www.nature.com/scientificreports

SCIENTIFIC REPORTS

OPEN

Multiplex visibility graphs to investigate recurrent neural network dynamics

Received: 27 September 2016
Accepted: 01 February 2017
Published: 10 March 2017

Filippo Maria Bianchi[1], Lorenzo Livi[2], Cesare Alippi[3,4] & Robert Jenssen[1]

A recurrent neural network (RNN) is a universal approximator of dynamical systems, whose performance often depends on sensitive hyperparameters. Tuning them properly may be difficult and, typically, based on a trial-and-error approach. In this work, we adopt a graph-based framework to interpret and characterize internal dynamics of a class of RNNs called echo state networks (ESNs). We design principled unsupervised methods to derive hyperparameters configurations yielding maximal ESN performance, expressed in terms of prediction error and memory capacity. In particular, we propose to model time series generated by each neuron activations with a horizontal visibility graph, whose topological properties have been shown to be related to the underlying system dynamics. Successively, horizontal visibility graphs associated with all neurons become layers of a larger structure called a multiplex. We show that topological properties of such a multiplex reflect important features of ESN dynamics that can be used to guide the tuning of its hyperparamers. Results obtained on several benchmarks and a real-world dataset of telephone call data records show the effectiveness of the proposed methods.

A current research trend aims at investigating complex time-variant systems through graph theory, by considering suitable features associated with vertices and edges[1]. Of particular interest are those systems that also perform a computation when driven by an external input signal. An example is that of artificial RNNs[2–4], which are computational dynamical systems whose link with physics and neurosciences dates back to the '80 with some pioneering works from Jordan[5] and Amit *et al.*[6]. Nowadays, RNNs are gaining renewed interest in neuroscience due to their biological plausibility[7–10] and in computer science and engineering for their modeling ability[11,12]. RNNs are capable to generate complex dynamics and perform inference based on current inputs and internal state, the latter maintaining a vanishing memory of past inputs[13,14].

Let us consider trajectories describing the evolution of a dynamical system in state space, e.g., the space containing all possible system states. As an example, in Fig. 1 we show the trajectories of a dynamical system operating in ordered (left) and chaotic (right) regimes, whose state is defined by the values of variables θ_1, θ_2, and θ_3 at time t. Discriminating between order and chaos is of fundamental importance for investigating the properties of a dynamical system. It emerges that, such properties are manifested in the memory of the dynamical system and in the divergence rate of its state trajectories[15]. Fading memory is a desirable property of a dynamical system, and is characterized by ordered (contractive) dynamics. This is also referred to as the echo state property in the reservoir computing community[16] and ensures that the current state/output of the system depends only on a finite number of past states/inputs[17]. At the same time, a high divergence rate between state trajectories (a property of chaotic dynamics) is also a desirable feature of RNNs. Results show that RNNs operating in a chaotic regime are able to produce meaningful patterns of activity[18] and a balance must be struck in order to meet both properties. As a consequence, a (computational) dynamical system has to operate on the transition between order and chaos, in a region of the controlling parameter space called "edge of criticality". On the edge of criticality, RNN internal dynamics becomes richer, meaning that neuron activations become heterogeneous[15]. Such a diversity both improves memory capacity of RNNs as well as their capability to reproduce complex target dynamics[19,20]. The notion of edge of criticality permeates several complex systems[21–23], including random Boolean networks[24]

[1]Machine Learning Group, Department of Physics and Technology, University of Tromsø, 9019 Tromsø, Norway. [2]Department of Computer Science, College of Engineering, Mathematics and Physical Sciences, University of Exeter, Exeter EX4 4QF, United Kingdom. [3]Department of Electronics, Information, and Bioengineering, Politecnico di Milano, 20133 Milan, Italy. [4]Faculty of Informatics, Universitá della Svizzera Italiana, 6900 Lugano, Switzerland. Correspondence and requests for materials should be addressed to F.M.B. (email: filippo.m.bianchi@uit.no)

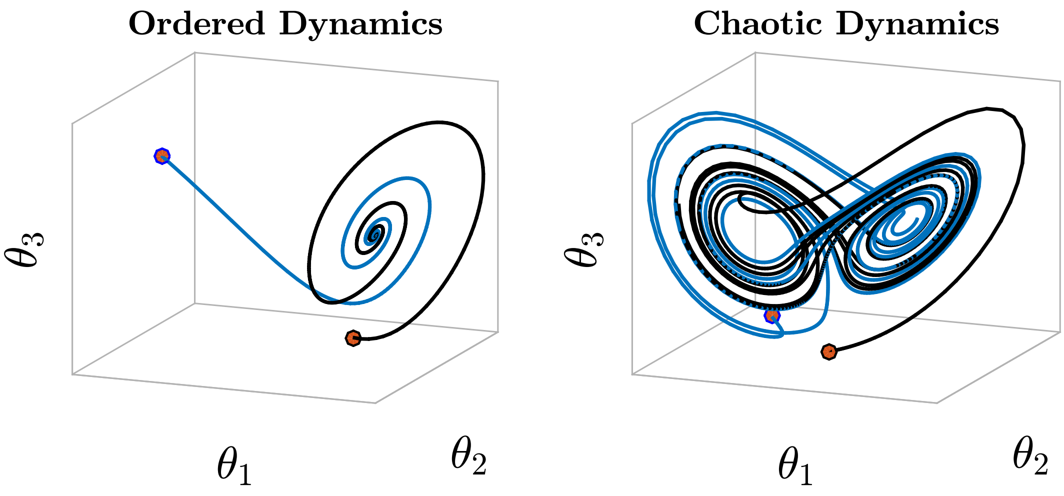


Figure 1. Trajectories in the state space of a system described by the evolution of variables θ_1, θ_2, and θ_3 over time. The behavior of the system is controlled by a parameter (not specified), which we assume to be able to produce ordered (left plot) and chaotic (right plot) dynamics, depending on its value. In the ordered regime, the trajectories converge to the same fixed point when starting from 2 different initial conditions (red dots). In this configuration, the system is characterized by a fading memory of previous states. In the chaotic regime, on the other hand, trajectories remain separated if the system starts from different initial conditions.

and populations of spiking neurons[25]. It is furthermore well-known that several complex systems spontaneously adapt to operate towards the edge of criticality, according to a mechanism known as self-organizing criticality in the statistical physics literature[26].

In this work, we study ESNs[27,28], a class of RNNs characterized by a large, untrained recurrent layer producing outputs by means of a linear (trained) readout layer. As such, ESNs are trained by optimizing readout weights only through a fast linear regression, rather than learning the weights of the recurrent layer by means of backpropagation through time, as in more general RNNs[11]. Please, note that more sophisticated readout layers, requiring longer training procedures, could be used as well[29]. ESNs trade the precision of gradient descent with the "brute force" redundancy of the random recurrent layer. This, inevitably, makes ESNs more sensitive to selection of the hyperparameters that control their internal dynamics. Therefore, the difficulty involved in finding the optimal hyperparameters is more crucial for ESNs than in other RNN models. The non-linear recurrent nature of ESNs hampers interpretability of internal state dynamics[30]. As such, ESNs (as any other RNN) are usually treated as *black box* models with hyperparameters tuned in a supervised manner through cross-validation. However, cross-validation requires evaluation of the computational performance on a validation set for each hyperparameter configuration. This might be a serious limitation in real-life applications, whenever the data are scarce and/or supervised information unavailable. Moreover, the model has to be re-trained for each hyperparameter configuration; this is an issue whenever the learning procedure is computationally demanding[29].

Several unsupervised methods for ESNs tuning have been proposed in the literature, e.g., see refs 31,32. Although unsupervised tuning seldom achieves the same level of accuracy of the supervised counterpart, it offers some insights to the interpretation of the system behavior[33]. Typically, unsupervised learning is achieved by observing the dynamics of the states. These methods are usually based on a statistical framework, which requires reservoir outputs to be independent and identically distributed (i.i.d.). However, internal states are not identically distributed when ESNs are driven by non-stationary inputs[34]. Independence, instead, is always violated both by dependent inputs and by temporal dependencies introduced by recurrent connections. Therefore, even if the input signal is i.i.d., neurons activations may be identically distributed but never independent.

