

Determination of the edge of criticality in echo state networks through Fisher information maximization

Lorenzo Livi, *Member, IEEE*, Filippo Maria Bianchi, and Cesare Alippi, *Fellow, IEEE*

Abstract—It is a widely accepted fact that the computational capability of recurrent neural networks is maximized on the so-called “edge of criticality”. Once the network operates in this configuration, it performs efficiently on a specific application both in terms of (i) low prediction error and (ii) high short-term memory capacity. Since the behavior of recurrent networks is strongly influenced by the particular input signal driving the dynamics, a universal, application-independent method for determining the edge of criticality is still missing. In this paper, we aim at addressing this issue by proposing a theoretically motivated, unsupervised method based on Fisher information for determining the edge of criticality in recurrent neural networks. It is proven that Fisher information is maximized for (finite-size) systems operating in such critical regions. However, Fisher information is notoriously difficult to compute and requires the analytic form of the probability density function ruling the system behavior. The paper takes advantage of a recently-developed non-parametric estimator of the Fisher information matrix and provides a method to determine the critical region of echo state networks, a particular class of recurrent networks. The considered control parameters, which indirectly affect the echo state network performance, are explored to identify those configurations lying on the edge of criticality and, as such, maximizing Fisher information and computational performance. Experimental results on benchmarks and real-world data demonstrate the effectiveness of the proposed method.

Index Terms—Edge of criticality; Echo state network; Fisher information; Non-parametric estimation.

ARIMA	Autoregressive integrated moving average
ESN	Echo state network
FIM	Fisher information matrix
GA	Genetic algorithm
MC	Memory capacity
MLLE	Maximum local Lyapunov exponent
MST	Minimum spanning tree
mSVJ	Minimum singular value of Jacobian
NARMA	Non-linear autoregressive moving average
NRMSE	Normalized root mean square error
PD	Positive semidefinite
PDF	Probability density function
RNN	Recurrent neural network
SVR	Support vector regression
ϕ	Region in parameter space where FIM is maximized
λ	Region in parameter space where MLLE crosses zero
η	Region in parameter space where mSVJ is maximized
γ	Prediction accuracy

I. INTRODUCTION

A Recurrent Neural Network (RNN) can approximate any dynamic system under mild hypotheses (see [1] and references therein). However, RNNs are difficult to train [2] and the interpretability of their modus operandi is still object of study [3], [4]. Interestingly, RNNs can generate complex dynamics characterized by sharp transitions permitting them to commute between ordered and chaotic regimes. In fact, experimental results on a multitude of application contexts suggest that RNNs achieve the highest information processing capability exactly when configured on the edge of this transition, resulting in high memory capacity (storage of past inputs) and good performance on the modeling/prediction task at hand (low prediction errors) [5]–[10]. Therefore, in order to determine such “critical” network configurations, RNNs require fine tuning of their controlling parameters. This general behavior is in agreement with the widely-discussed “criticality hypothesis” associated with the functioning of many biological (complex) systems [11]–[16], including the brain [17]–[23]. In fact, it was noted, e.g., see [15], that such complex systems tend to self-organize so as to operate in a critical regime. This still controversial issue has been supported by experiments showing that, in such a regime, systems are highly responsive to external stimuli and hence capable of introducing any dynamics as requested by the specific task [15]. Investigating whether a given complex system operates more efficiently in the critical regime or not requires, at first, theoretically sound methods for detecting the onset of criticality [24].

Determination of system configurations lying on the edge of criticality can be then carried out by means of appropriate sensitivity analyses (on the edge of criticality the sensitivity diverges, being the separation between ordered and chaotic

NOMENCLATURE

Manuscript received ; revised .

Lorenzo Livi is with the Department of Computer Science, College of Engineering, Mathematics and Physical Sciences, University of Exeter, Exeter EX4 4QF, United Kingdom (e-mail: l.livi@exeter.ac.uk).

Filippo Maria Bianchi is with the Machine Learning Group, Dept. of Physics and Technology, University of Tromsø, Tromsø, Norway (e-mail: filippo.m.bianchi@uit.no).

Cesare Alippi is with the Dept. of Electronics, Information, and Bio-engineering, Politecnico di Milano, Milan, Italy and Faculty of Informatics, Università della Svizzera Italiana, Lugano, Switzerland (e-mail: cesare.alippi@polimi.it)

regimes). In this direction, Fisher information, and its multivariate extension called Fisher information matrix (FIM) [25]–[28], provide a way to quantify the sensitivity of a (parametrized) probability density function with respect to its control parameters. Fisher information is tightly linked with statistical mechanics and, in particular, with the field of (continuous) phase transitions. In fact, as shown in [29], it is possible to provide a thermodynamic interpretation of Fisher information in terms of rate of change of the order parameter, quantities used to discriminate the different phases of a system. This fact provides an important bridge between the concept of criticality and statistical modeling of complex systems. It emerges that the critical phase of a thermodynamic system can be mathematically described as that region of the parameter space where the order parameters vanish and their derivatives diverge. This implies that, on the critical region, Fisher information diverges as well, hence providing a quantitative, well-justified tool for detecting the onset of criticality in both theoretical models and computational simulations [30]. Nonetheless, Fisher information is notoriously difficult to compute and, in principle, it requires the knowledge of the analytical form of the parametrized probability density function describing the system behavior.

The designer could consider directly the network weights and drive them towards the edge of criticality through a learning mechanism. Even though this problem is still open at current state of research, what we propose here strongly goes in this direction by accounting a special class of RNNs called Echo State Networks (ESNs) [31]. Although ESNs are typically randomly initialized, the network designer has access to a set of hyperparameters, which have an indirect effect (when considering inputs) on the resulting ESN dynamics and their related computational capability. We define the hyperparameter configurations that bring an ESN in a state where prediction accuracy and memory capacity are maximized as the critical region (or equivalently, edge of criticality). Here we show that the FIM can be used to determine the onset of criticality for a network designed to deal with a particular application. Notably, we provide an unsupervised algorithm that exploits the determinant of the FIM in order to determine the edge of criticality. Since the proposed algorithm is unsupervised, it does not depend on the particular model and related training mechanism adopted for the readout. This feature becomes particularly relevant when the readout layer is implemented by means of non-linear models, such as feed-forward neural networks or kernel-based support vector regression, which require a long training time. In the proposed algorithm, we use a non-parametric FIM estimator [32] that allows us to overcome some of the difficulties that arise when adopting a model-based approach to compute the FIM (e.g., availability of the analytical model ruling the system). Additionally, in order to robustly estimate the FIM, we follow an ensemble approach and perform a number of independent trials.

RNNs, as well as ESNs, are driven by inputs. Therefore, their dynamics and related computational capability depend on the type of input signal driving the network. During the last decade, many solutions have been proposed to characterize the input-driven dynamics of the network and perform

related tuning of the (hyper-)parameters [33]. Among the many contributions, we can cite approaches based on mean-field approximation of the neuron activations [34], information-theoretic methods inspired by the concept of intrinsic plasticity (based on the maximum entropy principle) borrowed from neuroscience [35], [36], and methods for characterizing the onset of criticality with measures of (directional) information transfer and information storage [37], together with related self-organized adaptation mechanisms [38].

To the best of our knowledge, FIM and related thermodynamic interpretations have not been considered yet to study the issue of criticality in ESNs. We stress that, in principle, our method can be extended to account for several hyperparameters, such as feedback scaling and percentage of noise in state update [39]. Finally, it is worth noticing that, as a consequence of the theoretical framework adopted here, we implicitly assume that the critical phase of ESNs can be described by a continuous phase transition. This assumption is highly justifiable, since a system can operate in a critical regime only if such a transition is continuous.

The novelty of our contribution can be summarized as:

- An unsupervised learning method that, by exploiting only the information coming from the neuron activations, permits to identify the edge of criticality. Since no assumption regarding the mathematical model of the (input-driven) dynamic system is made, the method can handle any type of applications;
- The proposed method is independent of the particular reservoir topology, since it is conceived to determine the critical ESN hyperparameters. This allows the network designer to instantiate a specific architecture based on problem-dependent design choices;
- The envisaged non-parametric FIM estimator [32] operates directly on data/observations: as such, there is no need to estimate the high-dimensional densities underlying the neuron activations. As a consequence, the number of reservoir neurons does not pose a serious technical issue from the estimation viewpoint and therefore it can be chosen by the network designer according to application requirements;
- The FIM estimator can be implemented in two different ways, one of which requires elaboration in order to properly define the related optimization problem. In this paper, we propose our own formulation for the constraints defining such an optimization problem – see Appendix A for details.

The remainder of this paper is structured as follows. In Section II, we introduce ESNs and the related considerations on the characterization of the dynamics. Section III introduces Fisher information matrix and the adopted non-parametric estimator. In Section IV, we present the proposed method for determining the ESN hyperparameters. In order to support our methodological developments, Section V presents experimental results performed on both well-known benchmarks and a real-world application involving the prediction of telephone call loads [40]. Conclusions and future research directions follow in Section VI.

II. ECHO STATE NETWORKS

ESNs [31] consist of a large recurrent layer of non-linear units with randomly generated weights and a linear, memoryless readout layer that is trainable by means of a simple regularized least-square optimization. The recurrent layer acts as a non-linear kernel [41], mapping the input to a high-dimensional space. A visual representation of the ESN architecture is reported in Fig. 1.

