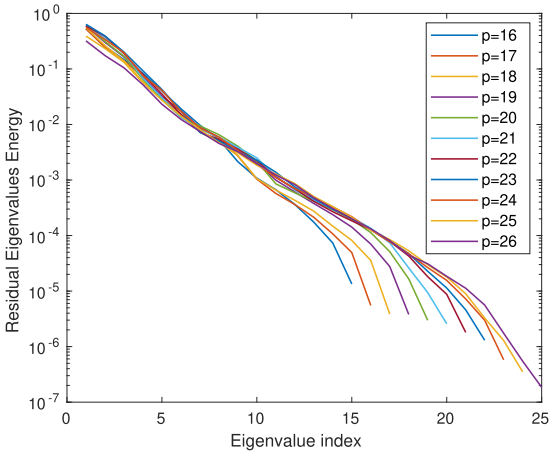


FIGURE 11. Residual Cumulative Eigenvalues Energy for Speech signals.

The global ID was obtained by weighting local estimates with the pdf given by the regression $\widehat{g}(u)$.

$$ID = \frac{\sum_j \widehat{ID}(u_j)\widehat{g}(u_j)}{\sum_j \widehat{g}(u_j)} \tag{86}$$

As shown in Fig. 12 the overall estimation converges to $d = 16$, as p increases.

The parametrization model $G(\cdot)$ was estimated by the multivariate regression based on Particle Bernstein Polynomials. From a signal of length 284200 samples a set of $N = 2030$ signals of length $n = 140$ samples was extracted, that represents the training set.

Fig. 13 shows a comparison between model and data. Also in this case the learning accuracy is very good.

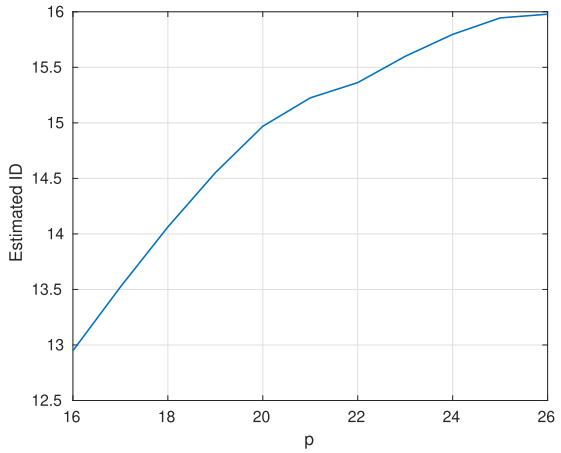


FIGURE 12. Global estimated Intrinsic dimension for Speech signals.

D. UNSUPERVISED LEARNING OF PPG DATA

This experiment was performed on data gathered from photoplethysmografic signals (PPG). The signals used to generate the time series to be identified, belong to the PhysioNet database available in [68], [69] and are referred to several different subjects. From 150470 samples a set of $N = 734$ signals of length $n = 205$ was extracted. Fig. 14 depictes the behaviour of several of such signals. The signals were filtered in order to reduce the amount of the superimposed noise and the signals so obtained are shown in Fig. 15.

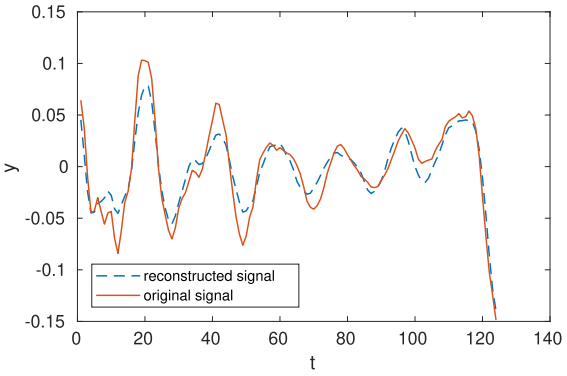
The proposed ID estimation technique was applied by partitioning $y = (u,\ u'),\ u' \in \mathbb{R}^p$ with $p = 8 \ldots 16$.

By deriving a global estimate for *ID* using the same method as for the speech signals, poorer results are obtained since a saturation in $\widehat{ID}$ is not reached as shown in Fig. 17. This is probably due to the lower number of dataset points than those used for previous experiments, which plays a key role in Nadaraya-Watson regression performances.

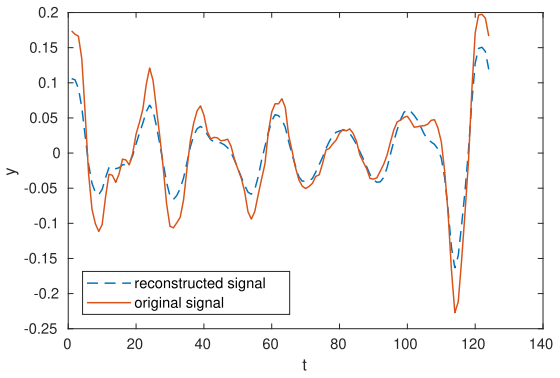
However a reasonable value for *ID* can be guessed by Jacobian eigenvalues shown in TABLE 4 which gives $d = 8$.

Concerning the learning of the parametrization model $G(\cdot)$, from a total of 293237 samples a set of $N = 1141$ signals of length $n = 257$ samples was extracted, that was used as the training set.

With this value of ID using the regression approach based on Particle Bernstein Polynomials to model data, the results shown in Fig. 18 are achieved. As you can see also in this case a good learning accuracy is obtained.



(a)



(b)

FIGURE 13. Comparison of the results achieved by the approximation (80) (broken line) with data (continuous line) in the experiment 3 for two different frames of the same signal.

TABLE 4. Experiment 4.

p	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	λ_8	λ_9	λ_{10}	λ_{11}	λ_{12}	λ_{13}	λ_{14}	λ_{15}	λ_{16}
8	2.1892	1.3426	1.0166	1.0019	1.0004	1.0001	1.0000	1.0000	0	0	0	0	0	0	0	0
9	2.0793	1.3233	1.0205	1.0019	1.0008	1.0001	1.0000	1.0000	1.0000	0	0	0	0	0	0	0
10	1.9888	1.3240	1.0281	1.0013	1.0011	1.0001	1.0001	1.0000	1.0000	1.0000	0	0	0	0	0	0
11	1.9566	1.3333	1.0364	1.0028	1.0009	1.0003	1.0001	1.0000	1.0000	1.0000	1.0000	0	0	0	0	0
12	1.9223	1.3386	1.0406	1.0044	1.0007	1.0004	1.0001	1.0000	1.0000	1.0000	1.0000	1.0000	0	0	0	0
13	1.8952	1.3246	1.0440	1.0054	1.0007	1.0005	1.0001	1.0001	1.0000	1.0000	1.0000	1.0000	1.0000	0	0	0
14	1.8757	1.3196	1.0467	1.0056	1.0011	1.0006	1.0002	1.0001	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0	0
15	1.8511	1.3203	1.0475	1.0055	1.0024	1.0007	1.0004	1.0001	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0
16	1.8466	1.3341	1.0572	1.0050	1.0045	1.0010	1.0005	1.0001	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

E. BIOSIGNAL GENERATION

This last experiment aims to show that the proposed parametrized model for nonlinear dynamical systems can be successfully used for biosignal generation.

In this case the signal to be generated is assumed to be a sequence of frames $y_t \in \mathbb{R}^n$, each partitioned according to

$$y_t = (y_t^{(1)}, y_t^{(2)})^T, \ t = 1, 2, \ldots \tag{87}$$

with $y_t^{(1)} \in \mathbb{R}^{n-d}$, $y_t^{(2)} \in \mathbb{R}^d$. Then two consecutive frames of the sequence are constrained by the following autoregressive model

$$y_{t+1} = G(y_t^{(2)} + \eta) \tag{88}$$

where $G(\cdot)$ is the regression function estimated with the d samples $y_t^{(2)}$ of the previous frame, and η is a noise component with standard deviation σ_η, introduced to avoid a perfect periodicity of the signal.

