Combining reservoir computing and variational inference for efficient one-class learning on dynamical systems

Diego Cabrera
Department of Mechanical Engineering
Universidad Politécnica Salesiana sede Cuenca
Cuenca, Ecuador

Fernando Sancho
Department of Computer Science and Artificial Intelligence
Universidad de Sevilla
Sevilla, España

Felipe Tobar
Center for Mathematical Modeling
Universidad de Chile
Santiago, Chile

***Abstract*—Usually, time series acquired from some measurement in a dynamical system are the main source of information about its internal structure and complex behavior. In this situation, trying to predict a future state or to classify internal features in the system becomes a challenging task that requires adequate conceptual and computational tools as well as appropriate datasets. A specially difficult case can be found in the problems framed under one-class learning. In an attempt to sidestep this issue, we present a machine learning methodology based in Reservoir Computing and Variational Inference. In our setting, the dynamical system generating the time series is modeled by an Echo State Network (ESN), and the parameters of the ESN are defined by an expressive probability distribution which is represented as a Variational Autoencoder. As a proof of its applicability, we show some results obtained in the context of condition-based maintenance in rotating machinery, where vibration signals can be measured from the system, our goal is fault detection in helical gearboxes under realistic operating conditions. The results show that our model is able, after trained only with healthy conditions, to discriminate successfully between healthy and faulty conditions.**

***Index Terms*—Dynamical System Modeling, Reservoir Computing, Variational Inference**

I. Introduction

Modelling a dynamical system from time series is an important but complex task. The approaches range from Takens' theorem based methods [1] to the discovery of strange attractors in the trajectories generated by the time series, or to new methods for parameters estimation of models with a predefined structure, e.g. [2]. A complete review of the methodologies that have been applied for the identification and prediction of dynamical systems can be seen in [3].

However, in most cases, a preliminary knowledge of the structure of the dynamical system is required in order to have a prefixed structure with optimizable parameters or, on the contrary, a greater diversity in the time series available for the generation of the model [4]. The first case has the drawback of limiting the resulting model only to a set of possible dynamical systems that can be expressed by the prefixed structure. The second one has the disadvantage of requiring a large number of time series obtained from a, as complete as possible, variety of states of the dynamical system.

The contribution of this paper is the introduction of a novel methodology for the modelling of a dynamical system from time series measured from one only known state. Here we combine ideas from Reservoir Computing [5], Variational Inference [6] and Deep Learning [7]. Our approach uses an Echo State Network (ESN) as a means for representing the time series as occurrences from a random variable, then its probability distribution is approximated using a variational autoencoder (VAE). After the model construction, new time series are classified using the reconstruction error as a metric from the generative model included on VAE.

This paper is organized as follows. In Section II we explain basics from Echo State Network and Variational Autoencoder. In Section III each phase of our methodology is shown in detail, which begins with the acquisition and preprocessing of signals and culminates with a similarity metric that will provide a classification of new signals. In Section IV we apply the methodology to detect faults in a helical gearbox from a set of vibration signals having a high complexity because of their variability. In this section the experimental configuration, the values chosen for the hyperparameters of the models, and the analysis of the results obtained in the application are detailed. Finally, we conclude highlighting the main features of our approach from a theoretical and practical point of view.

II. Background

In this section we review the fundamentals of the two models used in this work: Echo State Networks and Variational AutoEncoders.

A. Echo state network

Echo State Networks (ESN) is a biologically-inspired, recurrent neural network model proposed by Herber Jager in [8]. This bio-inspired model eliminates the problem of gradient loss, known to affect the training of deep networks (a recurrent network can be seen as a stack of layers receiving as input the

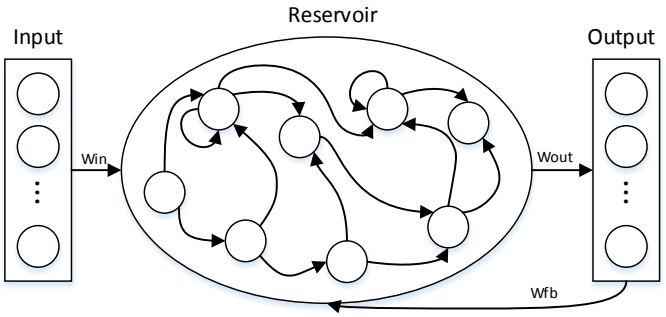


Fig. 1. Architecture of an ESN

response of the previous instant), by separating the model in independent layers: Input, Reservoir, and Output layer.