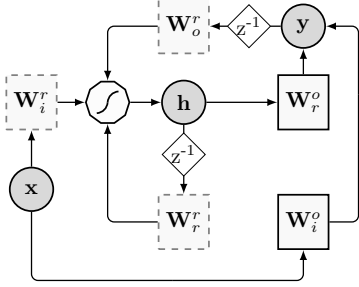


Fig. 1. Schematic depiction of an ESN. The circles represent input \mathbf{x} , state, \mathbf{h} , and output, \mathbf{y} , respectively. Solid squares \mathbf{W}_r^o and \mathbf{W}_i^o , are the trainable matrices, respectively, of the readout, while dashed squares, \mathbf{W}_r^r , \mathbf{W}_o^o , and \mathbf{W}_i^r , are randomly initialized matrices. The polygon represents the non-linear transformation performed by neurons and z^{-1} is the time lag operator.

The equations describing the ESN state-update and output are, respectively, defined as

$$\mathbf{h}[k] = f(\mathbf{W}_r^r \mathbf{h}[k-1] + \mathbf{W}_i^r \mathbf{x}[k] + \mathbf{W}_o^r \mathbf{y}[k-1]), \quad (1)$$

$$\mathbf{y}[k] = \mathbf{W}_i^o \mathbf{x}[k] + \mathbf{W}_r^o \mathbf{h}[k]. \quad (2)$$

The reservoir contains N_r neurons characterized by a transfer/activation function $f(\cdot)$, which is typically implemented as a hyperbolic tangent (*tanh*) function. At time instant k , the network is driven by the input signal $\mathbf{x}[k] \in \mathbb{R}^{N_i}$ and produces the output $\mathbf{y}[k] \in \mathbb{R}^{N_o}$, being N_i and N_o the dimensionality of inputs and outputs, respectively. The vector $\mathbf{h}[k]$ contains N_r components and describes the ESN (instantaneous) state. The weight matrices $\mathbf{W}_r^r \in \mathbb{R}^{N_r \times N_r}$ (reservoir connections), $\mathbf{W}_i^r \in \mathbb{R}^{N_i \times N_r}$ (input-to-reservoir), and $\mathbf{W}_o^r \in \mathbb{R}^{N_o \times N_r}$ (output-to-reservoir feedback) contain real values in the $[-1, 1]$ interval distributed according to a uniform distribution, but additional options have been explored in the recent literature [42], [43]. \mathbf{W}_i^o and \mathbf{W}_r^o , instead, are optimized for the task under consideration, usually by means of a (regularized) linear regression algorithm. Here, for the sake of brevity we do not report the expressions describing training (regularized linear regression of ESN readout) and refer the reader to [31] for details. In fact, the proposed method for finding hyperparameters is completely unsupervised and, hence, independent from the readout training.

The behavior of a given network can be controlled by tuning a set of scalar hyperparameters. Usually, the designer considers θ_{IS} , the scaling of the input weights \mathbf{W}_i^r , hence affecting the non-linearity introduced by the neurons; θ_{SR} , scaling of the spectral radius of \mathbf{W}_r^r , which influences both stability and computational capability of the network by shifting the transfer function poles [35]; θ_{RC} , which determines the sparsity of connectivity in \mathbf{W}_r^r , i.e., the number of weights set to 0; θ_{FB} ,

which affects \mathbf{W}_r^o , that is, the importance of output feedback connections. In this study, we set $\theta_{FB} = 0$ with a consequent simplification of ESN state-update (1):

$$\mathbf{h}[k] = f(\mathbf{W}_r^r \mathbf{h}[k-1] + \theta_{IS} \mathbf{W}_i^r \mathbf{x}[k]), \quad (3)$$

where \mathbf{W}_r^r is normalized as $\mathbf{W}_r^r = \theta_{SR} \mathbf{W}_r^r / \rho(\mathbf{W}_r^r)$, being $\rho(\mathbf{W}_r^r)$ the spectral radius of \mathbf{W}_r^r . θ_{SR} , θ_{IS} , and θ_{RC} are hyperparameters typically tuned through cross-validation to find the best-performing configuration for the task at hand. In this paper, we study how to set these three hyperparameters through an unsupervised approach. However, we stress that the proposed methodology is applicable to any number of hyperparameters.

In order to guarantee asymptotic stability, ESNs must satisfy the so-called echo state property [44]–[47], which requires the reservoir exhibiting short-term memory (exponential fading) [48], [49]. Recently, in [50] the author investigated the effects of criticality in ESN memory, showing that, under certain conditions, the echo state property can still be verified even if the memory vanishes slowly (i.e., following a power-law function).

The stability margin of a network can be assessed in practice by analyzing the Jacobian matrix of the reservoir state update (3). Notably, the maximal local Lyapunov exponent (MLLE) λ , used to approximate the separation rate in phase space of trajectories having very similar initial states [51], can be computed from such a matrix. In autonomous systems, $\lambda < 0$ indicates that the system (here ESN) is stable; $\lambda > 0$ denotes chaoticity. A transition point between those two different behaviors is obtained when $\lambda = 0$. The sign of λ provides thus a criterion for detecting the onset of criticality in reservoirs. Such a criterion is widely used also as a baseline to develop and compare novel criteria [37].

If reservoir neurons are equipped with a hyperbolic tangent activation function, the Jacobian at time k can be conveniently expressed as

$$\mathbf{J}(h[k]) = \begin{bmatrix} 1 - (h_1[k])^2 & 0 & \dots & 0 \\ 0 & 1 - (h_2[k])^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 - (h_{N_r}[k])^2 \end{bmatrix} \mathbf{W}_r^r, \quad (4)$$

where $h_l[k], l = 1, 2, \dots, N_r$, is the activation of the l -th reservoir neuron at time k . λ is then computed by means of the geometric average:

$$\lambda = \max_{n=1, \dots, N_r} \frac{1}{K} \sum_{k=1}^K \log(r_n[k]), \quad (5)$$

where $r_n[k]$ is the absolute value of the n -th eigenvalue of $\mathbf{J}(h[k])$ and K is the total number of samples in the time series under consideration.

Another indicator used to predict the network performance is the minimal singular value of the Jacobian matrix (shortened as mSVJ and denoted in the following as η), which provides accurate information regarding the ESN dynamics. The set of hyperparameter configurations that maximize mSVJ gives rise

to dynamical systems with good excitability, separating well the input signals in state space [51].

In this paper, as a means of numerical comparison with the proposed method based on FIM, we will use also MLLE and mSVJ criteria for detecting the onset of criticality in ESNs.

III. FISHER INFORMATION MATRIX AND THE NON-PARAMETRIC ESTIMATOR

The Fisher information matrix [25] is a symmetric positive semidefinite (PD) matrix whose elements are defined as follows:

$$F_{ij}(p_{\theta}(\cdot)) = \int_{\mathcal{D}} p_{\theta}(\mathbf{u}) \left(\frac{\partial \ln p_{\theta}(\mathbf{u})}{\partial \theta_i} \right) \left(\frac{\partial \ln p_{\theta}(\mathbf{u})}{\partial \theta_j} \right) d\mathbf{u}, \quad (6)$$

where $p_{\theta}(\cdot)$ is a parametric probability density function (PDF), which depends on d parameters $\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_d]^T \in \Theta \subseteq \mathbb{R}^d$ being Θ the parameter space. As will be formally discussed in the following sections, in ESN framework $\boldsymbol{\theta}$ contains the hyperparameters under consideration. In (6), $\ln p_{\theta}(\cdot)$ represents the log-likelihood function. For sake of simplicity, we denote $\mathbf{F}(p_{\theta}(\cdot))$ as $\mathbf{F}(\boldsymbol{\theta})$. The FIM contains $d(d+1)/2$ distinct entries encoding the sensitivity of the PDF with respect to the parameters in $\boldsymbol{\theta}$.

Elements of the FIM can be directly connected with the rate of change of the order parameters of a controlled (thermodynamic) system [29]. An order parameter is a quantity used to discern the phases of a thermodynamic system. For instance, in the liquid–vapor (first-order) transition of water, temperature acts as a control parameter (at constant pressure), while the difference in density of the two phases – liquid and gaseous states – is the order parameter. At the critical temperature, liquid water turns into vapor and the order parameter varies discontinuously. The mathematical relationship between Fisher information and order parameters is particularly interesting to provide a statistical description of continuous, second-order phase transitions, and, as a consequence, of any complex system approaching a critical transition. In fact, during a continuous phase transition the order parameter varies continuously. Therefore, differently from first-order transitions, a system can reside and operate in such a critical state. A well-known example of continuous phase transition is the ferromagnetic–paramagnetic transition of iron, where magnetization (the order parameter) is non-zero for temperatures lower than the critical (Curie) one and zero otherwise. However, second-order derivatives of the observed thermodynamic variable (or, equivalently for continuous transitions, the first-order derivatives of the order parameter) are discontinuous and divergent in at least one dimension. This implies that Fisher information diverges at criticality for infinite systems, while it is maximized in the finite-size system case [29]. This fact provides a clear mathematical justification explaining why the FIM (6) can be used to detect criticality in complex systems in terms of maximum sensitivity with respect to control parameter changes. Therefore, as we already mentioned, the critical region (edge of criticality) is a region in parameter space where the Fisher information is maximized; hence we assume here to deal with finite-size systems. Fig. 2 provides an intuitive illustration linking criticality and ESNs.