With this work, we propose two unsupervised graph-based methods for tuning ESN hyperparameters, with the aim to maximize (i) prediction accuracy and (ii) memory capacity. In our framework, temporal dependencies in time series of neurons activations (states) are converted into connections of a graph representation. This step allows to relax the i.i.d. assumption of statistical methods. Then, graphs associated with all neurons activations become layers of a structure called a multiplex. A multiplex graph[35–39] is a special type of multilayer network, whose vertices are replicated through each layer and connected across layers only with their replicas. The topology (i.e., the edges) of the graph in each layer can be different. In particular, here we use a multiplex visibility graph[40], whose layers consist of a particular class of graphs called horizontal visibility graphs (HVGs)[41]. HVGs are planar graphs built from real-valued time series, whose elements have one-to-one correspondence with the vertices. Important properties, linking the structure of HVGs with features (e.g., onset of chaotic behavior) of the dynamic system underlying the analyzed time series, have been recently studied[42–46]. This last aspect provides an important justification for using HVGs to model neuron activations for the purpose of hyperparameter tuning.

The novelty introduced in our paper refers to the characterization of ESN dynamics through structural characteristics of the associated multiplex network. To the best of our knowledge, this is the first time ESNs are studied within the framework of multiplex networks. However, it is worth citing a loosely related paper by Zhang *et al.*[47], where the authors set the basis for a graph-theoretical analysis of RNNs. Here, we represent the instantaneous state of an ESN through a set of vertex properties of the multiplex (e.g., degree, clustering coefficient). Edges in HVGs might cover a relevant time interval (contained within the longest period of the signal), while they are still local in terms of topology. Accordingly, the ESN state can be characterized also by information that is non-local in time. To find hyperparameters yielding highest prediction accuracy, we search hyperparameter configurations producing neuron activations as diverse as possible. This occurs when neurons dynamics is maximally heterogeneous (critical dynamics), a characteristic that, as we will show in the paper, is well-captured by the average entropy of vertex properties of the multiplex. Successively, to quantify the amount of memory in an ESN, we check the existence of neuron activations that are "similar" to different delayed versions of the input. In fact, memory in ESNs depends on the ability to reproduce past input sequences from information kept within some neuron activations. We describe dynamics of delayed inputs and of neuron activations through unsupervised graph-based measures. Then, by evaluating the agreement between such measurements, we quantify the memory capacity. We provide experimental evidence that our methods achieve performance comparable with supervised techniques for identifying hyperparamer configurations with high prediction accuracy and large memory capacity.

Methods

Echo state networks.

A standard ESN architecture consists of a large, recurrent layer of non-linear neurons sparsely interconnected by edges with randomly generated weights, called a reservoir, and a linear, feedforward readout layer that is usually trained with regularized least-squares optimization[27]. De facto, the recurrent layer acts as a non-linear kernel[48] that maps inputs to a high-dimensional space. The time-invariant difference equations describing the ESN state-update and output are, respectively, defined as

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

$$\mathbf{y}[t] = g(\mathbf{W}_i^o\mathbf{x}[t] + \mathbf{W}_r^o\mathbf{h}[t]), \tag{2}$$

The reservoir contains N_r neurons whose transfer/activation function $f(\cdot)$ is usually a hyperbolic tangent and $g(\cdot)$ is usually the identity function. At time instant t, the ESN is driven by the input signal $\mathbf{x}[t] \in \mathbb{R}^{N_i}$ and produces the output $\mathbf{y}[t] \in \mathbb{R}^{N_o}$, being N_i and N_o input and output dimensions, respectively. The vector $\mathbf{h}[t] \in \mathbb{R}^{N_r}$ 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 connections), and $\mathbf{W}_o^r \in \mathbb{R}^{N_o \times N_r}$ (output-to-reservoir connections) contain real values randomly sampled from a uniform or Gaussian distribution. Weights in $\mathbf{W}_i^o$ and $\mathbf{W}_r^o$, instead, are trained (typically by means of regularized least-squares algorithm) according to the task at hand.

Once given, the designer can control $\mathbf{W}_r^r$, $\mathbf{W}_o^r$, and $\mathbf{W}_i^r$ only through scaling coefficients. For instance $\mathbf{W}_o^r$ is scaled by a multiplicative constant ω_o, which has the effect to tune the impact of the ESN output on the next state. In this study, the output feedback is removed by setting $\omega_o = 0$. Hence, the resulting state-update difference equation (1) becomes

$$\mathbf{h}[t] = f(\mathbf{W}_r^r\mathbf{h}[t-1] + \mathbf{W}_i^r\mathbf{u}[t]), \tag{3}$$

while the output equation (2) remains unchanged.

The properties of the ESN described in (3) heavily depend on two hyperparameters. First, input weights in $\mathbf{W}_i^r$ are scaled by a multiplicative constant ω_i, which controls the amount of non-linearity introduced by neurons. Large values for ω_i tend to saturate the non-linear activation functions. The second hyperparameter is the spectral radius ρ (eigenvalue with largest absolute value) of $\mathbf{W}_r^r$, which is related to the echo-state property, discussed earlier. For a detailed discussion on the relationships between stability, performance and ρ, we suggest to the interested reader ref. 30 and references therein. Here, it suffices to say that a widely adopted rule-of-thumb[49] suggests to set ρ to a value slightly smaller than 1 (e.g., $\rho = 0.99$). However, to reach higher performance in some practical tasks, it could be necessary to pick a small value for ρ or to breach the aforementioned "safety" bound and push its value beyond unity. Note that in this latter case asymptotic stability of the ESN might still hold, even if some assumptions are locally violated.

In this work, we focus only on the tuning of the hyperparamers ρ and ω_i, without affecting generality of the proposed methods that can be adopted to tune also other hyperparamers. The optimal values of ρ and ω_i for the task at hand are typically identified with a cross-validation procedure, with the potential associated shortcomings mentioned above. For that reason, different unsupervised approaches have been proposed for their tuning[30,31,33]. For example, the effect of ρ and ω_i on the ESN computational capability can be investigated through the maximal local Lyapunov exponent (MLLE), which measures the divergence rate in state space of trajectories with similar initial conditions. In autonomous (not input-driven) systems, chaos occurs when the maximal Lyapuanov exponent becomes positive, while in input-driven systems, like ESN, one typically relies on local first-order approximations of this quantity (see ref. 30 for details). Accordingly, the onset of criticality in ESNs can be detected by checking when MLLE crosses 0. Another quantity, which was shown to be more accurate in detecting criticality in dynamic systems and well-correlated with ESN performance, is the minimal singular value (in average, over time) of the reservoir Jacobian, denoted as λ[33]. λ is unimodal and in correspondence of its maximum the dynamical system is far from singularity, has many degrees of freedom, has a good excitability, and it separates well the input signals in state space[33]. By assuming a null input in Eq. (3), the Jacobian matrix of the reservoir at time t is given by $\mathbf{J}_t = \mathrm{diag}(1-(\mathbf{h}_1[t])^2, 1-(\mathbf{h}_2[t])^2, \ldots, 1-(\mathbf{h}_{N_r}[t])^2)\mathbf{W}_r^r$, where the diag(·) operator returns a diagonal matrix. In this paper, λ will serve as a baseline for comparison to our proposed graph-based unsupervised methods for improved ESN hyperparameter tuning, as discussed in the next section.

Vertex degree	$\phi^{\mathrm{DG}}(v) = \sum_i \mathbf{A}[i, v]$	Number of edges incident to vertex v.
Clustering coefficient	$\phi^{\mathrm{CL}}(v) = \frac{\sum_{i \in \mathcal{C}_v} \sum_{j \in \mathcal{C}_v} \mathbf{A}[i,j]}{\lvert\mathcal{C}_v\rvert \cdot (\lvert\mathcal{C}_v\rvert - 1)}$	Clustering coefficient of vertex v, applicable to both weighted and binary graphs. In ref. 68, $\mathcal{C}_v = \{i \vert \mathbf{A}[i, v] \neq 0 \vee \mathbf{A}[v, i] \neq 0\}$. Here, we consider $\mathcal{C}_v = \{v \cup \mathcal{C}_v\}$ to include also v (and its edges).
Betweenness Centrality	$\phi^{\mathrm{BC}}(v) = \sum_{i \neq v \neq j} \frac{\sigma_{ij}(v)}{\sigma_{ij}}$	Measures the centrality of vertex v. σ_{ij} is the total number of shortest paths from i to j and $\sigma_{ij}(v)$ denotes the number of shortest paths passing through v.
Closeness centrality	$\phi^{\mathrm{CC}}(v) = \sum_{v \neq i} 2^{-\lvert\sigma(i,v)\rvert}$	Total distance of a vertex v from all other vertices in the graph. $\lvert\sigma(i, v)\rvert$ denotes the length of the shortest path between i and v

Table 1. Vertex characteristics used to generate different instances of the vertex property vector ϕ_t.