1) Architecture: The architecture of an ESN is shown in Fig. 1. For every time instant t we will denote the input by $\mathbf{u}(t) \in \mathbb{R}^{N_{in}}$, the output by $\mathbf{y}(t) \in \mathbb{R}^{N_{out}}$, and the activation of neurons in the reservoir by $\mathbf{x}(t) \in \mathbb{R}^{D}$, also called the state vector. Parameters of the input layer, output layer, reservoir and feedback are given by $W_{in} \in \mathbb{R}^{D \times N_{in}}$, $W_{out} \in \mathbb{R}^{N_{out} \times D}$, $W_{state} \in \mathbb{R}^{D \times D}$ and $W_{fb} \in \mathbb{R}^{D \times N_{out}}$ respectively.

In this situation, the dynamics that occurs in an ESN follows the next state equations:

$$\mathbf{x}(t) = f\Big(W_{in}\mathbf{u}(t) + W_{state}\mathbf{x}(t-1) + W_{fb}\mathbf{y}(t-1)\Big) \quad (1)$$

$$\mathbf{y}(y) = W_{out}\mathbf{x}(t) \quad (2)$$

where f is the (same) activation function in every neuron from the reservoir. Above equations exposes that an excitation at the input at an instant t induces a non-linear alteration in the state of the reservoir which is also affected by the state and outputs of the instant $t-1$. The output of the network depends linearly on the reservoir state.

2) Input-reservoir optimization: In this model W_{in}, W_{state} and W_{fb} are chosen and optimized independently of W_{out}. The reservoir matrix, W_{state}, defines the connections in the reservoir, which must mainly comply with two properties: echo state, and separability.

Echo state property means that the effect of the input and previous states decays with time, otherwise the reservoir is said to be unstable. For example, if $f = tanh$, $\mathbf{u}(t) = 0$, and $\rho(W_{state}) > 1$ (the spectral radius of W_{state}, the absolute value of its largest eigenvalue), then the reservoir is known to be unstable. Consequently, from a practical point of view, we want $\rho(W_{state}) < 1$, although this does not guarantee the stability of the reservoir. Spectral radius is directly related to the amount of memory required for the application, higher $\rho(W_{state})$, greater the memory capacity of the reservoir.

On the other hand, separability property means that it generates different states from different inputs. This property can be achieved by: 1) initializing W_{state} as a sparse matrix from a sample of a standard normal distribution, and 2) guaranteeing a large enough number of neurons in the reservoir. The size of the reservoir is directly linked to the computational capacity of the network.

Input parameters W_{in} are typically initialized with a sampling of the same distribution as W_{state}, with the difference that in this case the connections are dense. However, in most applications W_{fb} is initialized with 0, unless a network capable of generating time series from no input (free-running mode) is required.

3) Output layer optimization: Having achieved a stable reservoir and with enough memory capacity and computation power, the only parameters to optimize are W_{out}, which represents a linear regression model mapping a set of known states into a target time series $\mathbf{y}^{target}(t)$.

Consequently, the cost function for this model is given by the mean square error (MSE):

$$E(\mathbf{y}, \mathbf{y}^{target}) = \frac{1}{N_{out}} \frac{1}{T} \sum_{i=1}^{N_{out}} \sum_{t=1}^{T} [\mathbf{y}_i(t) - \mathbf{y}_i^{target}(t)]^2 \quad (3)$$