Computation of FIM (6) requires analytical availability of the PDF. However, in many experimental settings either (i) the PDF underlying the observed data is unknown or (ii) the relation linking the variation of control parameters $\boldsymbol{\theta}$ on $p_{\theta}(\cdot)$ is unknown. In a recent paper [32], a non-parametric estimator of the FIM was proposed, which is based on divergence measure

$$D_{\alpha}(p, q) = \frac{1}{4\alpha(1-\alpha)} \int_{\mathcal{D}} \frac{(\alpha p(\mathbf{u})(1-\alpha)q(\mathbf{u}))^2}{\alpha p(\mathbf{u})(1-\alpha)q(\mathbf{u})} d\mathbf{u} - (2\alpha - 1)^2, \quad (7)$$

belonging to the family of f -divergences; $\alpha \in (0, 1)$; $p(\cdot)$ and $q(\cdot)$ are PDFs both supported on \mathcal{D} .

It is well-known [13], [32], [52] that FIM can be approximated by using a proper f -divergence measure computed between the parametric PDF of interest and a perturbed version of it. Notably, by expanding (7) with Taylor up to the second order we obtain:

$$D_{\alpha}(p_{\theta}, p_{\hat{\theta}}) \simeq \frac{1}{2} \mathbf{r}^T \mathbf{F}(\boldsymbol{\theta}) \mathbf{r}, \quad (8)$$

where $\hat{\boldsymbol{\theta}} = \boldsymbol{\theta} + \mathbf{r}$, being $\mathbf{r} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_{d \times d})$ a small normally distributed perturbation vector with standard deviation σ .

Divergence (7) can be computed directly without the need to estimate the PDFs by means of an extension of the Friedman-Rafsky multi-variate two-sample test statistic [53]. The test operates by using two datasets, \mathcal{S}_p and \mathcal{S}_q , each one containing samples extracted from $p(\cdot)$ and $q(\cdot)$, respectively. Theorem 1 in [32] shows that, as the number of samples $n = |\mathcal{S}_p|$ and $m = |\mathcal{S}_q|$ increases asymptotically, we have:

$$1 - \mathcal{C}(\mathcal{S}_p, \mathcal{S}_q) \xrightarrow[n \rightarrow \infty]{a.s.} D_{\alpha}(p, q), \quad (9)$$

where $\mathcal{C}(\mathcal{S}_p, \mathcal{S}_q)$ is the outcome (expected to be normally distributed) of the Friedman-Rafsky test, which basically provides a way to measure the similarity between two datasets. Interestingly, such a test allows to analyze also high-dimensional data, since it makes use of a graph-based data representations (minimum spanning tree).

In the following, for the sake of brevity we omit $\boldsymbol{\theta}$ in most of the equations and refer to the estimated FIM as $\hat{\mathbf{F}}$. [32] proposes two different approaches for estimating the FIM (8). The first one is based on the well-known least-square optimization:

$$\hat{\mathbf{F}}_{\text{hvec}} = (\mathbf{R}^T \mathbf{R})^{-1} \mathbf{R}^T \mathbf{v}_{\theta}, \quad (10)$$

where $\mathbf{v}_{\theta} = [v_{\theta}(\mathbf{r}_1), \dots, v_{\theta}(\mathbf{r}_M)]^T$, with $v_{\theta}(\mathbf{r}_i) = 2D_{\alpha}(p_{\theta}, p_{\hat{\theta}_i})$, $i = 1, \dots, M$, and $D_{\alpha}(\cdot, \cdot)$ is computed by means of the left-hand side of (9). \mathbf{R} is a matrix containing all M perturbation vectors \mathbf{r}_i arranged as column vectors, and $\hat{\mathbf{F}}_{\text{hvec}}$ is the half-vector representation of $\hat{\mathbf{F}}$. Note that a vector representation $\hat{\mathbf{F}}_{\text{vec}}$ of $\hat{\mathbf{F}}$ reads as $[f_{11}, \dots, f_{m1}, f_{12}, \dots, f_{mn}]^T$. Since $\hat{\mathbf{F}}$ is symmetric, it can be represented through the half-vector representation, $\hat{\mathbf{F}}_{\text{hvec}}$, which is obtained by eliminating all superdiagonal elements of $\hat{\mathbf{F}}$ from $\hat{\mathbf{F}}_{\text{vec}}$ [54]. $\hat{\mathbf{F}}_{\text{hvec}}$ in (10) is hence defined as $[\hat{f}_{11}, \dots, \hat{f}_{dd}, \hat{f}_{12}, \dots, \hat{f}_{d(d-1)}]^T$, where the diagonal elements are located in the first components of the vector.

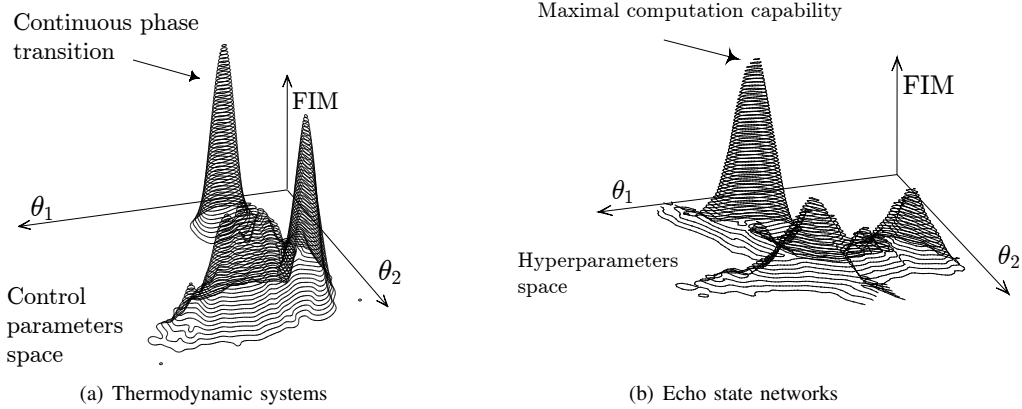


Fig. 2. The approach based on FIM maximization used to identify a continuous phase transition can be adopted also to characterize dynamics in ESNs. In this context, ESN hyperparameters (e.g., spectral radius, input scaling) play the same role of the control parameters in a thermodynamic system (e.g., temperature affects the magnetization phenomenon). FIM can be used to identify the critical region (edge of criticality) in the ESN hyperparameter space, where the computational capability is maximized. Note that the densities plotted in the two figures are not related and show the role played by FIM in the two different fields.

However, the least-square approach (10) does not guarantee to find an approximation of the FIM which is PD. A second approach requires solving a semidefinite optimization problem, which instead assures that the resulting FIM is PD:

$$\begin{aligned}
 & \underset{\mathbf{F}_{\text{hvec}}}{\text{minimize}} && \|\mathbf{R}\mathbf{F}_{\text{hvec}} - \mathbf{v}_\theta\|^2 \\
 & \text{subject to} && \mathbf{F}_{\text{hvec}}(i) = \text{diag}\left(\text{mat}\left(\hat{\mathbf{F}}_{\text{hvec}}\right)\right), \quad i \in \{1, \dots, d\}, \\
 & && \text{mat}\left(\mathbf{F}_{\text{hvec}}\right) \succeq \mathbf{0}_{d \times d}.
 \end{aligned} \tag{11}$$

The $\text{diag}(\cdot)$ operator returns the diagonal elements of a matrix and the $\text{mat}(\cdot)$ operator converts the argument from a vector form into a square $d \times d$ matrix. The diagonal values of the FIM as expressed by the first constraint are computed through the least-square optimization (10). The second constraint, instead, guarantees the estimated matrix to be PD.

Such a convex optimization problem (11) can be implemented by using the framework provided in [55], [56]. However, there, a non-trivial implementation in matrix form of the second constraint, i.e., $\text{mat}(\mathbf{F}_{\text{hvec}}) \succeq \mathbf{0}_{d \times d}$, must be provided to define a proper semidefinite problem. In this paper, we solve this issue and provide a novel method granting $\text{mat}(\mathbf{F}_{\text{hvec}}) \succeq \mathbf{0}_{d \times d}$ (see demonstration in Appendix A).

IV. CRITICAL REGION IDENTIFICATION FOR ESNs

Our goal is to find the edge of criticality, i.e., a region in parameter space $\mathcal{K} \subset \Theta$ where the ESN computational capability is maximized. Fig. 3 shows a schematic description of the main phases involved in the proposed method.