Horizontal visibility graph and multiplex network. The HVG[41] associated with a finite univariate time series $\mathbf{x} = \{\mathbf{x}[t]\}_{t=1}^{t_{\max}}$, is constructed by assigning a vertex v_t to each datum $\mathbf{x}[t]$. The adjacency matrix $\mathbf{A}$ characterizes the graph: two vertices v_i and v_j, $i \neq j$ are connected by an edge ($\mathbf{A}[i, j] = 1$) iff the corresponding data fulfill the criterion $\mathbf{x}[t_i], \mathbf{x}[t_j] > \mathbf{x}[t_p], \forall\, t_i < t_p < t_j$. In a multivariate scenario, the data stream is composed of N_r different time series $\{\mathbf{x}_l\}_{l=1}^{N_r}$, of equal length $t_{\max}$. A multivariate time series can be mapped into a multiplex visibility graph $\mathcal{M}$ with N_r layers[40]. Specifically, the lth multiplex layer is defined by the HVG G_l constructed from $\mathbf{x}_l$. In the multiplex, a vertex is replicated on all layers and such replicas are linked by inter-layer connections, while intra-layer connections might change in each layer. From now on, we denote $\mathbf{v}_l[t]$ to be the vertex of G_l in layer l associated with time interval t.

In this paper we introduce a weighted HVG (wHVG), with edge values defined as $\mathbf{A}[i, j] = 1/\sqrt{(j - i)^2 + (\mathbf{x}[i] - \mathbf{x}[j])^2} \in [0, 1], \forall\, 1 \leq i, j \leq t_{\max}$. Since self-loops are forbidden in HVGs, edge weights are always well-defined (i.e., finite). The use of weights permits to capture additional information, as it accounts for distance in time ($j - i$) and amplitude differences ($\mathbf{x}[i] - \mathbf{x}[j]$) of two data points connected by the visibility rule. This weighting scheme is motivated by our need to characterize and exploit the instantaneous state by means of a suitable measure of heterogeneity (discussed in the next section). To distinguish the original adjacency matrix from the one of the wHVG, we refer the former as *binary* adjacency matrix and the latter as weighted adjacency matrix. Algorithm 2 delivers the pseudo-code for constructing a HVG (and a wHVG) from time series $\mathbf{x}$. The worst case complexity of this algorithm is $\mathcal{O}(t_{\max}^2)$, which occurs when values in $\mathbf{x}$ are monotonically decreasing. Instead, the best case complexity is $\mathcal{O}(t_{\max})$ and arises in correspondence of monotonically increasing values in $\mathbf{x}$.

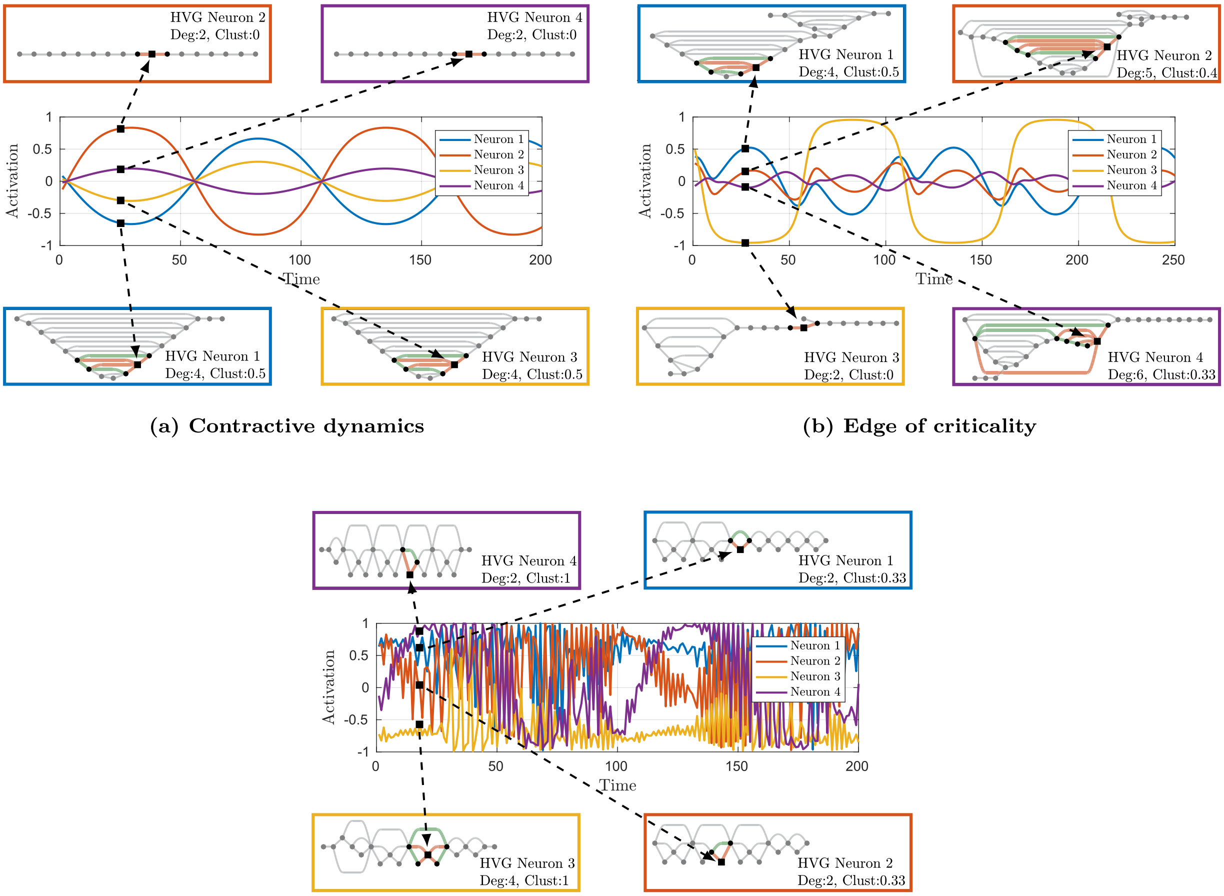
Algorithm 1 Construction of a (weighted) HVG.

```
Input: Time series $\{\mathbf{x}[t]\}_{t=1}^{t_{\max}}$
Output: Adjacency matrix $\mathbf{A}$ of (weighted) HVG
for $i = 1, \ldots, t_{\max} - 1$ do
    Set $j = i + 1$, max $= -\infty$, stop = false, count = 0
    while stop is false AND $j \leq t_{\max}$ do
        count = count + 1
        if $\mathbf{x}[j] >$ max then
            If unweighted, $\mathbf{A}[i, j] = 1$, otherwise, $\mathbf{A}[i, j] = 1/\sqrt{(j - i)^2 + (\mathbf{x}[i] - \mathbf{x}[j])^2}$
            max = $\mathbf{x}[j]$
            if max > $\mathbf{x}[i]$ then
                stop = true
            end if
        end if
        $j = j + 1$
    end while
end for
```

Vertex properties. Let us consider an ESN with a reservoir of size N_r, driven by an input of length $t_{\max}$. According to (3), the ESN generates N_r time series $\{\mathbf{h}_1[t]\}_{t=1}^{t_{\max}}, \ldots, \{\mathbf{h}_{N_r}[t]\}_{t=1}^{t_{\max}}$, of neuron activations (state). The multivariate time series $\{\mathbf{h}_l\}_{l=1}^{N_r}$ is represented here by a multiplex $\mathcal{M}$. Indexes of the vertices on each layer l of $\mathcal{M}$ are associated one-to-one with the time index of the original time series. Hence, the ESN state $\{\mathbf{h}_l[t]\}_{l=1}^{N_r}$ at time t is represented by the vertices $\{\mathbf{v}_l[t]\}_{l=1}^{N_r}$ and by vertex properties $\Phi^*[t] = \{\phi^*(\mathbf{v}_l[t])\}_{l=1}^{N_r}$. In Table 1 we introduce four indexes: vertex degree, clustering coefficient[50], betweenness and closeness centrality[51].