As usual, the model can be optimized in the following way:

$$W_{out} = \arg\min_{W_{out}} E(\mathbf{y}, \mathbf{y}^{target}) \quad (4)$$

To solve it we proceed to obtain a state $\mathbf{x}(t)$ at each instant of the input $\mathbf{u}(t)$, for $1 \leq t \leq T$. These states can be stacked as columns of a matrix that we denote by $\mathbf{X} \in \mathbb{R}^{D \times T}$. Similarly we stack each desired output vector $\mathbf{y}^{target}(t)$ in the matrix $\mathbf{Y}^{target} \in \mathbb{R}^{N_{out} \times T}$. Using these two new matrices, the optimization problem (4) is solved by Ridge Regression [9] as (I denotes the identity matrix):

$$W_{out} = \mathbf{Y}^{target}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T + \beta I)^{-1} \quad (5)$$

B. Variational autoencoder

Variational Autoencoder (VAE) [10] is a machine learning model based on the theory of variational inference and enhanced with the computational capacity of deep learning algorithms. VAE attempts to discover a hidden structure in the data from a set of samples which captures complex relationships in the elements of a visible random variable x to be modelled. This is achieved by projecting x, following a complex probability distribution, into a new variable z (called latent variable), with much simpler probability distribution, and then trying to recover x with a new projection from z.

This procedure is inspired by the inverse transform method:

$$x := F_x^{-1}(F_z(z)) \quad (6)$$

where F_x^{-1} is the quantile function of x and F_z is the cumulative density function of z. As this composition cannot be explicitly calculated, VAE approximates it through a generative model built from the available dataset of samples.

1) Formalization: From a formal point of view, VAE attempts to maximize a variational function $\mathcal{L}_{VAE}$ given by:

$$\mathcal{L}_{VAE}(q_\phi(z|x)) = \int q_\phi(z|x) ln\, p_\theta(x|z) dz - \int q_\phi(z|x) \frac{q_\phi(z|x)}{p(z)} dz \quad (7)$$

where $q_\phi(z|x)$ (depends on a set of parameters ϕ) is the model responsible for the first projection, and approximates the probability distribution of the best code z given the variable x; $p(z)$ is the probability distribution of the latent variable, which must be known and simple (usually, $p(z)$ is defined as a Gaussian distribution with as many components as the problem requires); and $p_\theta(x|z)$ (depends on a set of parameters θ) is a generative model responsible of the projection from z to x.

We seek to maximize $\mathcal{L}_{VAE}$ because it is a lower bound of the log-likelihood of the data $\ln p(x)$. If this lower bound is optimized it would represent the best computationally achievable model for the distribution of the x variable. To achieve this goal we note that the first term is the expectation $\mathbb{E}_{q_\phi}[ln\, p_\theta(x|z)]$, which can be maximized by finding the parameters θ that reduce the reconstruction error of x from a sampling of the latent variable z. The second term is the KL-divergence between the q_ϕ model and the prior $p(z)$, which can be minimized by finding the parameters ϕ that allows the mapping from a sampling of x to the parameters that define the prior. If $p(z)$ is chosen as a Gaussian distribution, then we try to go from the data sampled from an unknown and complex distribution to a set of means and variances according to the known distribution.

III. Methodology

In most cases, the only source of information available about a dynamical system comes from time series extracted from measurements of a subset of its variables. This information can be used for multiple purposes, such as: estimating the state of other variables, predicting future events in the dynamical system, or detecting anomalies in its evolution. A data-oriented approach for these tasks is based on the characterization of the dynamical system by direct measurement of the evolution of the variables of interest in their different states, which allows to obtain a set of tuples $(time\_series,\ state\_label)$, where $state\_label$ is the current state, or a future state or condition of the dynamical system. Our goal is to find high-level patterns in the time series data that allow their mapping towards the label. Although the acquisition of the data in all possible states of the dynamical system would help to achieve this goal, it is in fact a very expensive or even impossible labelling process.

As a extremal case of classification, in this section we propose a methodology for a dynamical system modelling which considers only information coming from time series obtained in one only specific known state of the variable of interest. This proposal is robust under changes on the other variables of the dynamical system. Fig. 2 shows the methodology, which is composed of four main stages:

1) Acquisition and preprocessing.
2) Unsupervised feature extraction by representation learning.
3) Learning of a probabilistic model over the new representation space
4) Inference over previous model.