The generative model given by (87) and (88) has been validated using two dataset of ECG signals from UEA & UCR Time Series Classification Repository [70], [71]: ECG200 and Two Lead ECG. ECG200 consists of 200 samples of an ECG series, of which 133 are labeled as normal and the remaining 67 are myocardial infarctions (abnormal). The length of each series is 96 and traces the electric

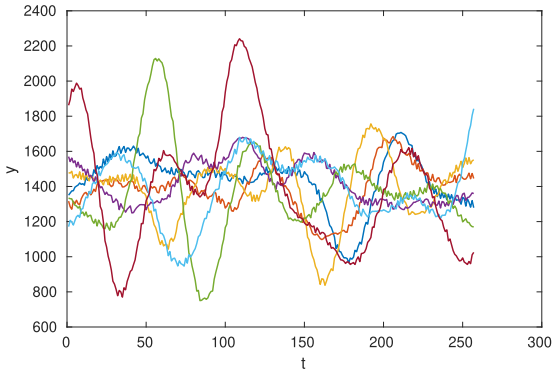


FIGURE 14. Some randomly chosen frames of PPG signals used in the fourth experiment.

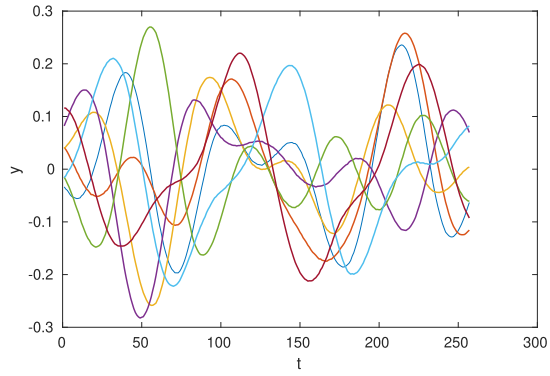


FIGURE 15. The same randomly chosen frames of Fig. 14 after filtering to reduce the effect of noise.

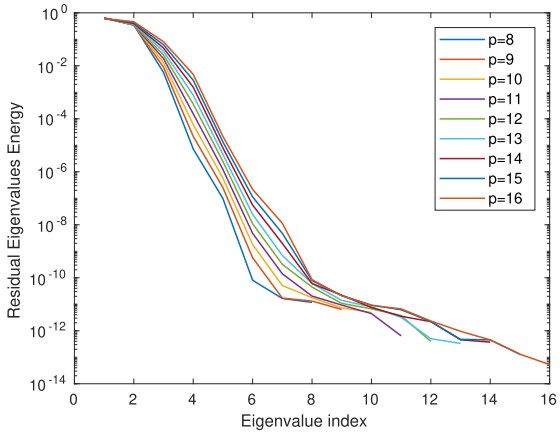


FIGURE 16. Residual Cumulative Eigenvalues Energy for PPG signals.

TABLE 5. Singular values in experiment 5.

p	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	λ_8	λ_9	λ_{10}	λ_{11}	λ_{12}	λ_{13}	λ_{14}	λ_{15}	λ_{16}	λ_{17}	λ_{18}	λ_{19}	λ_{20}	λ_{21}	λ_{22}	λ_{23}	λ_{24}	λ_{25}	λ_{26}	λ_{27}	λ_{28}
20	8.0916	1.9855	1.1639	1.0876	1.0574	1.0205	1.0095	1.0043	1.0026	1.0018	1.0016	1.0010	1.0009	1.0006	1.0004	1.0002	1.0002	1.0001	1.0001	1.0001	0	0	0	0	0	0	0	0
21	8.0458	1.9404	1.1676	1.0901	1.0609	1.0189	1.0100	1.0046	1.0026	1.0017	1.0017	1.0009	1.0008	1.0006	1.0005	1.0002	1.0002	1.0001	1.0001	1.0001	1.0000	0	0	0	0	0	0	0
22	7.9458	1.9002	1.1670	1.0890	1.0706	1.0174	1.0093	1.0044	1.0025	1.0019	1.0016	1.0010	1.0008	1.0006	1.0004	1.0003	1.0002	1.0001	1.0001	1.0001	1.0001	1.0000	0	0	0	0	0	0
23	7.8564	1.8942	1.1841	1.0858	1.0729	1.0171	1.0081	1.0044	1.0028	1.0017	1.0015	1.0009	1.0008	1.0005	1.0004	1.0003	1.0003	1.0001	1.0001	1.0001	1.0001	1.0000	1.0000	0	0	0	0	0
24	7.7886	1.8561	1.1959	1.0896	1.0730	1.0166	1.0074	1.0044	1.0028	1.0017	1.0014	1.0009	1.0007	1.0005	1.0004	1.0004	1.0003	1.0002	1.0001	1.0001	1.0001	1.0000	1.0000	1.0000	0	0	0	0
25	7.7140	1.8390	1.2005	1.0882	1.0719	1.0169	1.0069	1.0045	1.0026	1.0016	1.0013	1.0009	1.0007	1.0005	1.0004	1.0003	1.0003	1.0002	1.0001	1.0001	1.0001	1.0001	1.0000	1.0000	1.0000	0	0	0
26	7.6306	1.8214	1.2089	1.0869	1.0702	1.0163	1.0063	1.0046	1.0023	1.0015	1.0011	1.0009	1.0006	1.0005	1.0005	1.0004	1.0003	1.0002	1.0001	1.0001	1.0001	1.0001	1.0000	1.0000	1.0000	1.0000	0	0
27	7.4641	1.7881	1.2153	1.0880	1.0676	1.0154	1.0058	1.0046	1.0023	1.0014	1.0011	1.0008	1.0006	1.0006	1.0005	1.0004	1.0003	1.0002	1.0001	1.0001	1.0001	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0
28	7.0032	1.7502	1.2071	1.0915	1.0627	1.0255	1.0054	1.0040	1.0026	1.0013	1.0009	1.0008	1.0007	1.0005	1.0005	1.0003	1.0003	1.0002	1.0001	1.0001	1.0001	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

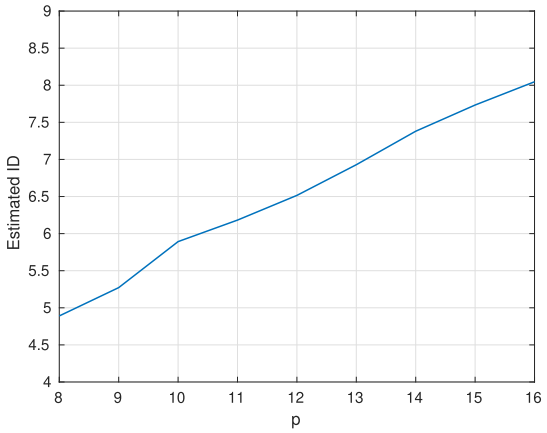


FIGURE 17. Global estimated Intrinsic dimension for PPG signals.

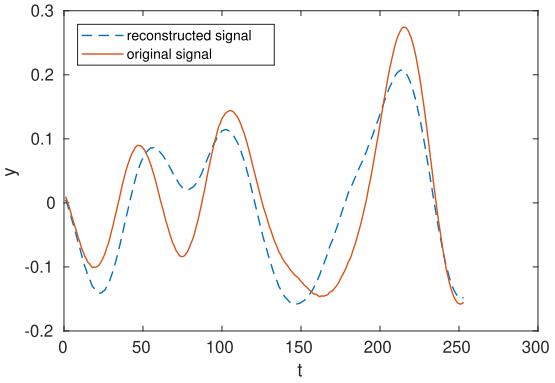


FIGURE 18. Comparison of the results achieved by the approximation (80) (broken line) with data (continuous line) for the experiment 4.