Heterogeneity of neurons dynamics. The capability of an ESN to reproduce the dynamics of a target system, hence to predict its trajectory in state space, is maximized on the edge of criticality, where the internal dynamical patterns of an ESN become sufficiently rich. In the literature, such a "richness" is usually expressed in terms of diversity of connection weights[52], entropy or rank of the matrix of neuron activations[1,5,32].

In the same spirit, in order to find hyperparameters yielding maximal prediction accuracy, here we look for those hyperparameter configurations giving rise to neuron activations that are as heterogeneous as possible. Figure 2 provides a visual example of the concept herein discussed. As an illustration, we consider an ESN driven by a sinusoid. We depict the neuron activations and the corresponding HVGs. We select a time step (marked by



(a) Contractive dynamics

(b) Edge of criticality

(c) Chaos

Figure 2. Heterogeneity of the graph-based representation of the ESN instantaneous state. From each time series, we visualize only a portion of its HVG and consider a specific time step (black square). In the HVGs, we highlight in red the edges connecting each vertex that corresponds to the selected time step. For the clustering coefficient, instead, one has to consider also edges highlighted in green. The non-local nature of ESN graph-based state is evident from such representations, since vertices can be connected to other vertices that are associated to time instants far away along the sequence.

a black square in the picture) and we show the correspondence between the element in each time series of the neuron activations and the vertices in the associated HVGs. When the ESN operates with a contractive dynamic (2a), neuron activations weakly depend on previous ESN states and they are all very similar to the input signal. This results in a lack of diversity among activations. Accordingly, the corresponding HVGs in the multiplex contain vertices with similar properties, e.g., similar degree and clustering coefficient. When the ESN approaches the edge of criticality, neuron activities highly depend on previous internal states, which encode information of past inputs and internal structure of neuron connections. As we can see from (2b), on the edge of criticality, the activations have different frequencies and their phases are shifted. Such a heterogeneity in the ESN instantaneous state is captured by vertex properties of the multiplex. Finally, if pushed beyond the edge of criticality, the ESN transits into a chaotic regime as shown in (2c). In this case, neuron activations become noise-like and disordered oscillations generate HVGs with vertex properties very different from previous configurations. However, the diversity of the patterns in different time series disappears, hence heterogeneity is again lost. This lack of variety is also highlighted by recurrence of similar motifs in the corresponding HVGs.

To determine the heterogeneity of neuron activations, we consider the entropy of the related vertex property distribution. In particular, heterogeneity is computed as follows:

1. At each time t, evaluate the vertex properties $\Phi^*[t] = \{\phi^*(\mathbf{v}_l[t])\}_{l=1}^{N_r}$ using one of the properties listed in Table 1.
2. Estimate distribution $p(\Phi^*[t])$ using a histogram with b bins.
3. Compute instantaneous entropy: $H_t = H(p(\Phi^*[t]))$, where $H(\cdot)$ is the Shannon entropy.
4. Define heterogeneity as average entropy over time instants: $\overline{H} = \frac{1}{t_{\max}}\sum_{t=1}^{t_{\max}} H_t$.

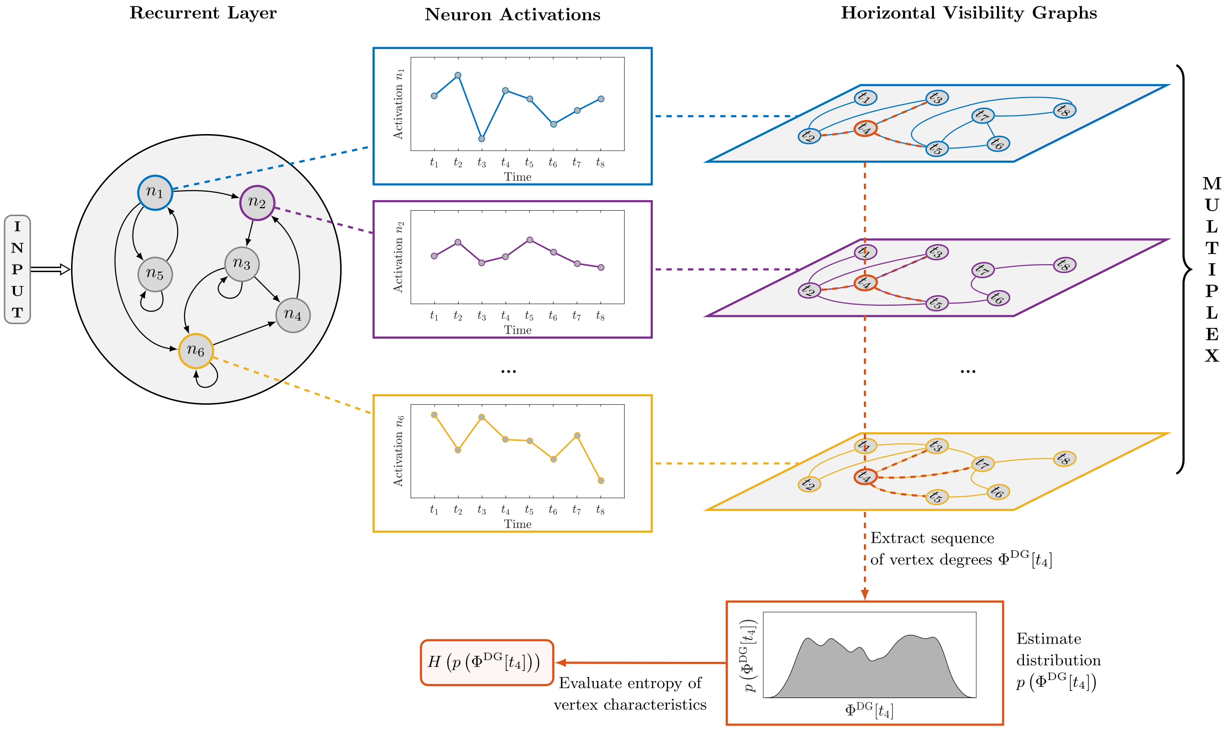

Recurrent Layer
Neuron Activations
Horizontal Visibility Graphs
INPUT
n_1
n_2
n_3
n_4
n_5
n_6
Activation n_1
Activation n_2
Activation n_6
t_1 t_2 t_3 t_4 t_5 t_6 t_7 t_8
Time
...
MULTIPLEX
Extract sequence of vertex degrees $\Phi^{\mathrm{DG}}[t_4]$
Estimate distribution $p(\Phi^{\mathrm{DG}}[t_4])$
$p(\Phi^{\mathrm{DG}}[t_4])$
$\Phi^{\mathrm{DG}}[t_4]$
$H(p(\Phi^{\mathrm{DG}}[t_4]))$
Evaluate entropy of vertex characteristics

Figure 3. High-level illustration of the proposed methodology. The activity of an ESN that is driven by some input signal is analyzed through a multivariate time series of neuron activations. We map such time series with a multiplex formed by HVGs on the layers. In this way, we get rid of problems related to stationarity and dependence of neuron activations. Note that with such a representation, the state of an ESN is defined through graph-theoretical tools, which allows to include non-local (in time) information in the ESN instantaneous state. For instance, at time instant t_4, instead of the original activations, we consider a vertex property (e.g., vertex degrees $\Phi^{\mathrm{DG}}[t_4]$) of the HVG to represent the ESN state. The instantaneous state is eventually characterized by a measure of heterogeneity of such values; here, entropy is taken into account.