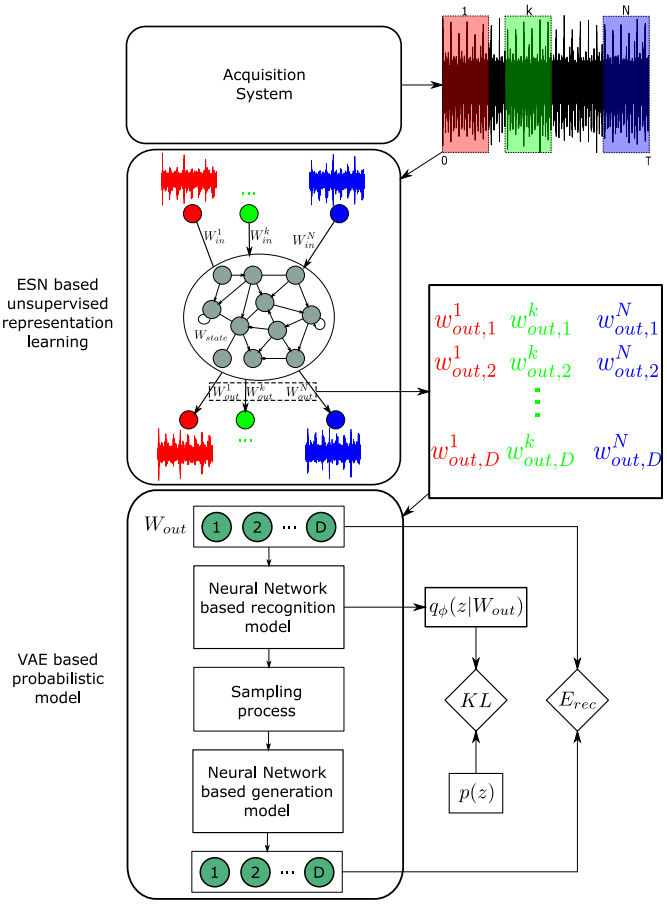


Fig. 2. Proposed methodology

A. Acquisition and preprocessing

A signal measured in a multi-component dynamical system can exhibit a chaotic behaviour [4] due to the interaction between internal/external noise sources and all its elements. When periodicity exists, to mitigate the effects of noise and highlight the important information, the signals are typically measured and synchronously averaged using external sensors to find the start and end of each period. Unfortunately, including additional devices increases the costs and it is not always physically possible to perform such measurement.

Our methodology avoids the use of additional devices for acquisition stage, but some simple preprocessing step is required. If $y(n) \in \mathbb{R}$ with $n = 1, ..., T$ is the signal, then the following normalizing correction is applied:

$$\mathbf{y}'(t) = \frac{\mathbf{y}(t) - min(\mathbf{y})}{max(\mathbf{y}) - min(\mathbf{y})} \in [0, 1] \tag{8}$$

From now on, we will consider only normalized signals. Also, for every signal $\mathbf{y}$, a set of N sub-signals $(\mathbf{y}^1, ..., \mathbf{y}^k, ..., \mathbf{y}^N)$ are extracted using a sliding window of prefixed length and sliding step. It is worth noting that depending of length and step, the sub-signals $\mathbf{y}^{k-1}, \mathbf{y}^k, \mathbf{y}^{k+1}$ could be overlapped

which is a desirable property to capture the intrinsic temporal relationships between them.

B. Unsupervised feature extraction

For every signal we will use the following ESN [5] to anticipate one step (predict $\mathbf{y}(s+1)$ from $\{\mathbf{y}(t) : t \leq s\}$):

$$\mathbf{x}(t) = f\left(W_{in}\mathbf{y}(t-1) + W_{state}\mathbf{x}(t-1)\right) \quad (9)$$

$$\mathbf{y}_{app}(t) = W_{out}\mathbf{x}(t) \quad (10)$$

where we can note that $\mathbf{y}_{app}(t)$ is computed from the previous input $\mathbf{y}(t-1)$.