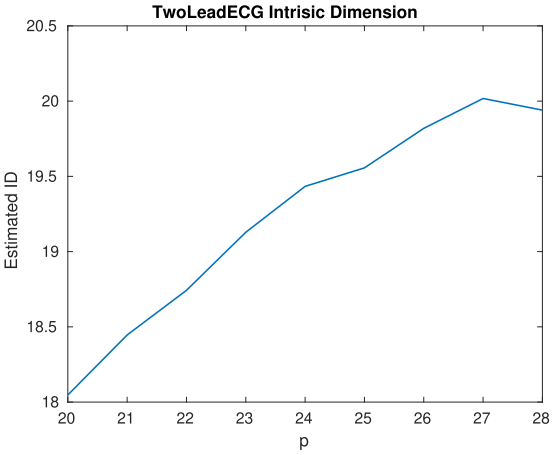


FIGURE 19. Global estimated ID for Two Lead ECG dataset.

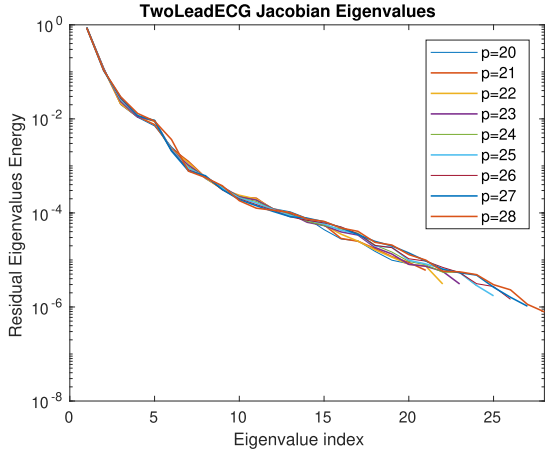


FIGURE 20. Residual cumulative energy for Two Lead ECG dataset.

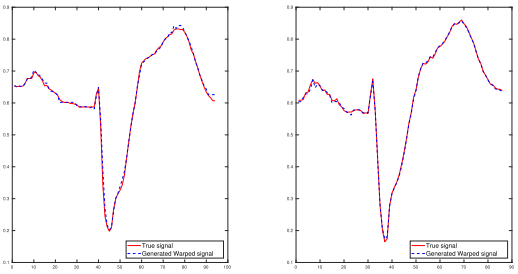
activity during one heartbeat. Two Lead ECG consists of 1, 162 samples of an ECG series whose length is 82. Each signal originates from one of two leads and labeled as class 1 or class 2 depending from which lead was originated. Out of 1, 162 samples, 581 are labeled as class 1 and the remaining 581 are class 2.

The ID for the dataset Two Lead ECG has been estimated with the method discussed in Section III-F using $\theta = 10^{-5}$ in (85). Table 5 reports the singular values so obtained, while Fig. 19 and Fig. 20 show the global estimated ID and the residual cumulative eigenvalues energy respectively. On the basis of these results a value ID $= 20$ has been chosen. For the data of ECG200 a similar approach cannot be used since the size of data matrix (96×100) is insufficient to guarantee an accurate ID estimation, thus the same value ID $= 20$ has been used in this case.

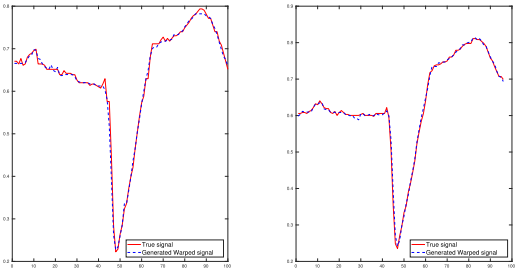
Other model parameters are the polynomial order m and noise standard deviation expressed as a fraction of the norm of the generated frame: $\sigma_\eta^{\%} = \sigma_\eta / ||G(y_t^{(2)})||_2$. The chosen parameters are reported in Table 6. They have been optimized by accurate fitting techniques, i.e., simulated annealing, genetic algorithms and also regression methods. Since the number of parameters to be optimized is very small, as reported in Table 6, we have chosen the well-known simulated annealing algorithm to optimize them, already available in Matlab.

TABLE 6. Parameters of generative model.

	Manifold dimension d	Bernstein order m	Noise standard deviation $\sigma_\eta^{\%}$
ECG200 class 1	20	2	0.02
ECG200 class 2	20	2	0.03
Two Lead ECG class 1	20	25	0.01
Two Lead ECG class 2	20	20	0.07



(a) Two Lead ECG class 1



(b) Two Lead ECG class 2

FIGURE 21. Some examples of the original data and data generated by the proposed approach. The two time-series data have been aligned using the DTW method.

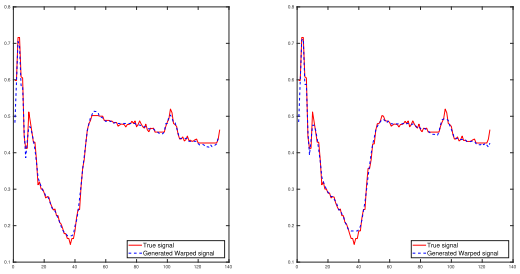
The similarity between the training data set and the data set achieved with the generative model was computed by using the dynamic time warping (DWT) distance between two time-series data $x^{(i)}$ and $x^{(j)}$, calculated as

$$DWT(x^{(i)}, x^{(j)}) = \sqrt{\underset{w_1, w_2 \ldots, w_R}{\operatorname{argmin}} \sum_{r=1, w_r=(k,l)}^{R} (x_k^{(i)}, x_l^{(j)})^2} \quad (89)$$

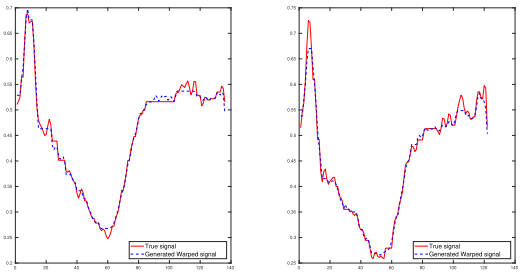
COMPARISON WITH GAN BASED MODEL

Fig. 21 shows the warping of frames generated by the model proposed in this article, compared with the best aligned frames of Two Lead ECG, using DTW for the alignment of the two datasets. Similar results are shown in Fig. 22 for the dataset ECG200. As you can see, the learning accuracy achieved with the proposed model is very good.

Fig. 23 and Fig. 24 compare the quality of the data generated with our model and the model based on generative adversarial networks (GANs), recently proposed in [56], for the data ECG200 and Two Lead ECG respectively. The average DTW distance and standard deviation were used as metrics



(a) ECG200 class 1



(b) ECG200 class 2

FIGURE 22. Some examples of the original data and data generated by the proposed approach. The two time-series data have been aligned using the DTW method.

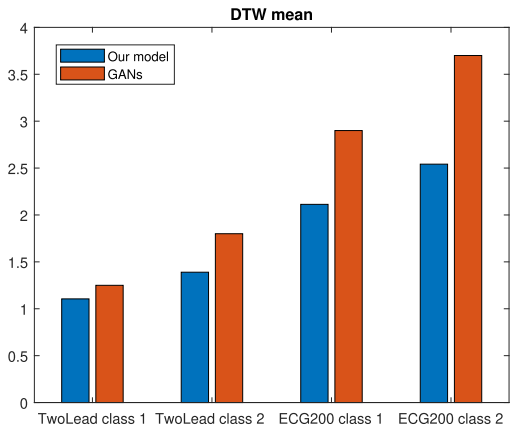


FIGURE 23. Average DTW distance for the two generative models.

to validate the results. The results show that the similarity obtained with our model always outperforms that obtained with GAN based model.