Let us discuss in details the proposed procedure. In order to determine \mathcal{K} , we propose an algorithm exploiting the FIM properties of a system undergoing a continuous phase transition. FIM defines a metric tensor on the smooth manifold of parametric PDFs embedded in Θ [29], allowing thus also for a geometric characterization of the system under analysis. It is possible to prove [57] that \mathcal{K} corresponds to a region in Θ characterized by the largest volume (high concentration of parametric PDFs). In addition, it worth mentioning that,

if $\Theta \subseteq \mathbb{R}^d$, then the related edge of criticality \mathcal{K} is a $d - 1$ manifold embedded in Θ . This geometric result can be exploited by computing the FIM determinant, $\det(\mathbf{F}(\theta))$, which is monotonically related to the aforementioned volume in the parameter space. Therefore, considering that the FIM is a PD matrix, and hence its determinant is always non-negative, we identify \mathcal{K} with all those hyperparameters θ^* for which:

$$\theta^* = \arg \max_{\theta \in \Theta} \det(\mathbf{F}(\theta)). \tag{12}$$

Algorithm 1 delivers the pseudo-code of the proposed procedure. As said before, the impact provided by the variation of the control parameters θ on the resulting ESN state cannot be described analytically without making further assumptions [34]. In fact, the (unknown) input signal driving the network plays an important role in the resulting ESN dynamics. Therefore, in order to calculate $\mathbf{F}(\theta)$, in Algorithm 1 we rely on the non-parametric FIM estimator described in Sec. III. The estimation of the FIM for a given θ is performed by analyzing the sequence $\mathcal{S}_\theta = \{\mathbf{h}[k]\}_{k=1}^K$ of reservoir neuron activations produced during the processing of a given input \mathbf{x} of length K . Since $\mathbf{h}[k] \in [-1, 1]^{N_r}$, the domain of the PDF in (6) is defined as $\mathcal{D} = [-1, 1]^{N_r}$. Additional sequences of activations, $\mathcal{S}_{\hat{\theta}_j}$, are considered (see line 7), which are obtained by perturbing M times the current network configuration θ under analysis, and processing the same input \mathbf{x} . Perturbations are modeled with a zero-mean noise with a spherical covariance matrix, thus characterized by a single scalar parameter σ controlling the magnitude of the perturbation. In this paper, we estimate the FIM by solving the optimization problem (11) according to our formulation as described in Appendix A. In order to make the estimation more robust, we follow an ensemble approach and perform a number of independent trials (see line 3). The determinant is computed only once on the resulting average FIM, which is obtained by using T independent random realizations of the ESN architecture chosen for the experiment (see line 16).

In theory, the parameter space Θ is continuous. However, here we assume that the parameter space Θ is quantized

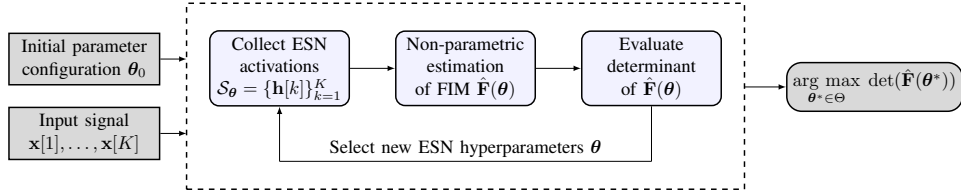


Fig. 3. Schematic, high-level description of the proposed unsupervised learning method.

according to some user-defined resolution. Although this is not a necessary assumption for the proposed methodology, it allows us to disentangle the problems of defining from finding the edge of criticality. In fact, our main goal here is to provide a principled definition of the critical region characterizing the ESN (hyper-)parameter space and related behaviors. More efficient and/or accurate search schemes will be considered in future research studies. Accordingly, the criterion in (12) identifies a “quantized” critical region \mathcal{K} in Θ represented by a single hyperparameter configuration, θ^* .

Algorithm 1 Procedure for determining an ESN configuration on the edge of criticality.

Input: An ESN architecture, input \mathbf{x} of K samples, quantized parameter space Θ , standard deviation σ for the perturbations, number of trials T and perturbations M .

Output: A configuration $\theta^* \in \mathcal{K}$

```

1: Select an initial parameter configuration,  $\theta \in \Theta$ ; maximum  $\eta = 0$ 
2: loop
3:   for  $t = 1$  to  $T$  do
4:     Randomly initialize the ESN weight matrices
5:     Configure ESN with  $\theta$  and process input  $\mathbf{x}$ 
6:     Collect the related activations  $\mathcal{S}_\theta = \{\mathbf{h}[i]\}_{i=1}^K$ 
7:     for  $j = 1$  to  $M$  do
8:       Generate a perturbation vector  $\mathbf{r}_j \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_{d \times d})$ 
9:       Randomly initialize the ESN weight matrices
10:      Configure ESN with perturbed version  $\hat{\theta}_j = \theta + \mathbf{r}_j$  and process
      input  $\mathbf{x}$ 
11:      Collect the related activations  $\mathcal{S}_{\hat{\theta}_j} = \{\mathbf{h}[i]\}_{i=1}^K$ 
12:    end for
13:    Define  $\mathcal{S}_{\hat{\theta}} = \cup_{j=1}^M \mathcal{S}_{\hat{\theta}_j}$ 
14:    Estimate the FIM  $\mathbf{F}^{(t)}(\theta)$  of trial  $t$  using  $\mathcal{S}_\theta$  and  $\mathcal{S}_{\hat{\theta}}$  with the
    non-parametric estimator introduced in Sec. III
15:  end for
16:  Compute the average FIM,  $\mathbf{F}(\theta)$ , using all  $\mathbf{F}^{(t)}(\theta)$ ,  $t = 1, \dots, T$ 
17:  if  $\det(\mathbf{F}(\theta)) > \eta$  then
18:    Update  $\eta = \det(\mathbf{F}(\theta))$  and  $\theta^* = \theta$ 
19:  end if
20:  if Stop criterion is met then
21:    return  $\theta^*$ 
22:  else
23:    Select a new  $\theta \in \Theta$  based on a suitable search scheme
24:  end if
25: end loop

```

A. Analysis of computational complexity

The asymptotic computational complexity (including also constant terms) of Algorithm 1 can be summarized as follows (assuming a grid search):

$$O(G(T(N_r^2 + KN_r + M(N_r^2 + KN_r)) + E_{\text{FIM}}) + d^3 + Td^2). \quad (13)$$

In 13, G is the number of hyperparameter configurations taken into account, T is the number of trials, N_r is the

number of neurons in the reservoir, K is the input signal length, and M is the number of perturbations. The cost related to the computation of the determinant of FIM is hence $O(d^3)$, where d is the number of hyperparameters taken into account. The last term, Td^2 , accounts for the computation of the average FIM. The E_{FIM} term describes the complexity of the non-parametric FIM estimator described in Sec. III. E_{FIM} cost can be decomposed in two different terms. First, the computation of α divergence, has a cost that is given by the MST computation on $z = (M + 1)K$ samples, that is bounded by $O(z^2 \log(z))$. That cost is multiplied by M , the number of perturbations. Second, the cost associated with solving the optimization problem shown in 20. The computational complexity of the constraint satisfaction is bounded by d^2 . The semidefinite optimization program can be solved in polynomial time, i.e., $O(d^p)$, where p is some positive integer [55], [56].

Typically, d is much smaller than both N_r and K . Therefore, polynomial terms in d do not pose a problem from the computational complexity viewpoint. The computation complexity (13) is hence dominated by the E_{FIM} cost.

V. EXPERIMENTS

In this section, we evaluate the effectiveness of the proposed method based on FIM for determining ESN hyperparameter configurations lying on the edge of criticality. The proposed method is firstly validated on a set of benchmarks used in the ESN literature. In particular, we consider the short-term memory capacity (Sec. V-A) and then a forecast task on different time series models (Sec. V-B). For such benchmarks, the training set consists of 5000 samples, while 500 samples are used for testing. Successively, in Sec. V-C we validate the proposed methodology on a real-world application involving the prediction of time series related to phone calls load [40]. Here, the training set consists of 3335 samples, while 500 samples are used for testing.

The hyperparameters are selected in a discretized space through a grid search, which considers 10 different configurations for each parameter. Specifically, we search for the spectral radius θ_{SR} in $[0.4, 1.6]$, input scaling θ_{IS} in $[0.3, 0.8]$, and reservoir connectivity θ_{RC} in $[0.1, 0.7]$, evaluating a total of 1000 hyperparameter configurations. Such intervals have been chosen by focusing on the ranges that produce relevant variations in the network behavior. For each hyperparameter configuration, in Algorithm 1 we perform $T = 10$ independent trials and $M = 80$ perturbations to compute the ensemble average of the FIM; the variance for the perturbations is set to $\sigma^2 = 0.25$. In each trial, we sample new (and independent)

input and reservoir connection weights (W_i^r and W_r^r). The readout layer is trained by using a standard ridge least-square regression, with the regularization parameter set to 0.05. For each test we use a reservoir with $N_r = 100$ neurons; a standard drop-out procedure is adopted [39], which discards the first 100 states not to consider ESN transient.

In Fig. 4, we report the critical regions of the parameter space identified in each test by maximization of FIM determinant, zero-crossing of MLLE, and maximization of mSVJ. For the sake of brevity, we refer to these regions as ϕ , λ , and η , respectively. The light gray manifold corresponds to the regions in parameter space where the performance of the network is maximized and the dark gray manifolds represent ϕ , λ , and η . In Tab. I, we report the numerical values of the correlations between the light gray manifold and the dark gray ones. In the following subsections, we discuss the details of obtained results.

A. Memory capacity

This test quantifies the capability of ESN to remember previous inputs, relative to an i.i.d. signal. Given a time delay $\delta > 0$, here we train an ESN to reproduce at time k the input $\mathbf{x}[k - \delta]$. Memory Capacity (MC) is measured as the squared correlation coefficient between the desired output, which is the input signal delayed by different δ time steps, and the observed network output $\mathbf{y}[k]$:

$$\text{MC} = \sum_{\delta=1}^{\delta_{\max}} \frac{\text{cov}^2(\mathbf{x}[k - \delta], \mathbf{y}[k])}{\text{var}(\mathbf{x}[k - \delta]) \text{var}(\mathbf{y}[k])}. \quad (14)$$

MC is computed by training several readout layers, one for each delay $\delta \in \{1, 10, \dots, 100\}$, while keeping fixed input and reservoir layers.