The number of inputs N_{in} depends on the number of signals obtained from the dynamical system. As the goal is to predict the same input signals, then $N_{out} = N_{in}$. W_{in} and W_{state} are initialized according to subSection II-A and fixed for all $\mathbf{y}^k$ signals. Therefore, the only parameters to be learned to predict $\mathbf{y}^k(t)$ are W^k_{out}. In this way, W^k_{out} encodes $\mathbf{y^k}$, and for $\mathbf{y^k}$ longer enough, this encoding is invariant under entering new temporal values, and then W^k_{out} is a static representation of the input time series. Matrix W_{out}, composed by W^k_{out} grouped column-wise, represents an encoding of the entire normalized time-series $\mathbf{y}$.

C. Learning of a probabilistic model

From a probabilistic perspective, W_{out} can be seen as a random vector with an unknown complex and hard to model probability distribution. In order to solve this problem, we propose to use a VAE that will encode W_{out} in a simpler lower dimensional latent variable $\mathbf{z}$.

Although both q_ϕ and p_θ in VAE can be approximated with several data-based learning models, the most common are neural networks because they are proven to be universal approximators [11]. We will require two neural networks, one *coding network* for q_ϕ, that will encode a multivariable Gaussian probability distribution; and one *generation network* for p_θ, that will reconstruct the original variable. As they are part of the same process of optimizing the lower bound $\mathcal{L}_{VAE}$, they are trained together by using some version of the gradient descendant algorithm, firstly back-propagating the reconstruction error through the layers of the generation network, and then back-propagating through the layers of the coding network. This implies that it is necessary to back-propagate the error through a random variable, which is not possible. In order to solve this problem, we will make use of the *reparametrization trick*, which prevents the back-propagation of the error by a random variable by choosing the variables z of our process as a Gaussian distribution $z = u + \sigma \odot \epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$ becomes one more input to the network. On the other hand, it is known that KL-divergence loss and reconstruction error constitute together the total loss to optimize and, knowing that $z \sim \mathcal{N}(\mu, cov)$ (being cov the covariance matrix), they can be obtained by using the next equations:

$$KL_{loss} = \frac{-1}{2N}\sum_{k=1}^{N}\sum_{j=1}^{|\mathbf{z}|} 1 + ln[(\sigma_j^k)^2] - (\mu_j^k)^2 - (\sigma_j^k)^2 \quad (11)$$

$$E_{rec} = \frac{-1}{N}\sum_{k=1}^{N}\sum_{i=1}^{D} w^k_{out,i} \cdot ln(\hat{w}^k_{out,i}) + (1 - w^k_{out,i}) \cdot ln(1 - \hat{w}^k_{out,i}) \quad (12)$$

$$Loss = KL_{loss} + E_{rec} \quad (13)$$

D. Inference

After the model has been obtained, it is possible to perform inference about the state of the variable of interest from new time series acquired from the dynamical system. For this, we maintain W_{in} and W_{out} for the ESN, and the weights from the VAE neural networks.

To evaluate a new signal the same Acquisition and Pre-processing process is followed to generate a batch of time series. Then, with the ESN a static representation is obtained from each batch element, obtaining a new matrix W_{out}. Later, every k column from W_{out} is passed through the coding and generation networks of the VAE model, where E^k_{rec} is calculated. Finally, L_{avg} is calculated by:

$$L_{avg} = \frac{1}{N}\sum_{k=1}^{N} E^k_{rec} \quad (14)$$

that provides the closeness between the new signal and the distribution of signals corresponding to the original dynamical system from where the model was generated.

IV. Application to Fault Detection

We are now able to offer an application of previous methodology to a real dynamical system. The objective is the fault detection in a mechanical component of a rotative machinery, namely a helical gear, where its wear is our variable of interest. To achieve this, two processes must be carried out: learning of healthy state (from the variable of interest) and on-line testing.

The learning of healthy state is carried out minimizing the Loss (13) for a set of vibration signals acquired from different operational conditions of rotation speed and load with the knowledge of the healthy condition of the gear (without wear). On-line testing is performed with the input of unknown-state vibration signal to the previously learned model and taking as output the average reconstruction error given by (14). Large negative values indicate a high probability that the signal corresponds to a state of machinery without fault, and small negative values (or non-negative) correspond to states of faulty machinery.