In addition, the approach suggested in this article has also the following advantages:

i) the model does not need to be trained before signal generation;
ii) the fundamental frequency can be easily varied.

Concerning the first point, the model (80) based on particle-Bernstein polynomials is a non parametric model, thus it does not require a training stage in contrast with GANs model which require time-consuming algorithms to determine the unknown coefficients [56].

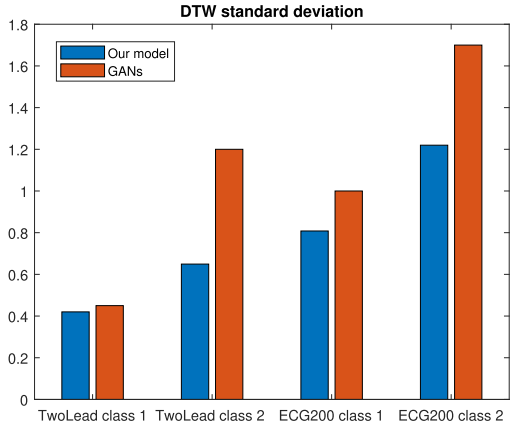


FIGURE 24. Standard deviation of DTW distance for the two generative models.

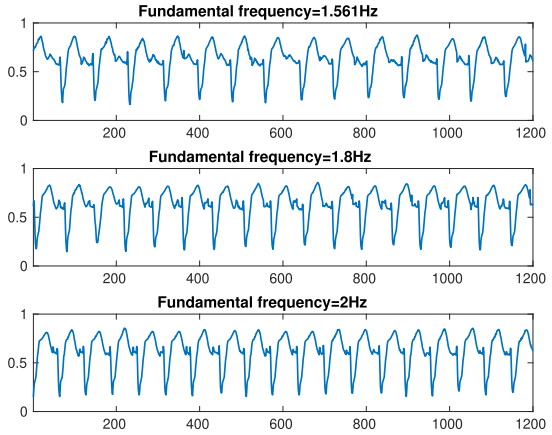


FIGURE 25. Examples of ECG signal generated by the proposed model from Two Lead ECG dataset with different values of f_0.

With reference to the point *ii)*, the fundamental frequency f_0 of the ECG signal can be easily varied transforming $G(\cdot) \in \mathbb{R}^n$ in (88) to the function $G^s(\cdot) \in \mathbb{R}^s$ such that

$$G^s(\cdot) = H^s G(\cdot), \tag{90}$$

where $H^s \in \mathbb{R}^{s \times n}$ is a random permutation matrix and the size $s = \left\lfloor \frac{F_s}{f_0} \right\rfloor < n$ is the ratio between the sampled frequency F_s and f_0. Here the symbol $\lfloor x \rfloor$ denotes the nearest integer less than or equal to x. It can be easily shown that (90) corresponds to downsampling G of a factor s/n proportional to the frequency f_0. Fig. 25 reports some examples of ECG signal generated by our model from Two Lead ECG dataset with different values of f_0.

A time-varying f_0 will result in a more realistic ECG signal whose fundamental frequency is not perfectly constant along time. Ideally we can consider a sinusoidal trend of the signal given by:

$$f_0(t) = F_0 + |\Delta f \sin(\Omega t)| \tag{91}$$

Fig. 26 compares the ideal frequency behaviour $f_0(t)$ and the true fundamental frequency of the generated signal achieved through the transformation (90) showing a good agreement between them. Obviously the generated f_0 cannot be chosen arbitrarily high, since this will lead to a total distortion of signal pulses. A reasonable maximum limit is twice the original fundamental frequency, $f_{0,\max} \approx 2F_0$.

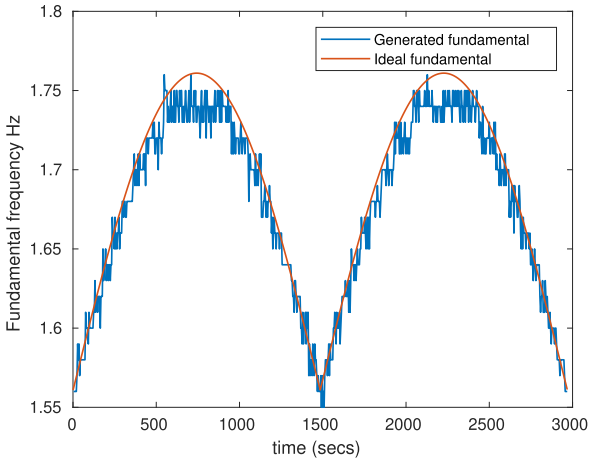


FIGURE 26. Comparison of the ideal fundamental $f_0(t)$ (red) and the true fundamental frequency (blue) generated by (90).

TABLE 7. Classification accuracy with data augmentation using GANs.

n° of synthetic generated samples	Data augmentation with GANs SVM accuracy [%]
0 (real data)	81.61
100	85.69
200	88.75
400	91.03
600	92.35
800	**93.32**

F. BIOSIGNAL CLASSIFICATION

This latest experiment was conducted to demonstrate the effectiveness of our approach in improving the classification of biosignals, by enhancing the size of the training data.

One of the main limitations of learning-based techniques for biosignal classification is that a large amount of training data is required to obtain enough accuracy. Indeed, use of machine learning techniques on small datasets subjects the models to issues of over-fitting, noise and outliers. Unfortunately, availability and access to large datasets is limited in the real-world. This is primarily due to the limited number of people submitting to data collection. Additionally, in order to label each sample of collected data special qualification or expert knowledge is required.

DATA AUGMENTATION BY SYNTHETIC DATA

Data augmentation, that is the technique used to increase the amount of data, is one of the easiest ways to improve classification accuracy [72]. In this context a very effective approach

TABLE 8. Classification accuracy with data augmentation using our approach.

n° of synthetic generated samples	Data augmentation with our approach SVM accuracy [%]
0 (real data)	81.61
100	87.34
200	90.99
400	94.12
600	95.65
800	**96.49**

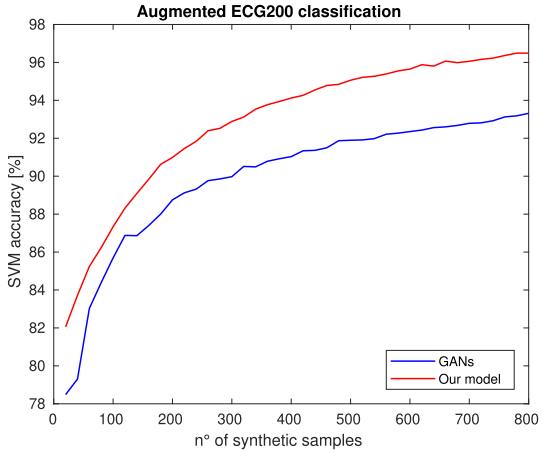


FIGURE 27. Comparison between GANs and the proposed approach for the classification of augmented ECG200 dataset, by varying the number of synthetic samples and using SVM classification algorithm.

is to add newly created synthetic data from existing data, to enlarge the dataset, using a biosignal generative model [2].