The procedure described above, is visually represented in Fig. 3. By referring to the figure, we observe that, at given time t, the ESN state is represented by those vertices in the multiplex tagged with the same label across the layers. For example, the ones in red describe the instantaneous state of the ESN at time step t_4 and a vector of vertex properties $\Phi^{*}[t_4]$ is computed for this set of vertices. Then, we estimate the distribution $p(\Phi^{*}[t_4])$ and compute the entropy $H_4^* = H(p(\Phi^*[t_4]))$. In order to compute $\overline{H}^*$, the procedure is repeated for each time step. Finally, $\overline{H}^*$ is used to characterize the current hyperparameter configuration. The average entropy depends on the specific vertex property chosen for the analysis (we taken into account four properties, e.g., vertex degree, clustering coefficient, betweenness and closeness centrality – see Table 1). These vertex properties lead to four different entropy values $\overline{H}^{\mathrm{DG}}$, $\overline{H}^{\mathrm{CL}}$, $\overline{H}^{\mathrm{BC}}$, and $\overline{H}^{\mathrm{CC}}$.

To summarize, given an input signal, we select hyperparameter configurations that maximize such entropy values. This criterion is inspired by the aforementioned observation linking performance of a computational dynamical system (i.e., prediction accuracy and memory) with heterogeneity of its dynamics (critical dynamics). Accordingly, in this case this information is exploited in order to derive, in an unsupervised way, the configuration yielding highest prediction accuracy.

Other multiplex complexity measures.

Recently[40], two measures have been proposed in order to characterize the dynamics of a system observed through a multivariate time series and represented as a multiplex composed of HVGs. Here, we consider these measures in order to evaluate whether they are useful for identifying the hyperparameter configurations yielding maximum accuracy performances or not.

The Average Edge Overlap (AEO) computes the expected number of layers of the multiplex on which an edge is present. For binary HVGs, it is defined as

$$\mathrm{AEO} = \frac{\sum_i \sum_{j>i} \sum_l \mathbf{A}_l[i, j]}{N_r \sum_i \sum_{j>i} \left(1 - \delta_{0, \sum_l \mathbf{A}_l[i,j]}\right)}, \tag{4}$$

where $\mathbf{A}_l[i, j]$ equals 1 if vertices i and j are connected in layer l. The second measure is the Interlayer Mutual Information (IMI), which quantifies the correlations between the degree distributions of two different layers l_i and l_j. It is defined, for binary HVGs, as

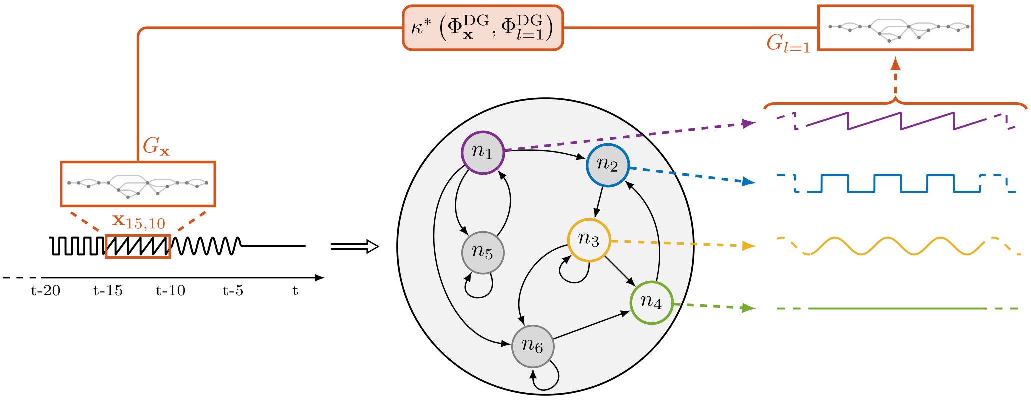


Figure 4. We extract a subsequence from the history of the input signal, for example $\mathbf{x}_{15,10}$, and compute the related HVG $G_{\mathbf{x}}$. We repeat the procedure for each neuron activation, ending up with $G_1, \ldots, G_{N_r}$. For each HVG, we compute the sequence of vertex degrees and evaluate the similarities $\kappa^*(\Phi_{\mathbf{x}}^{\text{DG}}, \Phi_l^{\text{DG}})$ for $l = 1, \ldots, N_r$. The largest similarity value determines the value of δ^{DG}. Here, the activation of neuron n_1 is clearly the most related one with the input sequence $\mathbf{x}_{15,10}$ and therefore it determines the value of δ^{DG}.

$$\text{IMI}(l_i, l_j) = \sum_{k_{l_i}}\sum_{k_{l_j}} p\left(k_{l_i}, k_{l_j}\right)\log\left(\frac{p\left(k_{l_i}, k_{l_j}\right)}{p\left(k_{l_i}\right)p\left(k_{l_j}\right)}\right), \tag{5}$$

where $p\left(k_{l_i}, k_{l_j}\right)$ is the joint probability to find a vertex with degree k_{l_i} in layer l_i and degree k_{l_j} in layer l_j, respectively. In the experiments, we use the average IMI between all pairs of multiplex layers.

Memory measures. The memory of an ESN is quantified by the capability to retain information about past inputs in its transient dynamics[53]. A supervised measure called memory capacity (MC) is usually adopted to quantify ESN memory[27]. When computing MC, the reservoir topology and weights are kept fixed and several readout layers are trained in order to reproduce delayed versions of the input at different time lags $\tau_{L_1}, \ldots, \tau_{L_{\max}}$. The MC is computed as the squared correlation coefficient between desired (delayed input) and computed outputs,

$$\text{MC} = \sum_{\tau_L} \text{cov}^2(\mathbf{x}[t - \tau_L], \mathbf{y}[t])/(\text{var}(\mathbf{x}[t - \tau_L])\text{var}(\mathbf{y}[t])). \tag{6}$$

In order to properly evaluate the capability of ESNs to introduce memory through recurrent connections, $\mathbf{x}[t]$ is defined as a stationary uncorrelated noisy signal.

In the following, we propose an unsupervised graph-based method to identify hyperparameter configurations for which an ESN achieves large memory capacity. Given the input time series $\mathbf{x} = \{x[t]\}_{t=1}^{t_{\max}}$, we determine if there exists a subset of neuron activations, which is correlated with a past input sequence $\mathbf{x}_{a,b} = \{x[t - \tau_a], \ldots, x[t - \tau_b]\}$, with $\tau_a > \tau_b$ and $\tau_a, \tau_b \in [1, t_{\max} - 1]$. Being G_l the HVG representing the lth layer of the multiplex, $\Phi_l^{\text{DG}} = \{\phi^{\text{DG}}(\mathbf{v}_l[t])\}_{t=1}^{t_{\max}}$ is the sequence of its vertex degrees ordered according to the time index (not to be confused with $\Phi^{\text{DG}}[t] = \{\phi^{\text{DG}}(\mathbf{v}_l[t])\}_{l=1}^{N_r}$, the degrees of vertices relative to time t across the different layers). With $G_{\mathbf{x}}$, instead, we refer to the HVG constructed over the input $\mathbf{x}[t]$, while $\Phi_{\mathbf{x}}^{\text{DG}}$ is the vector of its vertex degrees.

First, we define a measure of maximum agreement between $\Phi_{\mathbf{x}}^{\text{DG}}$ and each sequence Φ_l^{DG} as $\delta^{\text{DG}} = \max_l \kappa^*(\Phi_{\mathbf{x}}^{\text{DG}}, \Phi_l^{\text{DG}})$. $\kappa^*(\cdot,\cdot)$ is a similarity measure between sequences: in this paper we consider the Pearson correlation $\kappa^{\text{PC}}(\cdot,\cdot)$, the Spearman correlation $\kappa^{\text{SC}}(\cdot,\cdot)$, and the mutual information $\kappa^{\text{MI}}(\cdot,\cdot)$.

A second measure of agreement is defined on the adjacency matrix $\mathbf{Q}_l = \mathbf{A}_{\mathbf{x}} \wedge \mathbf{A}_l$, where $\mathbf{Q}_l[i, j] = 1$ if $\mathbf{A}_{\mathbf{x}}[i, j] = 1$ and $\mathbf{A}_l[i, j] = 1$; $\mathbf{Q}_l[i, j] = 0$ otherwise. The agreement is then computed as the largest number of edges among all possible intersection graphs, $\delta^{\text{AND}} = \max_l |\mathbf{Q}_l|$, where $|\cdot|$ counts the number of ones in the adjacency matrix.