A. Gear rig tests

The experiments were conducted to identify the presence of breakage tooth fault in a helical gear at different levels of severity independently of the operational speed and load in the mechanical system. The motor drives the input shaft to

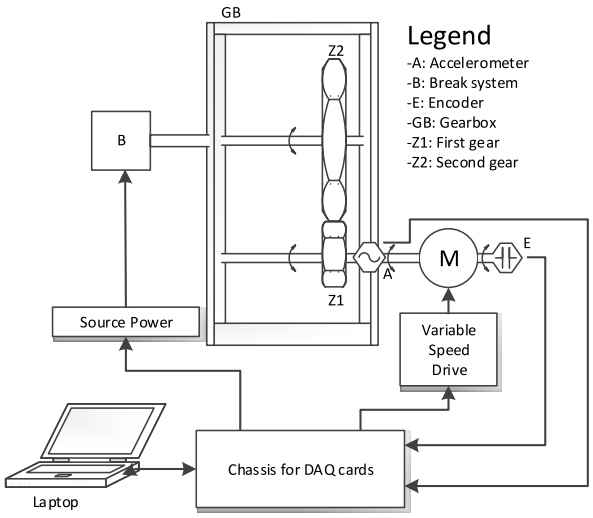


Fig. 3. Block diagram of the experimental setup

Code	Description	Damage (mm)	Percentage (%)
P1	Level 1 or Normal	0.0	100.0
P2	Level 2	2.37	88.42
P3	Level 3	4.0	80.42
P4	Level 4	5.73	71.94
P5	Level 5	7.6	62.81
P6	Level 6	10.57	48.29
P7	Level 7	12.37	39.48
P8	Level 8	14.33	29.85
P9	Level 9	17.5	14.36
P10	Level 10 or without tooth	20.43	0.0

TABLE I
DAMAGE LEVELS OF HELICAL GEAR TOOTH BREAKAGE FAULT

speeds of $480\,\mathrm{rpm}$, $720\,\mathrm{rpm}$ and $900\,\mathrm{rpm}$ (coded as F1, F2, and F3, respectively). The experiments were carried out on a gearbox fault diagnosis test-rig (fabricated by the GIDTEC group of the Universidad Politécnica Salesiana, Ecuador) with the configuration shown in Fig. 3. For each input speed, three output loads of $0\,\mathrm{V}$, $10\,\mathrm{V}$ and $30\,\mathrm{V}$ (coded as L1, L2, and L3, respectively) are applied through a magnetic break controlled by a high current voltage source and coupled to the output shaft through a belt. The test gear is the input pinion (diameter of $76\,\mathrm{mm}$, $30\,\mathrm{tooth}$, pressure angle of $20°$, helix angle of $20°$), with different levels of breakage in a tooth as specified in Table I (P1 represents healthy state).

For each possible combination of speed and load, 5 vibration signals of $280\,001\,\mathrm{samples}$ each one ($\approx 5.6s$), acquired at $50\,000\,\mathrm{samples/s}$, provide a set of 45 training signals used to build the model in healthy-state of gear. The sliding window has a length of $50\,000\,\mathrm{samples}$ ($1s$) and sliding step of $10\,000\,\mathrm{samples}$. For the unsupervised feature extraction stage, the size of W^k_{out} is 1001 (1000 weights and 1 bias term). Each, the recognition and generation models are composed by one hidden layer of 1000 neurons, and use 10 Gaussians (20 values) for the latent space $\mathbf{z}$.

For the testing of the model, for each possible combination of speed, load and P-state (including healthy-state), 5 vibration signals with the same conditions as above are acquired, given a total of 450 testing signals.