In this experiment we used both GANs and our approach to demonstrate the effectiveness of these techniques in biosignal classification. The experiment was conducted on the dataset ECG200, previously described. Table 7 reports the results of the classification performed by support vector machine (SVM) algorithm on real data and real data plus synthetic data generated with GANs. The experiment was repeated using a different number of synthetic generated samples. As you can see the classification accuracy increases as the size of synthetic data set increases, clearly showing the benefit of data augmentation in biosignal classification. Besides, in order to evaluate the effect of using an embedded low-dimensional generative model for classification, our approach was used to conduct the same experiment. Table 8 reports the results obtained and confirms that a generative model with reduced dimensionality is able to create new synthetic data that improve classification. Finally, the two techniques for biosignals generation used in this experiment, i.e. GANs and our approach, are compared in Fig. 27, that depicts the SVM classification accuracy as a function of the number of synthetic samples added to the ECG200 data set. As you can see our method outperforms GANs for all the sizes of synthetic data set, confirming the validity of the proposed approach.

VI. CONCLUSION

In this article a machine learning-based approach for biosignal generative modeling that takes advantage of the manifold concept, is presented. In particular it has been proven that data, assuming they are sampled from a biosignal generated by a nonlinear dynamical system, lie on a nonlinear manifold between data and some latent variables. The dimension of such variables, called intrinsic dimension of data, is a fundamental parameter in the unsupervised learning of data, as it allows data can be accurately modelled. Thus a crucial step in determining the local parametrization that represents the manifold is the estimation of ID, that is the dimension of the parametrization. A very effective method based on the Jacobian of the parametrization is used in this article, and to improve the accuracy an optimal version of Nadaraya-Watson derivative estimator is developed. Once the latent variables have been discovered, the initial unsupervised problem reduces to a supervised problem. To face this problem an effective machine learning technique for regression based on the set of the so called Particle-Bernstein polynomials has been adopted, that does not depend on unknown parameters to be determined, thus avoiding a time-consuming training stage as required by other learning techniques. Experimental tests on both synthetic and real-world data have validated the effectiveness of the proposed algorithm. With respect to models based on generative adversarial networks our approach offers some advantages: it guarantees a better similarity of the generated data, it does not require a time-consuming training stage, it is able to generate a large variety of signals.

APPENDIX
BANDWIDTH OPTIMIZATION FOR GRADIENT ESTIMATION

Here we will describe in details the p-dimensional error analysis of Nadaraya-Watson estimator and the bandwidth optimization technique proposed in this work.

In the same fashion as [64], gradient estimation error $\mathcal{E}_{H_k}(u)$ can be expressed by:

$$\begin{aligned}\mathcal{E}_{H_k}(u) &= \widehat{\nabla f} - \nabla f = \\ &= \frac{1}{\widehat{g}(u)}\left(\widehat{\nabla h} - \nabla h\right) - \frac{\nabla f}{\widehat{g}(u)}\left(\widehat{g}(u) - g(u)\right) - \\ &\quad - \frac{\widehat{f}(u)}{\widehat{g}(u)}\left(\widehat{\nabla g} - \nabla g\right) - \frac{\nabla g}{\widehat{g}(u)}\left(\widehat{f}(u) - f(u)\right) \end{aligned} \quad (92)$$

We now compute mean and variance of each term by making Taylor approximations of $f(t)$ and $g(t)$ around estimation point x:

$$\begin{aligned} f(t) &\approx f(u) + \nabla f^T(t-u) + \frac{1}{2}(t-u)^T H_f(u)(t-u) \\ g(t) &\approx g(u) + \nabla g^T(t-u) + \frac{1}{2}(t-u)^T H_g(u)(t-u) \end{aligned} \quad (93)$$

If we make the following substitution in the integrals $t = x - H_k y$, and denote as $z = H_k y$:

$$\begin{aligned}
&E(c_k(u)) \\
&= \int_{\mathbb{R}^p} \frac{1}{D_k} K\left(H_k^{-1}(t-u)\right) g(t)dt \\
&= \int_{\mathbb{R}^p} K(y)\left(g(u) - \nabla g^T H_k y + \frac{1}{2} y^T H_k^T H_g(u) H_k y\right) dt \\
&= g(u) + \int_{\mathbb{R}^p} \frac{1}{2} y^T H_k^2 H_g(u) y K(y) dy \\
&= g(u) + \frac{1}{2}\mathrm{Tr}\left(H_k^2 H_g(u)\right) = g(u) + \frac{1}{2}\sum_{j=1}^{p} h_j^2 \frac{\partial^2 g}{\partial u_j^2} \qquad (94)
\end{aligned}$$

$$\begin{aligned}
&E(c_k^2(u)) \\
&= \int_{\mathbb{R}^p} \frac{1}{D_k^2} K^2\left(H_k^{-2}(t-u)\right) g(t)dt \\
&= \frac{D_k}{D_k^2}\int_{\mathbb{R}^p} K^2(y)\left(g(u) - \nabla g^T H_k y + \frac{1}{2} y^T H_k^2 H_g(u) y\right) dy \\
&= g(u)\frac{\int_{\mathbb{R}^p} K^2(y)dy}{D_k} + \frac{1}{D_k}\int_{\mathbb{R}^p} y^T H_k^2 H_g(u) y K^2(y) dy \qquad (95)
\end{aligned}$$

For a generic Kernel we can define:

$$\xi_m = \int_{\mathbb{R}^p} y_i^m K^2(y)dy \;\; i = 1, \ldots, p$$

Therefore we can write:

$$E(c_k^2(u)) = \frac{1}{D_k}\left(\xi_0 g(u) + \frac{\xi_2}{2}\mathrm{Tr}(H_k^2 H_g(u))\right) \qquad (96)$$

Each terms differs from the target in both variance and mean; the bias is very difficult to estimate directly as it depends on second-order derivatives of $g(u)$ and it is a second-order polynomial of H_k: $E(\widehat{g}(u) - g(u)) = O(H_k^2)$. In the same manner we calculate mean and variance of other terms:

$$\begin{aligned}
E(d_k(u)) &= -g(u)\int_{\mathbb{R}^p} H_k^{-1} y K(y)dy + \nabla g \\
&\quad - \frac{H_k}{2}\int_{\mathbb{R}^p} y(y^T H_k^2 H_g(u) y) K(y)dy \\
&= \nabla g + O(H_k^2) \qquad (97) \\
E(||d_k(u)||_2^2) &= \frac{g(u)}{D_k}\int_{\mathbb{R}^p} ||H_k^{-1} y||_2^2 K^2(y)dy \\
&\quad - \frac{\nabla g^T}{D_k}\int_{\mathbb{R}^p} H_k y ||H_k^{-1} y||_2^2 K^2(y)dy + \\
&\quad + \frac{1}{2D_k}\int_{\mathbb{R}^p} y^T H_k^2 H_g(u) y ||H_k^{-1} y||_2^2 K^2(y)dy \\
&= \frac{\xi_2 p\langle \mathrm{diag}(H_k^{-2})\rangle}{D_k} g(u) + O\left(\frac{R(H_k^2)}{D_k}\right) \qquad (98)
\end{aligned}$$

where $R(H_k^2)$ is a second-order rational function of $H_k(j)^2$.

$$R(H_k) = \frac{\sum_{j=1}^{p} a_j H_k^2(j)}{\sum_{j=1}^{p} b_j H_k^2(j)} \qquad (99)$$

$$\begin{aligned}
E(b_k(u)) &= f(u)g(u) + O(H_k^2) \\
E(b_k(u)^2) &= \xi_0 \frac{f^2(u)g(u)}{D_k} + O\left(\frac{R(H_k^2)}{D_k}\right) \qquad (100) \\
E(a_k(u)) &= \nabla(fg) + O(H_k^2) \\
E(||a_k(u)||_2^2) &= \frac{\xi_2 p\langle \mathrm{diag}(H_k^{-2})\rangle}{D_k} f^2(u)g(u) + O\left(\frac{R(H_k^2)}{D_k}\right) \qquad (101)
\end{aligned}$$