Finally, one might consider the similarity between original input $\mathbf{x}$ and neuron activations $\mathbf{h}_l$, $\delta^{\text{TS}} = \max_l \kappa^*(\mathbf{x}, \mathbf{h}_l)$. In this case, $\kappa^*(\cdot,\cdot)$ is directly evaluated on the time series rather than on the sequences of vertex degrees. This last measure of agreement is taken into account in order to quantitatively show (in the experiments) the benefits of using HVGs for representing neuron activations.

By referring to the illustrative example in Fig. 4, we generate the HVGs $G_{\mathbf{x}}, G_1, \ldots, G_{N_r}$, relative to delayed input $\mathbf{x}_{15,10}$ and activations of the N_r reservoir neurons, respectively. Successively, we evaluate their similarities by means of an agreement measure (δ^{DG} in this example). The similarity measure κ^* is chosen among the three previously proposed measures: κ^{PC}, κ^{SC}, or κ^{MI}.

Results

In the following, we perform two experiments in order to evaluate the two proposed unsupervised methods for, respectively, finding hyperparameter configurations giving rise to ESNs with high prediction accuracy and large memory capacity. In the first experiment, we show that, on different real and synthetic tasks, (supervised) prediction accuracy is maximized for the same hyperparameter configurations that yields the largest heterogeneity for the vertex properties of the multiplex. In the second experiment, we show the reliability of the graph-based memory measures in identifying hyperparameters where the (supervised) MC is maximized.

Test for prediction accuracy. In this experiment, we consider several prediction tasks and, for each of them, we set the forecast step $\tau_f > 0$ to be the smallest time-lag that guarantees the measurements in a time window of size τ_f to be uncorrelated (e.g., the first zero in the autocorrelation function of the input signal). Prediction error is evaluated by Normalized Root Mean Squared Error, NRMSE $= \sqrt{\langle \|\mathbf{y}[t] - \hat{\mathbf{y}}[t]\|^2 \rangle / \langle \|\hat{\mathbf{y}}[t] - \langle \mathbf{y}[t] \rangle\|^2 \rangle}$, where $\hat{\mathbf{y}}[t]$ is the prediction provided by the ESN and $\mathbf{y}[t]$ is the desired/teacher output. The Prediction accuracy is defined as $\gamma = \max\{1 - \mathrm{NRMSE}, 0\}$.

In the following, we describe the datasets used in this experimental campaign.

Sinusoidal input. We feed an ESN with a sinusoid $y(t) = \sin(\psi t)$ and we predict future input values with a forecast step $\tau_f = 2\pi/\psi$.

Mackey-Glass time series. The Mackey-Glass (MG) system is commonly used as a benchmark in chaotic time series prediction. The input signal is generated from the MG time-delay differential equation

$$\frac{dx}{dt} = \frac{\alpha x(t - \tau_{\mathrm{MG}})}{1 + x(t - \tau_{\mathrm{MG}})^{10}} - \beta x(t). \tag{7}$$

We adopt the standard parameters $\tau_{\mathrm{MG}} = 17$, $\alpha = 0.2$, $\beta = 0.1$, initial condition $x(0) = 1.2$, and integration step equal to 0.1. The forecast step here is $\tau_f = 6$.

Multiple superimposed oscillator. Prediction of superimposed sinusoidal waves with incommensurable frequencies is a hard forecasting exercise, due to the extension of the wavelength[54]. The ESN is fed with the multiple superimposed oscillator (MSO)

$$y(t) = \sin(0.2t) + \sin(0.311t) + \sin(0.42t). \tag{8}$$

For this task, the ESN is trained to predict future input values, with forecast horizon $\tau_f = 16$.

NARMA. The chosen Non-Linear Auto-Regressive Moving Average (NARMA) tas[55] consists in modeling the output of the r-order system:

$$y[t+1] = 0.3y[t] + 0.05y[t]\left(\sum_{i=0}^{r} y[t-i]\right) + 1.5x[t-r]x[t] + 0.1. \tag{9}$$

$x[t]$ is a uniform random noise signal in [0, 1] and is the input of the ESN, which is trained to reproduce $y[t+1]$. The NARMA task is known to require a memory of at least r past time-steps, since the output is determined by inputs and outputs from the last r time-steps. For this task we set $r = 20$ and $\tau_f = 15$.

Polynomial task. The ESN is fed with uniform noise in $[-1, 1]$ and is trained to reproduce the following output

$$y[t] = \sum_{i=0}^{p}\sum_{i=0}^{p-i} c_{i,j} x^i[t] x^j[t-d] \qquad \text{s.t.} \qquad i + j \leq p, \tag{10}$$

where $c_{i,j}$ is uniformly distributed in [0, 1][56]. The difficulty of prediction can be controlled by varying the polynomial degree p and the time delay d. For this task we set $p = 7$, $d = 10$, and $\tau_f = d$.

Telephone call load time series. As a last test, we consider a real-world dataset relative to the load of phone calls registered over a mobile network. The data comes from the Orange telephone dataset, published in the Data for Development (D4D) challenge[57]. D4D is a collection of call data records, containing anonymized events of Orange's mobile phone users in the Ivory Coast, in a period spanning from December 1, 2011 to April 28, 2012. The dataset consists of 6 time series consisting of: number and volume of incoming calls, number and volume of outgoing calls, day and time (1 hour resolution) when the telephone activity was registered. More detailed information is available at the related website[58]. All 6 time series are fed into the ESN as inputs; the goal is to predict 6 hours ahead the volume of incoming calls – the profile of this latter time series is depicted in Fig. 5.

In each test, we evaluate the correlation between the average entropy of vertex properties in the multiplex and the prediction accuracy γ, as we vary the hyperparameters ρ and ω_i. Multiplexes are generated using both the binary and weighted version of the HVG adjacency matrix. To appreciate the effectiveness of our methodology, we also estimate the correlations of γ with λ, the minimum singular value of the Jacobian of the reservoir (see Methods). Additionally, we consider the correlation of γ with the two layer-based measures IMI and AEO (see Methods). Correlations are evaluated as follows. For each configuration ($\rho = k$, $\omega_i = j$), we have the prediction accuracy $\gamma_{k,j}$, the entropy $\overline{H}_{k,j}^{\mathrm{DG}}$, and so on. The values assumed by these quantities by varying ρ and ω_i generate a two-dimensional manifold. The

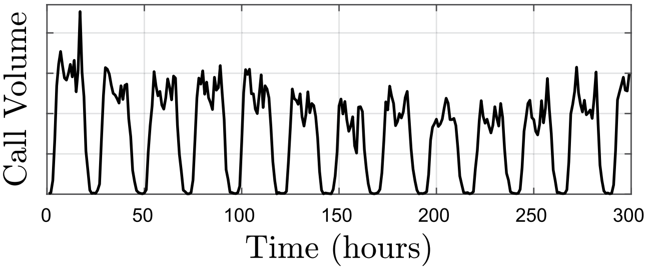


Figure 5. D4D dataset – load profile of incoming calls volume for the first 300 time intervals.

point-wise linear (Pearson) correlation between the manifolds relative to γ and the other considered measures is the result we are interested in. The configurations of the hyperparameters that are examined are generated by varying ρ in [0.5, 1.3] (20 different values) and ω_i in [0.2, 0.9] (10 different values). A total of 200 configurations are evaluated. Due to the stochastic nature of the ESN initialization, for each configuration ($\rho = k$, $\omega_i = j$) we compute $\gamma_{k,j}$ and all the other measures 15 different times. Successively, we compute the correlations among their average values. We use a reservoir with $N_r = 100$ neurons and sparsity of the internal connectivity equal to 25%. The readout is trained by standard ridge regression with regularization parameter set to 0.05. The distributions of the vertex properties are estimated by histograms with $b = 50$ bins. We comment that the reservoir size N_r can be increased/decreased without affecting the applicability of the proposed methodology (only the number of bins used to estimate the vertex properties distribution might be modified – see Methods). However, the number of neurons has an impact on the overall performance of the network and on the time and space complexities of the proposed method.