B. Results and Analysis

Fig. 4 shows the L_{avg} obtained by testing the learned model with new acquired vibration signals. Note that the maximum value of L_{avg} in a healthy-state is -147.2 (reached in F1 L2 conditions, Fig. 4(a)). On the other hand, under fault presence, the lower value of L_{avg} is -8.6 (reached in F2 L3 conditions, Fig. 4(b)). The large distance between these two cases (138.4) increments the confidence of the result. As speed increases this distance between healthy and fault states increases (-391.2, in Fig. 4(c)). Also, we can observe that higher the load, higher the distance.

Consequently, and independently on speed and load conditions, and on the detection of faults (incipient as P2, or higher level severity damage as P3-P7), the model allows to select a decision threshold providing a 100% of accuracy rate.

V. CONCLUSIONS

In this paper a novel methodology, based on a machine learning approach, for modelling dynamical systems has been presented. The resulting model can successfully approximate complex probability distributions on a new representation space which is learned in an unsupervised way from time series without a prior knowledge. As we have shown in the applications, the methodology can be used in fault detection tasks in rotative machinery where the model is built uniquely from healthy vibration signals at different operational conditions, and later it can be used for fault detection.

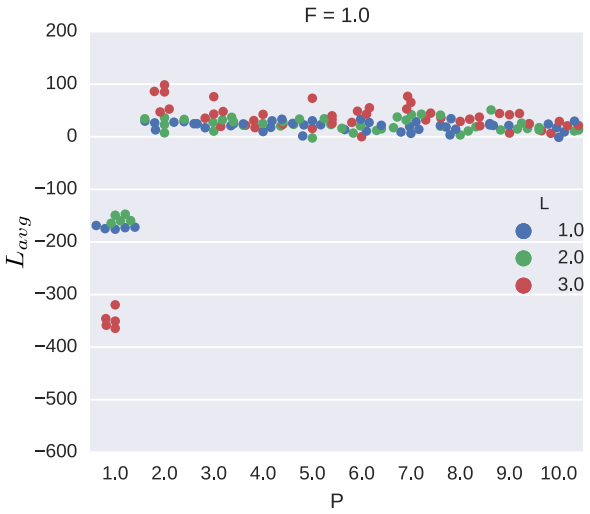
The average reconstruction error, L_{avg}, is proposed as a similarity metric to discriminate between healthy and faulty states, where small errors represent a healthy-state and big errors represent a faulty-state.

Our methodology presents two main advantages in relation to other machine learning and signal-based approaches to fault detection reported in the literature:

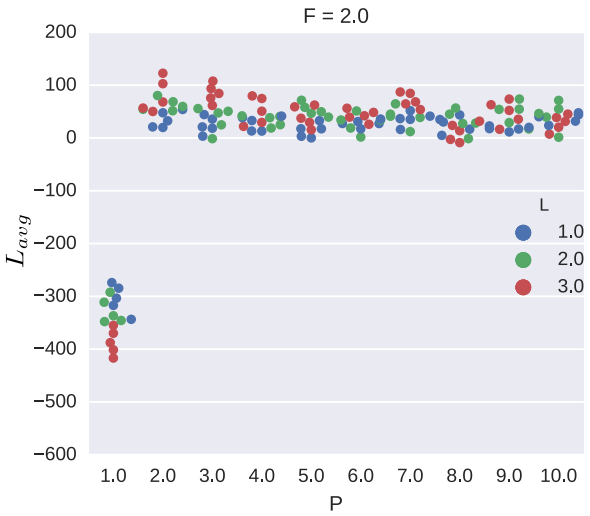
1) Contrarily to other learning-based methodologies, that address the problem as binary (or multi) classification task requiring examples from normal and faulty states, the presented method only needs normal-state vibration signals, which are easily (and cheaper) to obtain.
2) In comparison to other signal-based methods, where a specific prior knowledge of the dynamical system under study is needed, the proposed method does not require expert knowledge due to unsupervised feature extraction stage.