By taking expectation of eqn. 92 we obtain nonzero cross-correlation terms in addition to absolute errors:

$$\begin{aligned}
&g^2(u)E(||\nabla\widehat{f} - \nabla f||_2^2) \\
&\quad = E(||\nabla\widehat{h} - \nabla h||_2^2) \\
&\qquad + f^2(u)E(||\nabla\widehat{g} - \nabla g||_2^2) \\
&\qquad + ||\nabla f||_2^2 E(\widehat{g}(u) - g(u))^2 \\
&\qquad - 2f(u)E\left[(\nabla\widehat{h} - \nabla h)^T(\nabla\widehat{g} - \nabla g)\right] \\
&\qquad - 2\nabla f^T E\left[(\nabla\widehat{h} - \nabla h)(\widehat{g}(u) - g(u)\right] \\
&\qquad + 2f(u)\nabla f^T E\left[(\nabla\widehat{g} - \nabla g)(\widehat{g}(u) - g(u))\right. \qquad (102)
\end{aligned}$$

If we evaluate these correlation terms we can write:

$$\begin{aligned}
E(\nabla\widehat{h}^T\nabla\widehat{g}) &= \frac{\xi_2 p\left\langle \frac{1}{H_k(j)^2}\right\rangle}{D_k} f(u)g(u) + O(\frac{R(H_k^2)}{D_k}) \\
E(\widehat{g}(u)\nabla\widehat{h}) &= O(\frac{R(H_k^2)}{D_k}) \\
E(\widehat{g}(u)\nabla\widehat{g}) &= O(\frac{R(H_k^2)}{D_k}) \qquad (103)
\end{aligned}$$

By substituting these results into 103 it is easy to verify that similarly to the 1-dimensional case [64], total variance term proportional to $\frac{R(H_k^2)}{D_k}$ is only dependent on noise power $\sigma^2 = E(\epsilon_k^2)$.

Therefore the total error will be expressed by:

$$\mathcal{E}_{H_k}^2(u) = \frac{\sigma^2\xi_2 p}{g(u)N^2}\sum_{k=1}^{N}\frac{\langle \mathrm{diag}(H_k^{-2})\rangle}{D_k} + \frac{1}{N^2}\left\|\sum_{k=1}^{N} B_k(u)\right\|_2^2 \qquad (104)$$

where $B_k(u)$ is the k-th point contribute to $\mathrm{Bias}(u) = E(\nabla\widehat{f} - \nabla f)$. If we suppose to use the same bandwidth for all data points $H_k \equiv H$ we will have:

$$\mathcal{E}_H^2(u) = \frac{\sigma^2\xi_2 p\langle \mathrm{diag}(H_k^{-2})\rangle}{Ng(u)D_k} + ||\mathrm{Bias}(u)||_2^2 \qquad (105)$$

Another simplification we can apply is to use the same bandwidth across dimensions: $H = \widetilde{h}I$; since $\text{Bias}(u) = O(H^2)$ it will be a simple quadratic function of $\widetilde{h}^2$:

$$\text{Bias}(u) = \widetilde{h}^2 Q(u) \tag{106}$$

The Jacobian estimation error with a constant and scalar bandwidth finally becomes:

$$\mathcal{E}^2_{\widetilde{h}}(u) = \frac{\xi_2 \sigma^2}{Ng(u)\widetilde{h}^{p+2}} + \widetilde{h}^4 ||Q(u)||_2^2 \tag{107}$$

The minimum value of $\mathcal{E}^2_{\widetilde{h}}$ is obtained with:

$$h^{\text{opt}}(u) = \left(\frac{(p+2)\sigma^2 \xi_2}{Ng(u)||Q(u)||_2^2} \right)^{\frac{1}{p+6}} \tag{108}$$

Practical implementation of 64 however requires estimation of the quantities σ^2, $Q(u)$.

Additive noise power σ^2 can be estimated with leave-one-out regression on function $f(u)$ [73]:

$$\widehat{\sigma^2} = \left\langle \left(f(u_i) - \widehat{f}_{-i}(u_i) \right)^2 \right\rangle \tag{109}$$

where $\widehat{f}_{-i}(u_i)$ is the estimate of $f(u)$ made by excluding point u_i from regression:

$$\widehat{f}_{-i}(u_i) = \frac{\sum_{k \neq i} \frac{w_k}{D_k} K\left(H_k^{-1}(u_k - u) \right)}{\sum_{k \neq i} \frac{1}{D_k} K\left(H_k^{-1}(u_k - u) \right)} \tag{110}$$

An approximate estimation of the bias can be given with results developed in [74]:

$$\begin{aligned} E(\nabla\widehat{h} - \nabla h) &\approx \frac{1}{N} \sum_{k=1}^{N} f(u_k) L(u_k) - \nabla\widehat{h} \\ E(\nabla\widehat{g} - \nabla g) &\approx \frac{1}{N} \sum_{k=1}^{N} L(u_k) - \nabla\widehat{g} \\ L(u_k) &= \frac{1}{D_k} (K * \nabla K)_{H_k^{-1}(u-u_k)} \end{aligned} \tag{111}$$

where $*$ is the convolution operator and ∇K is the gradient of unitary kernel. If using the Gaussian kernel we have simply:

$$L(u_k) = \frac{1}{D_k} \nabla K \left(\frac{H_k^{-1}}{\sqrt{2}} (u - u_k) \right) \tag{112}$$

Denoted as $M(u) = g(u)Q(u)$, it simpler to use the estimated quantity $\widehat{M}(u)$ than $\widehat{Q}(u)$. By observing 111 and 112 we can write in a compact form:

$$\widehat{M}(u) = \frac{1}{s^2} \left[\left(\nabla\widehat{h} - \widehat{f}\nabla\widehat{g} \right)_{\sqrt{2}s} - \left(\nabla\widehat{h} - \widehat{f}\nabla\widehat{g} \right)_s \right] \tag{113}$$

where bandwidth s needs to be properly adjusted.

By substituting into (108) we obtain the final formula for the choice of bandwidth:

$$\widehat{h}^{\text{opt}}(u) = \left(\frac{(p+2)\widehat{\sigma}^2 \xi_2 \widehat{g}(u)}{N||\widehat{M}(u)||_2^2} \right)^{\frac{1}{p+6}} \tag{114}$$

REFERENCES

[1] V. Singhal, A. Majumdar, and R. K. Ward, "Semi-supervised deep blind compressed sensing for analysis and reconstruction of biomedical signals from compressive measurements," *IEEE Access*, vol. 6, pp. 545–553, 2018.

[2] R. Shamsuddin, B. M. Maweu, M. Li, and B. Prabhakaran, "Virtual patient model: An approach for generating synthetic healthcare time series data," in *Proc. IEEE Int. Conf. Healthcare Informat. (ICHI)*, Jun. 2018, pp. 208–218.

[3] D. Farina, A. Crosetti, and R. Merletti, "A model for the generation of synthetic intramuscular EMG signals to test decomposition algorithms," *IEEE Trans. Biomed. Eng.*, vol. 48, no. 1, pp. 66–77, 2001.

[4] P. E. McSharry, G. D. Clifford, L. Tarassenko, and L. A. Smith, "A dynamical model for generating synthetic electrocardiogram signals," *IEEE Trans. Biomed. Eng.*, vol. 50, no. 3, pp. 289–294, Mar. 2003.

[5] V. Z. Marmarelis, *Nonlinear Dynamic Modeling of Physiological Systems*, vol. 10. Hoboken, NJ, USA: Wiley, 2004.