Figure 6 depicts the values assumed by γ and four graph-based measures in the case of the MSO prediction task. As can be seen, high correlation emerges between the (average) entropy of the vertex properties and the prediction accuracy. Since our approach is fully unsupervised, the proposed graph-based measures can approximate well the accuracy γ, regardless of the task learned by the readout (prediction, function approximation, reconstruction of past inputs, etc). In Table 2, we report the average correlation values and their statistical significance (expressed by p-values) on all tasks. As we can see, the highest (and statistically significant) correlation is achieved by using one of the four average entropy measures of vertex properties. In particular, the measure based on vertex cluster coefficient distribution, $\overline{H}^{\mathrm{CL}}$, achieves the best results in 5 of the 6 tasks. For what concerns the D4D time series, we observe that λ achieves high correlation with γ, but still lower than the one achieved by $\overline{H}^{\mathrm{CL}}$. This demonstrates the effectiveness of the proposed methodology, also in the case of a real-world application. In SIN and MSO tasks, the graph-based quantifiers estimated on the weighted HVG achieve a higher degree of accuracy, with respect to the binary counterpart. In these cases, additional qualitative information relative to temporal and amplitude differences in the connected data allows to better represent the dynamics of the system. Finally, it is worth noting that IMI takes high, yet negative correlation values on both MG and POLY tasks. In such cases, results are close to the ones achieved with our approach.

Test for memory capacity. The performed experiment consists in generating 100 different random reservoirs, each one characterized by an increasing value of spectral radius ρ in the [0.1, 2] interval. As ρ varies, we evaluate the MC by training four readouts in order to reproduce different time-lagged versions of input signal $\mathbf{x}_{10,5}, \ldots, \mathbf{x}_{25,20}$. Then, on the output of each reservoir, we evaluate the similarities δ^{TS}, δ^{DG}, and δ^{AND}, which are high if there exists at least one series of activations that is similar to the considered past input sequence. This is evaluated in such as way that the measure κ^{*} taken into account. Even if some neurons retain dynamics of previous input sequences, the reservoir introduces shifts in the phase and the amplitude of the input signal. To filter out these differences, in this test we consider only HVGs defined by binary adjacency matrices, which do not account for differences in the amplitude of the connected values. To evaluate the effectiveness of the proposed unsupervised memory measures, we compute the correlation between the supervised MC and δ^{TS}, δ^{DG}, δ^{AND}, as ρ varies within the chosen interval. Note that we only monitor the effect of ρ on the dynamics, since it is the hyperparameter that mostly affects the memory capacity[32]. We kept the input scaling fixed, $\omega_i = 0.7$, while the remaining hyperparamers are configured as in the previous experiment. As before, we repeated each experiment 15 times with different and independent random initializations. In Table 3, we show the mean correlation values, along with the standard deviations, between the MC and the proposed unsupervised measures of memory capacity.

From the table, we observe that the best agreement with the MC is achieved by the measures derived from the HVGs. In particular, δ^{DG} configured with the Spearman rank κ^{SC} is always highly correlated with MC and, in three of the four delayed input sequences taken into account, is the best performing one. In each setup, δ^{DG} works better if configured with κ^{SC} rather than κ^{PC}. Instead, results obtained with κ^{MI} are significantly worse in all cases. δ^{AND} achieves the best results only for the first time lag taken into account, while the agreement with MC is lower in the remaining cases. Interestingly, several measures show a high degree of correlation with the MC as the size of the delay increases. δ^{TS}, the unsupervised measure computed directly on the input time series and neuron activations, shows positive correlations with the MC, but the agreement is always lower with respect to the graph-based measures. For δ^{TS}, the setting with κ^{PC} works better than κ^{SC}. Finally, also in this case by using κ^{MI} we obtain the worst performance. In Fig. 7, we show an example of the values of MC, δ^{TS} (configured with κ^{PC}), δ^{DG} (configured with κ^{SC}), and δ^{AND}, as ρ is varied within the [0.2, 2] interval.

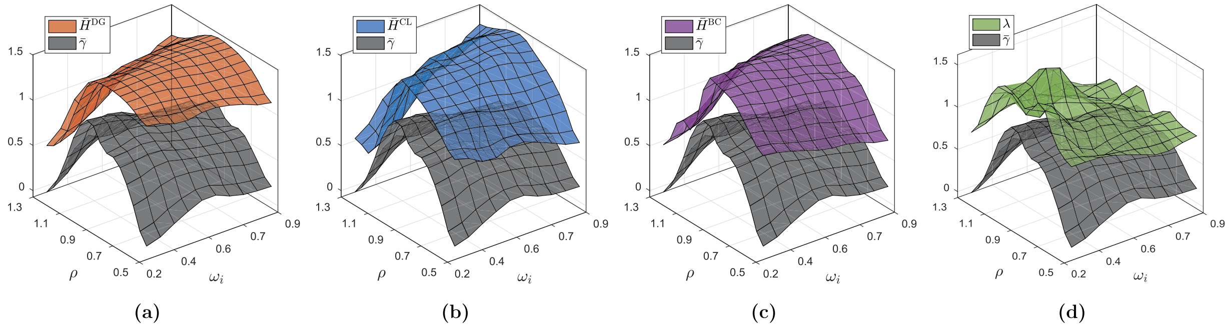


Figure 6. In each panel, the two-dimensional gray manifold represents the values of the prediction accuracy γ in the MSO task, for different configurations of ρ and ω_i. The colored manifolds are the average entropy of the distributions of (**a**) the vertex degree $\overline{H}^{\text{DG}}$, (**b**) the clustering coefficient $\overline{H}^{\text{CL}}$, (**c**) the betweenness centrality $\overline{H}^{\text{CC}}$. The manifold in (**d**) represents λ, which is computed from the ESN Jacobian. From the figure, we observe strong correlations between supervised prediction accuracy and the unsupervised measures (**a**–**c**) derived from the multiplex representation of the ESN activity.

Task	adj	corr(γ, $\overline{H}^{\text{DG}}$)	corr(γ, $\overline{H}^{\text{CL}}$)	corr(γ, $\overline{H}^{\text{BC}}$)	corr(γ, $\overline{H}^{\text{CC}}$)	corr(γ, IMI)	corr(γ, AEO)	corr(γ,λ)
SIN	b	0.489 (0.006)	0.488 (0.006)	0.157 (0.333)	0.042 (0.797)	−0.091 (0.632)	−0.326 (0.040)	0.154 (0.343)
	w	0.662 (0.000)	**0.705 (0.000)**	0.694 (0.000)	−0.127 (0.436)			
MG	b	0.577 (0.000)	**0.652 (0.000)**	−0.37 (0.019)	0.438 (0.005)	−0.617 (0.000)	0.414 (0.008)	−0.19 (0.239)
	w	−0.138 (0.396)	0.330 (0.038)	0.046 (0.777)	0.564 (0.000)			
MSO	b	−0.215 (0.183)	−0.206 (0.201)	0.427 (0.006)	0.333 (0.036)	−0.238 (0.139)	−0.312 (0.05)	0.571 (0.000)
	w	0.628 (0.000)	0.810 (0.000)	**0.820 (0.000)**	−0.246 (0.125)			
NARMA	b	0.511 (0.001)	**0.514 (0.001)**	−0.332 (0.037)	−0.473 (0.002)	−0.543 (0.000)	−0.472 (0.002)	0.399 (0.011)
	w	−0.373 (0.018)	−0.185 (0.254)	−0.420 (0.007)	−0.376 (0.017)			
POLY	b	0.755 (0.000)	**0.765 (0.000)**	−0.393 (0.012)	−0.306 (0.055)	−0.745 (0.000)	0.440 (0.005)	−0.557 (0.000)
	w	−0.133 (0.412)	−0.113 (0.487)	0.47 (0.002)	−0.171 (0.291)			
D4D	b	0.632 (0.000)	**0.677** (0.000)	−0.233 (0.104)	0.061 (0.676)	−0.611 (0.000)	0.168 (0.243)	0.670 (0.000)
	w	−0.455 (0.001)	−0.604 (0.000)	0.189 (0.188)	−0.409 (0.003)			

Table 2. Correlations and related *p*-values (in brackets) of the proposed graph-based measures with accuracy γ in different prediction tasks, as ρ and ω_i change. We report the correlation of the manifolds generated by the values of γ with the manifolds relative to $\overline{H}^{\text{DG}}$, $\overline{H}^{\text{CL}}$, $\overline{H}^{\text{BC}}$, and $\overline{H}^{\text{CC}}$, which are the average entropy values of the distributions of vertex degree, clustering coefficient, betweenness and closeness centrality. Each measure is computed on both the binary (b) and weighted (w) versions of the HVG adjacency matrix (adj). We also report the correlations of γ with the manifolds relative to the minimum singular value of the reservoir Jacobian over time (λ) and the two multiplex-based measures AEO and IMI, presented in ref. 40. In each task, the highest correlations with γ are highlighted in bold.