ACKNOWLEDGMENTS

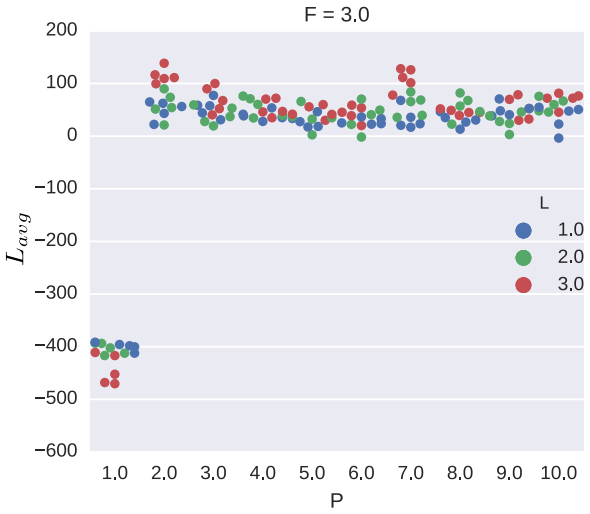
The authors want to thank to R&D projects TIN2012-37434 and TIN2013-41086-P supported by Ministerio de Economía y Competitividad of Gobierno de España and co-financed by the European FEDER funds by support of this research work. F. Tobar acknowledges financial support from CONICYT projects PAI-82140061 and Basal-CMM. The work was



(a) Test to Speed F1



(b) Test to Speed F2



(c) Test to Speed F3

Fig. 4. Testing results of the learned model at all possible combinations of P-state F-speed and L-load.

sponsored in part by the GIDTEC project No. 003-002-2016-03-03. The experimental work was developed at the GIDTEC research group lab of the Universidad Politécnica Salesiana de Cuenca, Ecuador.

References

[1] F. Takens, "Detecting strange attractors in turbulence," Dynamical Systems and Turbulence, Warwick 1980, pp. 366–381, 1981. [Online]. Available: http://dx.doi.org/10.1007/BFb0091924

[2] C. Tao, Y. Zhang, and J. Jiang, "Estimating system parameters from chaotic time series with synchronization optimized by a genetic algorithm," Physical Review E, vol. 76, no. 1, Jul 2007. [Online]. Available: http://dx.doi.org/10.1103/PhysRevE.76.016209

[3] W. Wang, Y. Lai, and C. Grebogi, "Data based identification and prediction of nonlinear and complex dynamical systems," Physics Reports, vol. 644, pp. 1–76, Jul 2016. [Online]. Available: http://dx.doi.org/10.1016/j.physrep.2016.06.004

[4] I. Wattar, W. Hafez, and Z. Gao, "Model-based diagnosis of chaotic vibration signals," IECON'99. Conference Proceedings. 25th Annual Conference of the IEEE Industrial Electronics Society (Cat. No.99CH37029), 1999. [Online]. Available: http://dx.doi.org/10.1109/IECON.1999.819378

[5] M. Lukosevicius and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," Computer Science Review, vol. 3, no. 3, pp. 127–149, Aug 2009. [Online]. Available: http://dx.doi.org/10.1016/j.cosrev.2009.03.005

[6] D. Blei, A. Kucukelbir, and J. McAuliffe, "Variational inference: A review for statisticians," Jan. 2016.

[7] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016, http://www.deeplearningbook.org.

[8] H. Jaeger, "Adaptive nonlinear system identification with echo state networks," in Advances in Neural Information Processing Systems 15 [Neural Information Processing Systems, NIPS 2002, December 9-14, 2002, Vancouver, British Columbia, Canada], S. Becker, S. Thrun, and K. Obermayer, Eds. MIT Press, 2002, pp. 593–600. [Online]. Available: http://papers.nips.cc/paper/2318-adaptive-nonlinear-system-identification-with-echo-state-networks

[9] D. Marquardt and R. Snee, "Ridge regression in practice," The American Statistician, vol. 29, no. 1, p. 3, Feb 1975. [Online]. Available: http://dx.doi.org/10.2307/2683673

[10] D. Kingma and M. Welling, "Auto-encoding variational bayes." CoRR, vol. abs/1312.6114, 2013. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr1312.html#KingmaW13

[11] K. Hornik, "Approximation capabilities of multilayer feedforward networks," Neural Networks, vol. 4, no. 2, pp. 251–257, Jan 1991. [Online]. Available: http://dx.doi.org/10.1016/0893-6080(91)90009-T