As it is possible to notice in Fig. 4(a), the critical regions identified by each one of the three methods follow, with good accuracy, the region of the hyperparameter space where MC is maximized. The degrees of correlation for the MC task are provided in Tab. I. It is interesting to note that λ shows a very high correlation (81%) performing better than η for this task. The correlation between ϕ and the region with maximum MC is also very high (75%), showing that both ϕ and λ can be used as reliable indicators to identify the optimal configurations that enhance the short-term memory capacity of ESNs. The p -values for each correlation measure are lower than 0.05, indicating statistical significance of the results.

B. Prediction accuracy on benchmarks

In this test, we evaluate the effectiveness of using ϕ , λ , and η to identify regions of hyperparameters where prediction accuracy is maximal. We define the prediction accuracy as $\gamma = \max\{1 - \text{NRMSE}, 0\}$, where NRMSE is the Normalized Root Mean Squared Error of the ESN. The accuracy is evaluated on three prediction tasks of increasing difficulty. For each of them, we set the forecast step $\tau_f > 0$ equal to the smallest time lag that guarantees input measurements to be decorrelated, which corresponds to the first zero of the autocorrelation function of the input signal.

TABLE I
CORRELATIONS BETWEEN THE REGIONS WHERE FIM DETERMINANT IS MAXIMIZED (ϕ), MLLE CROSSES ZERO (λ), mSVJ IS MAXIMIZED (η) AND PERFORMANCES ARE MAXIMIZED (γ/MC). BEST RESULTS ARE SHOWN IN BOLD, p -VALUES ARE REPORTED IN BRACKETS.

Test	Corr (ϕ , γ/MC)	Corr (λ , γ/MC)	Corr (η , γ/MC)
MC	0.75 (1e-5)	0.81 (1e-8)	0.65 (1e-4)
Predict – SIN	0.58 (0.02)	0.52 (1e-3)	0.56 (1e-3)
Predict – MG	0.71 (1e-5)	0.66 (1e-4)	0.38 (0.06)
Predict – NARMA	0.52 (0.01)	0.25 (0.22)	0.48 (0.02)
Predict – D4D	0.63 (1e-4)	0.34 (0.09)	0.54 (0.01)

Sinusoidal input. In the first test, an ESN is trained to predict a sinusoidal (SIN) input using a forecast step equal to 1/4 of its period. In Fig. 4(b), both ϕ and η are consistent with γ , while λ shows a lower agreement. From Tab. I, we see that ϕ achieves the best results, all the measures have positive degrees of correlation with γ and small p -values (hence statistical significant).

Mackey-Glass. The input signal in this test is generated by the Mackey-Glass (MG) system, described by the following differential equation:

$$\frac{dx}{dk} = \frac{\alpha x(k - \tau_{\text{MG}})}{1 + x(k - \tau_{\text{MG}})^{10}} - \beta x(k). \quad (15)$$

We generated a time series using $\tau_{\text{MG}} = 17$, $\alpha = 0.2$, $\beta = 0.1$, initial condition $x(0) = 1.2$, 0.1 as integration step and we trained the system to predict $\tau_f = 6$ step ahead. As we can see from Fig. 4(c) and the results in the table, for this test both ϕ and λ provide much better results than η for identifying the optimal configuration. Notably, the correlation between γ and η has a p -value beyond the confidence level 0.05, suggesting that correlations are not different from zero.

NARMA. This task, originally proposed in [39], consists in modeling the output of the following order- r system:

$$\mathbf{y}[k + 1] = 0.3\mathbf{y}[k] + 0.05\mathbf{y}[k] \left(\sum_{i=0}^{r-1} \mathbf{y}[k - i] \right) + 1.5\mathbf{x}[k - r]\mathbf{x}[k] + 0.1, \quad (16)$$

being $\mathbf{x}[k]$ an i.i.d. uniform noise in $[0, 1]$. According to the results shown in Fig. 4(d) and Tab. I, in this case ϕ and η perform significantly better than λ for identifying the critical region. In fact, the correlation between γ and λ is low and not statistically significant. Even in this case, the best results in terms of correlation are achieved by ϕ .

C. Prediction of mobile traffic load time series

Here, we analyze time series of data related to nationwide mobile telephone loads. Such time series have been generated from the data collected in the Orange telephone dataset, published in the Data for Development (D4D) challenge [58]. D4D is an open collection of call data records, containing anonymized events of Orange's mobile phone users in Ivory Coast, Africa. More detailed information on the challenge is available on the related web-

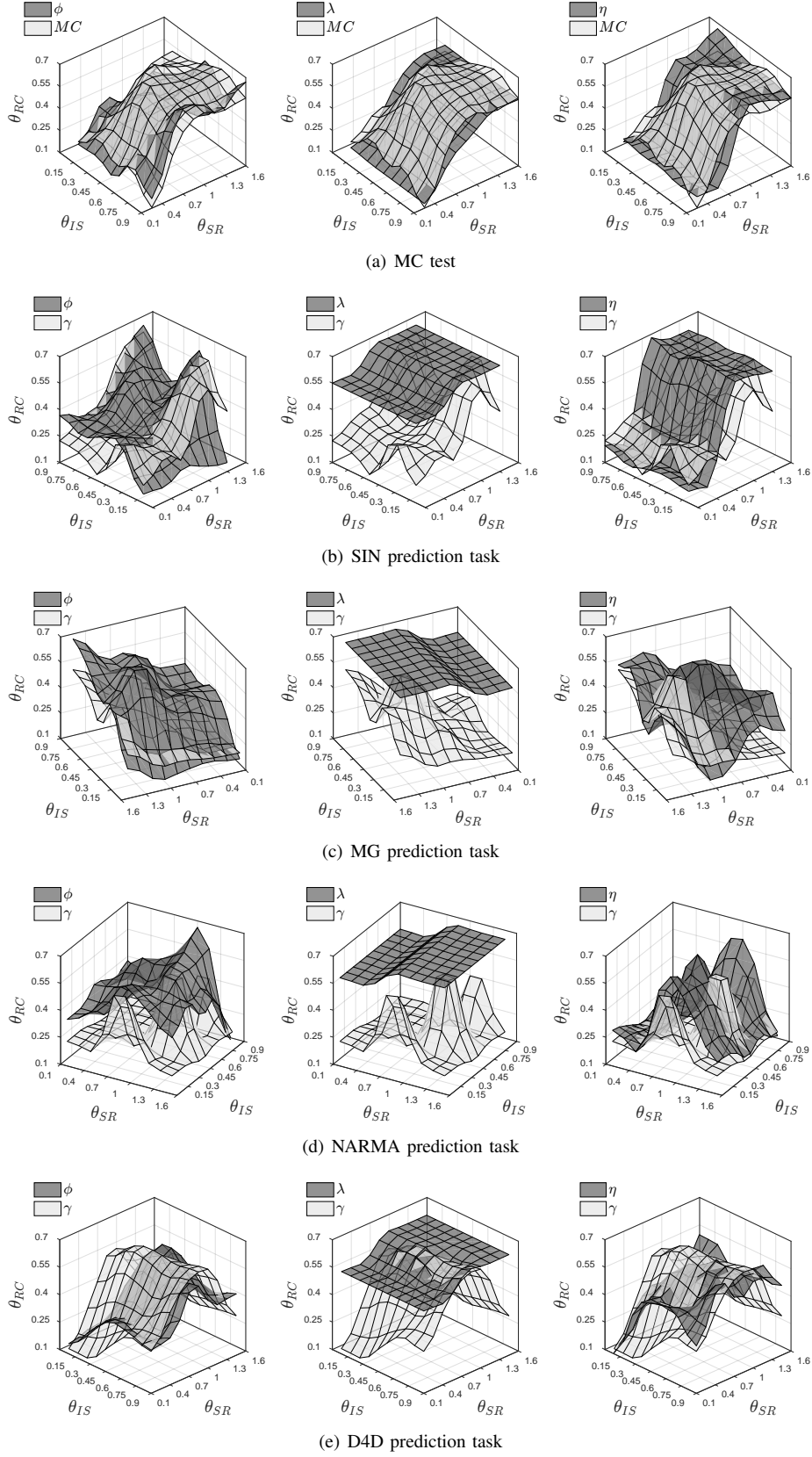


Fig. 4. In each figure, we graphically represent the computed edge of criticality for each of the considered methods. The light gray manifold represents configurations of spectral radius (θ_{SR}), input scaling (θ_{IS}), and reservoir connectivity (θ_{RC}) that maximize Memory Capacity (MC) or prediction accuracy (γ). The dark gray manifolds represent (from left to right): configurations where the FIM determinant is maximized (ϕ); configurations where MLE crosses zero (λ); configurations where mSVJ is maximized (η).

site¹. The dataset considered here span from December 1, 2011 to April 28, 2012. It includes antenna-to-antenna traffic on an hourly basis, relative to mobile phone calls and SMS. Each record in the dataset has the following structure: $\langle \text{DateTime}, \text{ID}_a, \text{ID}_b, \text{NumCalls}, \text{TotTime} \rangle$. DateTime is the time (with hourly resolution) and date when an activity between the two antennas a and b has been registered; ID_a and ID_b are the identifiers of the transmitting and receiving antenna, respectively; NumCalls is the number of calls started from a and received by b in the time interval under consideration; finally TotTime is the sum of the durations (in seconds) of all calls recorded in the interval. We selected a specific antenna and retrieved from the dataset all those records relative to the activity involving that antenna. We have accordingly generated the following 7 distinct time series:

- ts1 : constant input (a time series with all values set to 1). This is a standard practice in prediction with neural networks, since a constant input acts as a bias for the individual neurons of the network [39];
- ts2 : number of incoming calls in the area covered by the antenna;
- ts3 : volume in minutes of the incoming calls in the area covered by the antenna;
- ts4 : number of outgoing calls in the area covered by the antenna;
- ts5 : volume in minutes of the outgoing calls in the area covered by the antenna;
- ts6 : hour of the day when the telephone activity was registered;
- ts7 : day of the week when the telephone activity was registered.