[6] G. Shi, H. Huang, and L. Wang, "Unsupervised dimensionality reduction for hyperspectral imagery via local geometric structure feature learning," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 8, pp. 1425–1429, Aug. 2020.

[7] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 1, pp. 4–37, Jan. 2000.

[8] S. Løkse, F. M. Bianchi, and R. Jenssen, "Training echo state networks with regularization through dimensionality reduction," *Cognit. Comput.*, vol. 9, no. 3, pp. 364–378, Jun. 2017.

[9] S. S. On, "Dimensionality reduction methods," *IEEE Signal Process. Mag.*, vol. 28, no. 2, pp. 1–128, Mar. 2011.

[10] F. Luo, H. Huang, Y. Duan, J. Liu, and Y. Liao, "Local geometric structure feature for dimensionality reduction of hyperspectral imagery," *Remote Sens.*, vol. 9, no. 8, p. 790, Aug. 2017.

[11] F. Luo, L. Zhang, B. Du, and L. Zhang, "Dimensionality reduction with enhanced hybrid-graph discriminant learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 8, pp. 5336–5353, Aug. 2020.

[12] H. S. Seung, "COGNITION: The manifold ways of perception," *Science*, vol. 290, no. 5500, pp. 2268–2269, Dec. 2000.

[13] S. T. Roweis, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, Dec. 2000.

[14] L. K. Saul and S. T. Roweis, "Think globally, fit locally: Unsupervised learning of low dimensional manifolds," *J. Mach. Learn. Res.*, vol. 4, pp. 119–155, Dec. 2003.

[15] J. B. Tenenbaum, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, Dec. 2000.

[16] R. Talmon, S. Mallat, H. Zaveri, and R. R. Coifman, "Manifold learning for latent variable inference in dynamical systems," *IEEE Trans. Signal Process.*, vol. 63, no. 15, pp. 3843–3856, Aug. 2015.

[17] M. H. C. Law and A. K. Jain, "Incremental nonlinear dimensionality reduction by manifold learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 3, pp. 377–391, Mar. 2006.

[18] T. Lin and H. Zha, "Riemannian manifold learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 5, pp. 796–809, May 2008.

[19] Z. Zhang, J. Wang, and H. Zha, "Adaptive manifold learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 2, pp. 253–265, Feb. 2012.

[20] Y. Wang, J. Yang, and H. Liu, "Acoustic targets feature extraction method based on manifold learning," *Electron. Lett.*, vol. 48, no. 3, pp. 139–140, Feb. 2012.

[21] H. Qiao, P. Zhang, D. Wang, and B. Zhang, "An explicit nonlinear mapping for manifold learning," *IEEE Trans. Cybern.*, vol. 43, no. 1, pp. 51–63, Feb. 2013.

[22] V. S. Tomar and R. C. Rose, "A family of discriminative manifold learning algorithms and their application to speech recognition," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 22, no. 1, pp. 161–171, Jan. 2014.

[23] X. Xing, K. Wang, Z. Lv, Y. Zhou, and S. Du, "Fusion of local manifold learning methods," *IEEE Signal Process. Lett.*, vol. 22, no. 4, pp. 395–399, Apr. 2015.

[24] X. Xu, Z. Huang, L. Zuo, and H. He, "Manifold-based reinforcement learning via locally linear reconstruction," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 4, pp. 934–947, Apr. 2017.

[25] T. Shnitzer, R. Talmon, and J.-J. Slotine, "Manifold learning with contracting observers for data-driven time-series analysis," *IEEE Trans. Signal Process.*, vol. 65, no. 4, pp. 904–918, Feb. 2017.