All these 7 time series are fed as input to an ESN, while we predict the values relative only to ts2 .

The dataset contains a small percentage of missing values; they appear when there is no outgoing and/or incoming telephone activities for the monitored antenna at a given hour. They are replaced by 0s, to guarantee each time series to have the same length. Corrupted data are relative to periods where the telephone activity is not correctly registered and, according to [59], they have been replaced with the average value of the corresponding periods (i.e., same weekday and hour of the day) in two adjacent weeks. All data have been standardized by a z-score transformation prior to processing. This is successively reversed when the forecast must be evaluated. In Fig. 5, we show the profile of ts2 relatively to the load in the first 300 time intervals (corresponding to 1 hour of activity).

D4D time series have been previously studied in [40], where ESNs and other standard methods (ARIMA and Triple Exponential Smoothing) were adopted to perform both 1-step and 24-steps ahead predictions. In [40], hyperparameters were tuned using a genetic algorithm (GA) optimization scheme and different optimization procedures were evaluated for training the readout: least-square regression; elastic net penalty; linear and nonlinear ν -SVR. It was shown that ESN achieved higher prediction accuracy with respect to the other forecast methods,

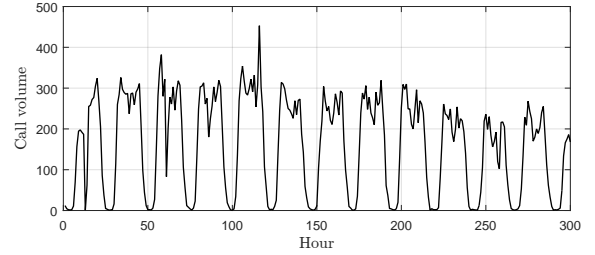


Fig. 5. The load profile of ts2 for the first 300 time intervals.

especially when using non-linear ν -SVR, at the expense of a much higher computational cost. It is worth mentioning that, when GAs are used in a supervised cross-validation scheme, they yield a single solution through a black-box process, which does not follow a mathematically motivated criterion to determine the edge of criticality. Additionally, a non-linear model such as ν -SVR, must be trained at each iteration of the global optimization procedure, with consequent increment of computational complexity.

We focus again on the analysis of the hyperparameters θ_{SR} , θ_{IS} , and θ_{RC} , while for the remaining ones we adopt the optimal configuration found in [40]. In particular, we set $N_r = 680$ and the regularization parameter for the ridge regression equal to 0.04. The forecast step τ_f is set to 1, i.e., we predict the telephonic load of the next hour. Note that this differs from the other tasks previously considered, where τ_f was set equal to the first zero of the autocorrelation function.

In Fig. 4(e), we show the results of γ (prediction accuracy on ts2) with respect to ϕ , λ , and η . According to Tab. I, even in this case ϕ gives rise to a manifold having high correlation with γ ; η produces lower yet positive and statistically significant correlations; finally, λ achieves the worst results. Interestingly, by using the FIM-based criterion we find a critical region in the three-dimensional ESN hyperparameter space containing the optimal values for θ_{SR} and θ_{IS} , which are reported in [40]. In fact, the quantized area centred in $\{\theta_{SR} = 1, \theta_{IS} = 0.35, \theta_{RC} = 0.55\}$, belongs to the ESN critical region, according to the FIM-based criterion. Such region contains also the values $\theta_{SR} = 0.98$ and $\theta_{IS} = 0.33$, which were identified as optimal in [40]. Instead, for what concerns the sparsity of reservoir connectivity θ_{RC} , the upperbound considered in the GA optimization for this value was set to 0.4. Accordingly, the optimal configuration with $\theta_{RC} = 0.55$ could not be obtained. This represents a good example of how the proposed method provides a more flexible approach for tuning the network hyperparameters. In fact, while several cross-validation methods (in fact, using a GA is just an example) treat the model as a black-box, the framework we developed allows to analyze the dynamics of the systems and visualize the critical regions during the process of optimizing the network. For example, it is possible to assess if the critical region is too close to one of the bounds considered for a given hyperparameter, hence allowing to redefine the bounds accordingly.

¹<http://www.d4d.orange.com>

VI. CONCLUSIONS AND FUTURE DIRECTIONS

Echo state networks, as a class of networks in reservoir computing, offer a compromise between training time and network performance in terms of prediction error and short-term memory capacity. Experiments showed that such networks operate more efficiently when configured on the so-called edge of criticality, a region in hyperparameter space separating ordered and chaotic regimes. Hyperparameters (indirectly) affecting the behavior of the network are hence tuned according to some criterion. In this paper, we proposed a principled approach for configuring an echo state network on the edge of criticality. The proposed method is completely unsupervised and is based on the interplay between the theory of continuous phase transitions and Fisher information. In fact, it is possible to prove that Fisher information diverges on the critical region and hence can be used to determine the onset of criticality. Nonetheless, Fisher information presumes analytic knowledge of the parametric distribution describing the system/network; in addition its computation is known to be difficult and prone to numerical errors. In order to deal with these issues, here we have followed an ensemble estimation approach based on a recently proposed non-parametric Fisher information matrix estimator. Such an estimator is applicable to high-dimensional densities, since it operates by means of a graph-based data representation. This last aspect is very important in our case, since we analyze the network through a multivariate sequence of reservoir neuron activations.

We evaluated the proposed method on well-known benchmarks as well as on a real-world application involving telephone call load prediction. The benchmarks taken into account were conceived to evaluate both the short-term memory capacity (in terms of the squared correlation between past inputs and network outputs) and the prediction accuracy (in terms of normalized root mean square error). In order to compare our method with other unsupervised approaches, we have taken into account (i) a criterion based on the sign of the maximum local Lyapunov exponent computed on the activations and (ii) a criterion based on the maximum value of the minimum singular value of the Jacobian matrix of the reservoir. Results showed that the proposed method based on Fisher information is more accurate than those two unsupervised methods (on both the benchmarks and the real-world application) in determining critical ESN hyperparameter configurations in terms of accuracy. However, test on memory capacity showed that Lyapunov exponent provides better results, although differences are not as high as in the accuracy case.

The methodology proposed here offers a sound and appealing solution to determine the onset of criticality in echo state networks, with potential extension to other families of recurrent neural networks. Nonetheless, our contribution comes with some technical difficulties that we have only partially solved so far. First of all, potential non-stationarities and dependencies of the neuron activations might affect the estimation outcomes. Here, we have addressed this issue by following an ensemble approach to estimate the Fisher information matrix. However, other approaches might be considered in the future, for instance by following a window-

based analysis of the activations. Second, the non-parametric Fisher information matrix estimator we used requires to set a parameter controlling the magnitude of the perturbations. This parameter turned out to be very sensitive and difficult to determine in practice, hence posing some technical limitations when trying to automatize the procedure. Such issues will be object of future research efforts.

There are many possible routes that we intend to follow in future research studies. Among the many, we believe it is worth focusing on (i) how to enable and control the output feedback connections and (ii) the application of the proposed unsupervised learning method for training other types of recurrent neural networks.

ACKNOWLEDGEMENTS

We wish to thank Prof. Visar Berisha and Prof. Alfred Olivier Hero III for providing us with part of the code required to implement the non-parametric Fisher information matrix estimator used in this work.

APPENDIX A PROPOSED FORMULATION OF THE SEMIDEFINITE CONSTRAINT

Here we provide the details of the formulation in matrix form of the $\text{mat}(\cdot)$ operator in Eq. 11. This is a necessary step to implement the semidefinite constraint in matrix form.

First, we express the constraint with $\text{mat}(\cdot)$ using the inverse operator, $\text{vec}(\cdot)$, which transforms a matrix into its vector representation. A matrix $\mathbf{F} \in \mathbb{R}^{m \times n}$ is converted into the vector representation as follows:

$$\mathbf{F}_{\text{vec}} = \sum_{i=1}^n \mathbf{B}_i \mathbf{F} \mathbf{E}_i, \quad (17)$$

where \mathbf{E}_i is the i -th canonical basis vector of an n -dimensional Euclidean space, i.e., $\mathbf{E}_i = [0, \dots, 0, 1, 0, \dots, 0]^T$ has a 1 in the i -th position and 0 elsewhere. \mathbf{B}_i is a $(mn) \times m$ block matrix defined as a stack of n blocks, which are defined as $m \times m$ zero-matrix with the exception of the i -th block, which is the identity matrix:

$$\mathbf{B}_i = [\mathbf{0}_{m \times m}, \dots, \mathbf{0}_{m \times m}, \mathbb{I}_{m \times m}, \mathbf{0}_{m \times m}, \dots, \mathbf{0}_{m \times m}]^T. \quad (18)$$

Notice that, in our case, $m = n = d$, where d is the number of ESN hyperparameters taken into account. To convert the half-vector representation \mathbf{F}_{hvec} in Eq. 11 into the vector form \mathbf{F}_{vec} , we rely on the following expression:

$$\mathbf{D}(\mathbf{S}\mathbf{F}_{\text{hvec}}) = \mathbf{F}_{\text{vec}}, \quad (19)$$

where \mathbf{D} and \mathbf{S} are the multiplication and the shuffling matrices, respectively. These matrices cannot be expressed in closed-form [54]. Therefore, in the following we provide the pseudo-code of Algorithms 2 and 3 that implement them.

Algorithm 2 Duplication matrix computation.**Input:** Dimensionality d of the hyperparameter space**Output:** Duplication matrix, \mathbf{D}

```

1:  $\mathbf{D} = \mathbb{I}_{d^2 \times d^2} = [\mathbf{d}_1, \dots, \mathbf{d}_{d^2}]^T$ 
2:  $\gamma = \delta = \emptyset$ 
3: for  $i = 1, \dots, d - 1$  do
4:    $\gamma \cup \{i + di, \dots, i + d(d - 1)\}$ 
5:    $\delta \cup \{i + d(i - 1) + 1, \dots, i + d(i - 1) + d - i\}$ 
6: end for
7: for  $i = 1, \dots, d(d - 1)/2$  do
8:    $\mathbf{d}_{\delta(i)} = \mathbf{d}_{\gamma(i)} + \mathbf{d}_{\delta(i)}$ 
9: end for
10: for  $i = 1, \dots, d(d - 1)/2$ 
11:   remove  $\mathbf{d}_{\gamma(i)}$  from  $\mathbf{D}$ 
12: end for

```

Algorithm 3 Shuffling matrix computation.**Input:** Dimensionality d of the hyperparameter space**Output:** Shuffling matrix, \mathbf{S}

```

1:  $\mathbf{S} = \mathbf{0}_{d(d+1)/2 \times d(d+1)/2} = [\mathbf{s}_1, \dots, \mathbf{s}_{d(d+1)/2}]^T$ 
2:  $\mathbf{I} = \mathbb{I}_{d(d+1)/2 \times d(d+1)/2} = [\mathbf{i}_1, \dots, \mathbf{i}_{d(d+1)/2}]^T$ 
3:  $\mathbf{s}_1 = \mathbf{i}_1$ 
4: for  $j = 2, \dots, d$  do
5:    $\gamma = 1 + d(j - 1) - (j - 1)(j - 2)/2$ 
6:    $\mathbf{s}_j = \mathbf{i}_\gamma$ 
7:   remove  $\mathbf{i}_\gamma$  from  $\mathcal{I}$ 
8: end for
9: for  $j = d + 1, \dots, d(d + 1)/2$  do
10:    $\mathbf{s}_j = \mathbf{i}_j$ 
11: end for

```

Hence, the optimization problem in (11) is formalized as:

$$\begin{aligned}
& \underset{\mathbf{F}_{\text{hvec}}}{\text{minimize}} && \|\mathbf{R}\mathbf{F}_{\text{hvec}} - \mathbf{v}_\theta\|^2 \\
& \text{subject to} && \mathbf{F}_{\text{hvec}}(i) = \hat{\mathbf{F}}_{\text{hvec}}(i), \quad i \in \{1, \dots, d\}, \\
& && \mathbf{D}(\mathbf{S}\mathbf{F}_{\text{hvec}}) = \mathbf{F}_{\text{vec}}, \\
& && \mathbf{F}_{\text{vec}} = \sum_{i=1}^d \mathbf{B}_i \mathbf{F} \mathbf{E}_i, \\
& && \mathbf{F} \succeq \mathbf{0}_{d \times d}.
\end{aligned} \tag{20}$$

REFERENCES

- [1] W. Maass, P. Joshi, and E. D. Sontag, "Computational aspects of feedback in neural circuits," *PLoS Computational Biology*, vol. 3, no. 1, p. e165, 2007.
- [2] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," *arXiv preprint arXiv:1211.5063*, 2012.
- [3] D. Sussillo and O. Barak, "Opening the black box: Low-dimensional dynamics in high-dimensional recurrent neural networks," *Neural Computation*, vol. 25, no. 3, pp. 626–649, 2013.
- [4] F. Bianchi, L. Livi, and C. Alippi, "Investigating echo state networks dynamics by means of recurrence analysis," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–13, 2016.
- [5] R. Legenstein and W. Maass, "Edge of chaos and prediction of computational performance for neural circuit models," *Neural Networks*, vol. 20, no. 3, pp. 323–334, 2007.
- [6] N. Bertschinger and T. Natschläger, "Real-time computation at the edge of chaos in recurrent neural networks," *Neural Computation*, vol. 16, no. 7, pp. 1413–1436, 2004.
- [7] T. Toyozumi and L. F. Abbott, "Beyond the edge of chaos: Amplification and temporal integration by recurrent networks in the chaotic regime," *Physical Review E*, vol. 84, no. 5, p. 051908, 2011.
- [8] K. Rajan, L. F. Abbott, and H. Sompolinsky, "Stimulus-dependent suppression of chaos in recurrent neural networks," *Physical Review E*, vol. 82, no. 1, p. 011903, 2010.
- [9] J. Aljadeff, M. Stern, and T. Sharpee, "Transition to chaos in random networks with cell-type-specific connectivity," *Physical Review Letters*, vol. 114, p. 088101, Feb. 2015.
- [10] G. Wainrib and J. Touboul, "Topological and dynamical complexity of random neural networks," *Physical Review Letters*, vol. 110, p. 118101, Mar. 2013.
- [11] M. Scheffer, J. Bascompte, W. A. Brock, V. Brovkin, S. R. Carpenter, V. Dakos, H. Held, E. H. Van Nes, M. Rietkerk, and G. Sugihara, "Early-warning signals for critical transitions," *Nature*, vol. 461, no. 7260, pp. 53–59, 2009.
- [12] G. Tkačik and W. Bialek, "Information processing in living systems," *Annual Review of Condensed Matter Physics*, vol. 7, no. 1, pp. 89–117, 2016.
- [13] J. Hidalgo, J. Grilli, S. Suweis, M. A. Muñoz, J. R. Banavar, and A. Maritan, "Information-based fitness and the emergence of criticality in living systems," *Proceedings of the National Academy of Sciences*, vol. 111, no. 28, pp. 10095–10100, 2014.
- [14] P. Grigolini, "Emergence of biological complexity: Criticality, renewal and memory," *Chaos, Solitons & Fractals*, 2015.
- [15] T. Mora and W. Bialek, "Are biological systems poised at criticality?" *Journal of Statistical Physics*, vol. 144, no. 2, pp. 268–302, 2011.
- [16] A. Roli, M. Villani, A. Filisetti, and R. Serra, "Dynamical criticality: overview and open questions," *arXiv preprint arXiv:1512.05259*, 2015.
- [17] J. J. Torres and J. Marro, "Brain performance versus phase transitions," *Scientific Reports*, vol. 5, 2015.
- [18] L. De Arcangelis, F. Lombardi, and H. J. Herrmann, "Criticality in the brain," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2014, no. 3, p. P03026, 2014.
- [19] J. Hesse and T. Gross, "Self-organized criticality as a fundamental property of neural systems," *Frontiers in Systems Neuroscience*, vol. 8, p. 166, 2014.
- [20] T. Mora, S. Deny, and O. Marre, "Dynamical criticality in the collective activity of a population of retinal neurons," *Physical Review Letters*, vol. 114, no. 7, p. 078105, 2015.
- [21] P. Massobrio, L. de Arcangelis, V. Pasquale, H. J. Jensen, and D. Plenz, "Criticality as a signature of healthy neural systems," *Frontiers in Systems Neuroscience*, vol. 9, p. 22, 2015.
- [22] G. Tkačik, T. Mora, O. Marre, D. Amodei, S. E. Palmer, M. J. Berry, and W. Bialek, "Thermodynamics and signatures of criticality in a network of neurons," *Proceedings of the National Academy of Sciences*, vol. 112, no. 37, pp. 11508–11513, 2015.
- [23] P. Moretti and M. A. Muñoz, "Griffiths phases and the stretching of criticality in brain networks," *Nature Communications*, vol. 4, 2013.
- [24] M. Scheffer, S. R. Carpenter, T. M. Lenton, J. Bascompte, W. Brock, V. Dakos, J. van De Koppel, I. A. van De Leemput, S. A. Levin, E. H. van Nes, M. Pascual, and J. Vandermeer, "Anticipating critical transitions," *Science*, vol. 338, no. 6105, pp. 344–348, 2012.
- [25] P. Zegers, "Fisher information properties," *Entropy*, vol. 17, no. 7, pp. 4918–4939, 2015.
- [26] T. Toyozumi, K. Aihara, and S-I. Amari, "Fisher information for spike-based population decoding," *Physical Review Letters*, vol. 97, no. 9, p. 098102, 2006.
- [27] J. Beck, V. R. Bejjanki, and A. Pouget, "Insights from a simple expression for linear Fisher information in a recurrently connected population of spiking neurons," *Neural Computation*, vol. 23, no. 6, pp. 1484–1502, 2011.
- [28] M. K. Transtrum, B. B. Machta, K. S. Brown, B. C. Daniels, C. R. Myers, and J. P. Sethna, "Perspective: Slowness and emergent theories in physics, biology, and beyond," *The Journal of Chemical Physics*, vol. 143, no. 1, p. 010901, 2015.
- [29] M. Prokopenko, J. T. Lizier, O. Obst, and X. R. Wang, "Relating Fisher information to order parameters," *Physical Review E*, vol. 84, no. 4, p. 041116, 2011.
- [30] X. Wang, J. Lizier, and M. Prokopenko, "Fisher information at the edge of chaos in random boolean networks," *Artificial Life*, vol. 17, no. 4, pp. 315–329, Oct. 2011.
- [31] M. Lukoševičius and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," *Computer Science Review*, vol. 3, no. 3, pp. 127–149, 2009.
- [32] V. Berisha and A. O. Hero III, "Empirical non-parametric estimation of the Fisher information," *IEEE Signal Processing Letters*, vol. 22, no. 7, pp. 988–992, Jul. 2015.
- [33] O. Obst and J. Boedecker, "Guided self-organization of input-driven recurrent neural networks," in *Guided Self-Organization: Inception*, M. Prokopenko, Ed. Heidelberg, Germany: Springer Berlin, 2014, pp. 319–340.
- [34] M. Massar and S. Massar, "Mean-field theory of echo state networks," *Physical Review E*, vol. 87, no. 4, p. 042809, 2013.
- [35] M. C. Ozturk, D. Xu, and J. C. Principe, "Analysis and design of echo state networks," *Neural Computation*, vol. 19, no. 1, pp. 111–138, 2007.