[26] H. Gu, X. Wang, X. Chen, S. Deng, and J. Shi, "Manifold learning by curved cosine mapping," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 10, pp. 2236–2248, Oct. 2017.
[27] M. Längkvist, L. Karlsson, and A. Loutfi, "A review of unsupervised feature learning and deep learning for time-series modeling," *Pattern Recognit. Lett.*, vol. 42, pp. 11–24, Jun. 2014.
[28] K. Fukunaga and D. R. Olsen, "An algorithm for finding intrinsic dimensionality of data," *IEEE Trans. Comput.*, vol. C-20, no. 2, pp. 176–183, Feb. 1971.
[29] F. Camastra, "Data dimensionality estimation methods: A survey," *Pattern Recognit.*, vol. 36, no. 12, pp. 2945–2954, Dec. 2003.
[30] K. W. Pettis, T. A. Bailey, A. K. Jain, and R. C. Dubes, "An intrinsic dimensionality estimator from near-neighbor information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-1, no. 1, pp. 25–37, Jan. 1979.
[31] G. V. Trunk, "Stastical estimation of the intrinsic dimensionality of a noisy signal collection," *IEEE Trans. Comput.*, vol. C-25, no. 2, pp. 165–171, Feb. 1976.
[32] P. J. Verveer and R. P. W. Duin, "An evaluation of intrinsic dimensionality estimators," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 1, pp. 81–86, Jan. 1995.
[33] M. Kirby, *Geometric Data Analysis: An Empirical Approach to Dimensionality Reduction and the Study of Patterns*. Hoboken, NJ, USA: Wiley, 2000.
[34] R. N. Shepard, A. K. Romney, and S. B. Nerlove, *Multidimensional Scaling: Theory*. New York, NY, USA: Seminar Press, 1972.
[35] C. Kuan Chen and H. C. Andrews, "Nonlinear intrinsic dimensionality computations," *IEEE Trans. Comput.*, vol. C-23, no. 2, pp. 178–184, Feb. 1974.
[36] P. Grassberger, "An optimized box-assisted algorithm for fractal dimensions," *Phys. Lett. A*, vol. 148, nos. 1–2, pp. 63–68, Aug. 1990.
[37] B. Kégl, "Intrinsic dimension estimation using packing numbers," in *Proc. NIPS*, 2002, pp. 681–688.
[38] P. Grassberger and I. Procaccia, "Measuring the strangeness of strange attractors," *Phys. D, Nonlinear Phenomena*, vol. 9, nos. 1–2, pp. 189–208, Oct. 1983.
[39] C. Turchetti and L. Falaschetti, "A manifold learning approach to dimensionality reduction for modeling data," *Inf. Sci.*, vol. 491, pp. 16–29, Jul. 2019.
[40] G. B. Giannakis and E. Serpedin, "A bibliography on nonlinear system identification," *Signal Process.*, vol. 81, no. 3, pp. 533–580, Mar. 2001.
[41] M. Schetzen, "Nonlinear system modeling based on the Wiener theory," *Proc. IEEE*, vol. 69, no. 12, pp. 1557–1573, Dec. 1981.
[42] M. Inagaki and H. Mochizuki, "Bilinear system identification by volterra kernels estimation," *IEEE Trans. Autom. Control*, vol. 29, no. 8, pp. 746–749, Aug. 1984.
[43] A. Novak, B. Maillou, P. Lotton, and L. Simon, "Nonparametric identification of nonlinear systems in series," *IEEE Trans. Instrum. Meas.*, vol. 63, no. 8, pp. 2044–2051, Aug. 2014.
[44] C. Yu, C. Zhang, and L. Xie, "A new deterministic identification approach to hammerstein systems," *IEEE Trans. Signal Process.*, vol. 62, no. 1, pp. 131–140, Jan. 2014.
[45] C. Turchetti, G. Biagetti, F. Gianfelici, and P. Crippa, "Nonlinear system identification: An effective framework based on the Karhunen–Loève transform," *IEEE Trans. Signal Process.*, vol. 57, no. 2, pp. 536–550, Feb. 2009.
[46] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Trans. Neural Netw.*, vol. 1, no. 1, pp. 4–27, Mar. 1990.
[47] S. Lu and T. Basar, "Robust nonlinear system identification using neural-network models," *IEEE Trans. Neural Netw.*, vol. 9, no. 3, pp. 407–429, May 1998.
[48] M. Espinoza, J. A. K. Suykens, and B. De Moor, "Kernel based partially linear models and nonlinear identification," *IEEE Trans. Autom. Control*, vol. 50, no. 10, pp. 1602–1606, Oct. 2005.
[49] J. Xie, "Time series prediction based on recurrent LS-SVM with mixed kernel," in *Proc. Asia–Pacific Conf. Inf. Process.*, vol. 1, Jul. 2009, pp. 113–116.
[50] R. T. Farouki, "The bernstein polynomial basis: A centennial retrospective," *Comput. Aided Geometric Des.*, vol. 29, no. 6, pp. 379–419, Aug. 2012.
[51] G. Biagetti, P. Crippa, L. Falaschetti, and C. Turchetti, "Machine learning regression based on particle bernstein polynomials for nonlinear system identification," in *Proc. IEEE 27th Int. Workshop Mach. Learn. Signal Process. (MLSP)*, Sep. 2017, pp. 1–6.
[52] A. Koski, "Modelling ECG signals with hidden Markov models," *Artif. Intell. Med.*, vol. 8, no. 5, pp. 453–471, Oct. 1996.
[53] A. Gogna, A. Majumdar, and R. Ward, "Semi-supervised stacked label consistent autoencoder for reconstruction and analysis of biomedical signals," *IEEE Trans. Biomed. Eng.*, vol. 64, no. 9, pp. 2196–2205, Sep. 2017.
[54] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
[55] S. Haradal, H. Hayashi, and S. Uchida, "Biosignal data augmentation based on generative adversarial networks," in *Proc. 40th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2018, pp. 368–371.
[56] S. Harada, H. Hayashi, and S. Uchida, "Biosignal generation and latent variable analysis with recurrent generative adversarial networks," *IEEE Access*, vol. 7, pp. 144292–144302, 2019.
[57] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
[58] V. Volterra, *Theory of Functionals and of Integral and Integro-Differential Equations*. New York, NY, USA: Dover, 1959.
[59] J. J. Bussgang, L. Ehrman, and J. W. Graham, "Analysis of nonlinear systems with multiple inputs," *Proc. IEEE*, vol. 62, no. 8, pp. 1088–1119, Aug. 1974.
[60] M. B. Priestley, *Non-Linear and Non-Stationary Time Series Analysis*. London, U.K.: Academic, 1988.
[61] H. Cramér and M. R. Leadbetter, *Stationary and Related Stochastic Processes: Sample Function Properties and Their Applications*. Hoboken, NJ, USA: Wiley, 1967.
[62] L. H. Koopmans, *The Spectral Analysis of Time Series*. Amsterdam, The Netherlands: Elsevier, 1995.
[63] M. Priestley, *Spectral Analysis and Time Series, Two-Volume Set*, vols. 1–2. Amsterdam, The Netherlands: Elsevier, 1981.
[64] B. Bercu, S. Capderou, and G. Durrieu, "Nonparametric estimation of the derivative of the regression function: Application to sea shores water quality," 2016, *arXiv:1606.06033*. [Online]. Available: https://arxiv.org/abs/1606.06033
[65] A. Carini and G. L. Sicuranza, "Fourier nonlinear filters," *Signal Process.*, vol. 94, pp. 183–194, Jan. 2014.
[66] A. Carini, S. Cecchi, L. Romoli, and G. L. Sicuranza, "Legendre nonlinear filters," *Signal Process.*, vol. 109, pp. 84–94, Apr. 2015.
[67] J. Berchtold and A. Bowyer, "Robust arithmetic for multivariate bernstein-form polynomials," *Comput.-Aided Des.*, vol. 32, no. 11, pp. 681–689, Sep. 2000.
[68] A. J. Casson, A. Vazquez Galvez, and D. Jarchi, "Gyroscope vs. accelerometer measurements of motion from wrist PPG during physical exercise," *ICT Express*, vol. 2, no. 4, pp. 175–179, Dec. 2016.
[69] A. J. Casson. (2017). *Wrist PPG During Exercise*. [Online]. Available: https://physionet.org/works/WristPPGduringexercise
[70] A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh, "The great time series classification bake off: A review and experimental evaluation of recent algorithmic advances," *Data Mining Knowl. Discovery*, vol. 31, no. 3, pp. 606–660, May 2017.
[71] A. Bagnall, J. Lines, W. Vickers, and E. Keogh, *The UEA & UCR Time Series Classification Repository*. Accessed: 2018. [Online]. Available: http://www.timeseriesclassification.com
[72] Q. Wen, L. Sun, X. Song, J. Gao, X. Wang, and H. Xu, "Time series data augmentation for deep learning: A survey," 2020, *arXiv:2002.12478*. [Online]. Available: http://arxiv.org/abs/2002.12478
[73] B. E. Hansen, "Lecture notes on nonparametrics," Univ. Wisconsin-Madison, Madison, WI, USA, Lecture Notes, 2009.
[74] A. V. Dobrovidov and I. M. Ruds'ko, "Bandwidth selection in nonparametric estimator of density derivative by smoothed cross-validation method," *Autom. Remote Control*, vol. 71, no. 2, pp. 209–224, Feb. 2010.

LORENZO MANONI received the B.Sc. and M.Sc. degrees in electronics engineering from Università Politecnica delle Marche, Ancona, Italy, in 2015 and 2018, respectively, where he is currently pursuing the Ph.D. degree with the Dipartimento di Ingegneria dell'Informazione (DII). His current research interests include signal processing, embedded systems, machine learning, algorithms analysis and design, and bio-signal analysis.

CLAUDIO TURCHETTI (Life Member, IEEE) received the Laurea degree in electronics engineering from the University of Ancona, Ancona, Italy, in 1979. He joined Università Politecnica delle Marche, Ancona, in 1980, where he was the Head of the Department of Electronics, Artificial Intelligence and Telecommunications for five years and is currently a Full Professor of micro-nanoelectronics and design of embedded systems. He has published more than 160 journal and conference papers, and two books. The most relevant articles were published in IEEE JOURNAL OF SOLID-STATE CIRCUITS, IEEE TRANSACTIONS ON ELECTRON DEVICES, IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON SIGNAL PROCESSING, IEEE TRANSACTIONS ON CYBERNETICS, IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS, IEEE TRANSACTIONS ON CONSUMER ELECTRONICS, and *Information Sciences*. He has held a variety of positions as the Project Leader in several applied research programs developed in cooperation with small, large, and multinational companies in the field of microelectronics. His current research interests include statistical device modeling, RF integrated circuits, device modeling at nanoscale, computational intelligence, signal processing, pattern recognition, system identification, machine learning, and neural networks. He has served as a Program Committee Member for several conferences and a Reviewer for several scientific journals. He is a member of the Computational Intelligence and Signal processing Society. He has been an Expert Consultant of the Ministero dell'Università e Ricerca.

LAURA FALASCHETTI (Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degree in electronics engineering from Università Politecnica delle Marche, Ancona, Italy, in 2008, 2012, and 2016, respectively. She has collaborated as a Research Fellow with the Department of Information Engineering (DII), Università Politecnica delle Marche, from 2012 to 2013. She is currently a Postdoctoral Research Fellow with DII and a Contract Professor for the course electronic systems, at electronic and biomedical engineering, Università Politecnica delle Marche. Her current research interests include embedded systems, machine learning, neural networks, manifold learning, pattern recognition, signal processing, image processing, speech and speaker recognition, and speech synthesis.