- [36] B. Schrauwen, M. Wardermann, D. Verstraeten, J. J. Steil, and D. Stroobandt, "Improving reservoirs using intrinsic plasticity," *Neurocomputing*, vol. 71, no. 7, pp. 1159–1171, 2008.
- [37] J. Boedecker, O. Obst, J. T. Lizier, N. M. Mayer, and M. Asada, "Information processing in echo state networks at the edge of chaos," *Theory in Biosciences*, vol. 131, no. 3, pp. 205–213, 2012.
- [38] O. Obst, J. Boedecker, and M. Asada, "Improving recurrent neural network performance using transfer entropy," in *Neural Information Processing. Models and Applications*. Springer Berlin Heidelberg, 2010, vol. 6444, pp. 193–200.
- [39] H. Jaeger, "Adaptive nonlinear system identification with echo state networks," in *Advances in Neural Information Processing Systems*, S. Becker, S. Thrun, and K. Obermayer, Eds. MIT Press, 2002, pp. 593–600.
- [40] F. M. Bianchi, S. Scardapane, A. Uncini, A. Rizzi, and A. Sadeghian, "Prediction of telephone calls load using echo state network with exogenous variables," *Neural Networks*, vol. 71, pp. 204–213, 2015.
- [41] M. Hermans and B. Schrauwen, "Recurrent kernel machines: Computing with infinite echo state networks," *Neural Computation*, vol. 24, no. 1, pp. 104–133, 2012.
- [42] A. Rodan and P. Tiño, "Minimum complexity echo state network," *IEEE Transactions on Neural Networks*, vol. 22, no. 1, pp. 131–144, Nov. 2011.
- [43] L. Appeltant, M. C. Soriano, G. Van der Sande, J. Danckaert, S. Massar, J. Dambre, B. Schrauwen, C. R. Mirasso, and I. Fischer, "Information processing using a single dynamical node as complex system," *Nature Communications*, vol. 2, p. 468, 2011.
- [44] K. Caluwaerts, F. Wyffels, S. Dieleman, and B. Schrauwen, "The spectral radius remains a valid indicator of the echo state property for large reservoirs," in *Proceedings of the IEEE International Joint Conference on Neural Networks*. Dallas, TX, USA: IEEE, Aug. 2013, pp. 1–6.
- [45] M. Buehner and P. Young, "A tighter bound for the echo state property," *IEEE Transactions on Neural Networks*, vol. 17, no. 3, pp. 820–824, May 2006.
- [46] G. Manjunath and H. Jaeger, "Echo state property linked to an input: Exploring a fundamental characteristic of recurrent neural networks," *Neural Computation*, vol. 25, no. 3, pp. 671–696, 2013.
- [47] I. B. Yildiz, H. Jaeger, and S. J. Kiebel, "Re-visiting the echo state property," *Neural Networks*, vol. 35, pp. 1–9, 2012.
- [48] P. Tiño and A. Rodan, "Short term memory in input-driven linear dynamical systems," *Neurocomputing*, vol. 112, pp. 58–63, 2013.
- [49] S. Ganguli, D. Huh, and H. Sompolinsky, "Memory traces in dynamical systems," *Proceedings of the National Academy of Sciences*, vol. 105, no. 48, pp. 18970–18975, 2008.
- [50] N. M. Mayer, "Input-anticipating critical reservoirs show power law forgetting of unexpected input events," *Neural Computation*, vol. 27, no. 5, pp. 1102–1119, 2015.
- [51] D. Verstraeten and B. Schrauwen, "On the quantification of dynamics in reservoir computing," in *Artificial Neural Networks-ICANN 2009*. Springer Berlin Heidelberg, 2009, pp. 985–994.
- [52] J. Hidalgo, J. Grilli, S. Suweis, A. Maritan, and M. A. Muñoz, "Cooperation, competition and the emergence of criticality in communities of adaptive systems," *Journal of Statistical Mechanics: Theory and Experiment*, no. 3, p. 033203, 2016.
- [53] J. H. Friedman and L. C. Rafsky, "Multivariate generalizations of the Wald-Wolfowitz and Smirnov two-sample tests," *The Annals of Statistics*, vol. 7, no. 4, pp. 697–717, 1979.
- [54] J. R. Magnus and H. Neudecker, *Matrix Differential Calculus with Applications in Statistics and Econometrics*. New York, NY: John Wiley & Sons, 1995.
- [55] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," Available at: <http://cvxr.com/cvx>, Mar. 2014.
- [56] —, "Graph implementations for nonsmooth convex programs," in *Recent Advances in Learning and Control*, ser. Lecture Notes in Control and Information Sciences, V. Blondel, S. Boyd, and H. Kimura, Eds. Springer-Verlag Limited, 2008, pp. 95–110.
- [57] I. Mastromatteo and M. Marsili, "On the criticality of inferred models," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2011, no. 10, p. P10012, 2011.
- [58] V. D. Blondel, M. Esch, C. Chan, F. Clérot, P. Deville, E. Huens, F. Morlot, Z. Smoreda, and C. Ziemlicki, "Data for Development: the D4D Challenge on Mobile Phone Data," *ArXiv preprint arXiv:1210.0137*, 2012.
- [59] H. Shen and J. Z. Huang, "Analysis of call centre arrival data using singular value decomposition," *Applied Stochastic Models in Business and Industry*, vol. 21, no. 3, pp. 251–263, 2005.



Lorenzo Livi (M'14) received the B.Sc. and M.Sc. degrees from the Department of Computer Science, Sapienza University of Rome, Italy, in 2007 and 2010, respectively, and the Ph.D. degree from the Department of Information Engineering, Electronics, and Telecommunications at the Sapienza University of Rome, in 2014. He has been with the ICT industry during his studies. From January 2014 until April 2016, he was a Post Doctoral Fellow at Ryerson University, Toronto, Canada. From May 2016 until September 2016, he was a Post Doctoral Fellow at the Politecnico di Milano, Italy and Università della Svizzera Italiana, Lugano, Switzerland. Currently, he is a Lecturer (Assistant Professor) in Data Analytics with the Department of Computer Science at the University of Exeter, UK. He is member of the editorial board of Applied Soft Computing (Elsevier) and a regular reviewer for several international journals, including IEEE Transactions on Fuzzy Systems and Information Sciences (Elsevier). He is a member of IEEE. His research interests include computational intelligence methods, time-series analysis and complex dynamical systems, with focused applications in computational biochemistry and biophysics.



Filippo Maria Bianchi received the B.Sc. in Computer Engineering (2009), the M.Sc. in Artificial Intelligence and Robotics (2012) and PhD in Machine Learning (2016) from Sapienza University, Rome. He worked as research assistant at the Computer Science dept. at Ryerson University, Toronto and currently he is a postdoc fellow in Machine Learning Group at UiT, Tromsø. Research interests in machine learning and pattern recognition include graph matching, clustering, classification, time series analysis, reservoir computing and deep learning.



Cesare Alippi (F'06) received the degree in electronic engineering cum laude in 1990 and the PhD in 1995 from Politecnico di Milano, Italy. Currently, he is a Full Professor of information processing systems with the Politecnico di Milano, Italy and Università della Svizzera Italiana, Switzerland. He has been a visiting researcher at UCL (UK), MIT (USA), ESPCI (F), CASIA (RC), A*STAR (SIN). Alippi is an IEEE Fellow, Distinguished lecturer of the IEEE CIS, Member of the Board of Governors of INNS, Vice-President education of IEEE CIS,

Associate editor (AE) of the IEEE Computational Intelligence Magazine, past AE of the IEEE-Trans. Instrumentation and Measurements, IEEE-Trans. Neural Networks, and member and chair of other IEEE committees. In 2004 he received the IEEE Instrumentation and Measurement Society Young Engineer Award; in 2013 he received the IBM Faculty Award. He was awarded the 2016 IEEE TNNLS outstanding paper award. Among the others, Alippi was General chair of the International Joint Conference on Neural Networks (IJCNN) in 2012, Program chair in 2014, Co-Chair in 2011. He was General chair of the IEEE Symposium Series on Computational Intelligence 2014. Current research activity addresses adaptation and learning in non-stationary environments and Intelligence for embedded systems.

Alippi holds 5 patents, has published in 2014 a monograph with Springer on "Intelligence for embedded systems" and (co)-authored more than 200 papers in international journals and conference proceedings.