Input	*corr*(δTS, MC)			*corr*(δDG, MC)			
sequence	κPC	κSC	κMI	κPC	κSC	κ^{MI}	*corr*(δ^{AND}, MC)
$\mathbf{x}_{10,5}$	0.608 ± 0.094	0.594 ± 0.096	0.235 ± 0.054	0.682 ± 0.036	0.693 ± 0.029	0.607 ± 0.057	0.771 ± 0.034
$\mathbf{x}_{15,10}$	0.542 ± 0.042	0.546 ± 0.048	0.484 ± 0.047	0.547 ± 0.052	0.645 ± 0.038	0.343 ± 0.056	0.518 ± 0.043
$\mathbf{x}_{20,15}$	0.556 ± 0.068	0.550 ± 0.065	0.387 ± 0.038	0.776 ± 0.040	0.818 ± 0.046	0.182 ± 0.068	0.665 ± 0.075
$\mathbf{x}_{25,20}$	0.607 ± 0.043	0.603 ± 0.050	0.431 ± 0.066	0.811 ± 0.015	0.828 ± 0.019	0.501 ± 0.052	0.468 ± 0.147

Table 3. Mean correlations and standard deviations of MC with the unsupervised memory quantifiers δ^{TS}, δ^{DG}, and δ^{AND}. We consider 4 different sequences of past inputs $\mathbf{x}_{10,5}$,. $\mathbf{x}_{25,20}$. Values for each measure are computed as the spectral radius ρ of the ESNs reservoirs varies from 0.1 to 2. To compute δ^{TS} and δ^{DG}, we consider three different similarities: the Pearson correlation κ^{PC}, the Spearman correlation κ^{SC}, and the mutual information κ^{MI}. Best results for each input sequence are reported in bold.

Discussion

Experimental results show satisfactory correlations for average entropy of vertex properties with respect to prediction accuracy. Moreover, the two unsupervised graph-based memory measures that we proposed (δ^{DG} and δ^{AND}) correlate well with the supervised measure of memory capacity.

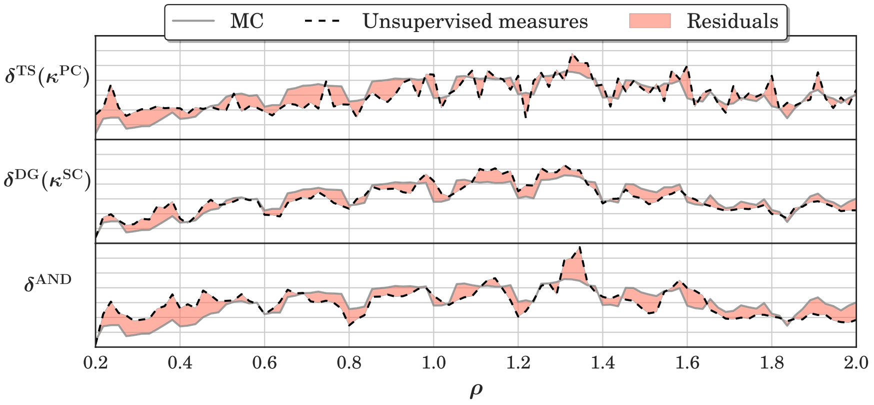


Figure 7. Values of supervised MC and of two selected unsupervised memory measures, when the input sequence $\mathbf{x}_{20,15}$ is taken into account. The smaller the residuals (red areas), the better. We can observe that the agreement of the MC with δ^{DG} (using κ^{SC}) is higher than δ^{TS} (using κ^{PC}) and δ^{AND}, as ρ varies in [0.2, 2].

We first discuss the results of the prediction accuracy test, where we analyzed topological properties of vertices in the multiplex, representing the ESN instantaneous state. On all tests taken into account, we observed a remarkable correlation between γ and the average entropy of the clustering coefficient distribution $\overline{H}^{CL}$, hence suggesting that the clustering coefficient is able to describe well the heterogeneity of the activations. To explain this result, it is necessary to elaborate on the properties of the clustering coefficient CL(·). In the HVG literature, CL(·) its behavior has been analyzed for time series characterized by different Hurst exponent[59]. Additionally, an upper bound ($\mathrm{CL}(v) \in [0, 2/\mathrm{DG}(v)]$) is provided for HVGs derived from random time series[41]. In the following, we present an in-depth interpretation of the results by accounting for geometrical properties of the clustering coefficient.

In a HVG, CL(v) measures the inter-visibility among neighbors of v. For convex functions, it is possible to connect any two points with a straight line. This feature is also (partially) captured by the HVG. If v is contained in a convex part of the related time series, there is a high degree of intervisibility among the neighbor vertices to which v is connected, hence CL(v) is high. Additionally, moving along the same convex part of the time series results only in minor changes of the clustering coefficient in the associated HVG vertices. Instead, if v is a local maximum of a concave part, then it is connected to points belonging to two different basins, which do not have reciprocal visibility. In this case, CL(v) is low and its value rapidly changes as one moves away from the maximum. This results in great losses of visual information. Therefore, large values of CL(·) indicate the presence of dominating convexities, while low values characterize concavities[60]. Accordingly, CL(·) can be used to measure the length of a convex (concave) part of the time series and how fast the convexity is changing, which is a measure of the fluctuations in the time series[61]. In a regime characterized by contractive dynamics, convexity changes at the same (slow) rate in different neuron activations and this results in a low entropy value of the clustering coefficient distribution among vertices in different layers. On the edge of criticality, instead, convex and concave parts in the time series of activations are characterized by heterogeneous lengths and they change at different rates. This corresponds to a high degree of clustering coefficient diversity of the same HVG vertex, replicated at different layers in the multiplex. Finally, in the chaotic regime, all time series fluctuate very rapidly and their convexity changes every few time steps. In this case, in each time series of activations the lengths of convex and concave parts are always very short and hence the desired heterogeneity is again lost.

For what concerns experiments on memory, the best overall results in terms of agreement with the supervised MC are achieved by the graph-based measure δ^{DG}. As previously discussed, such a measure evaluates the maximum similarity between the sequence of vertex degrees on the input HVG $G_{\mathbf{x}}$ and the HVG G_l of neuron activations. This measure is closely related with the degree distribution $P(k)$, whose importance is known in the HVG literature[41]. For example, it has been shown that for time series generated from an i.i.d. process, $P(k)$ follows $P(k) = (1/3)(2/3)^{k-2}$ and the mean degree is $\langle k \rangle = 4$. As the correlations in the time series increase, the i.i.d. assumption is lost and $P(k)$ decays faster. Furthermore, vertex degrees are key parameters to describe dynamic processes on the graph, such as synchronization of coupled oscillators, percolation, epidemic spreading, and linear stability of equilibrium in networked coupled systems[62]. Their role has been studied also in the HVG framework[63]. HVGs have been studied in the context of time series related to processes with power-law correlations[59]. In our case, the time series of neuron activations have short-term correlations and increments in the correlation coefficients can have opposite signs at consecutive time lags. For these cases, we are not aware of any previous study in terms of HVGs.

In networks which are inherently degree disassortative, the range of degree values increases with network size, with a consequent decrease of the assortativity value[64]. In such networks, the Spearman rank correlation provides a more suitable choice with respect to calculating degree-degree Pearson correlations. It is important to notice that the rank is computed through a non-linear rescaling, which is data dependent. The information on the actual values of the data is discarded as only its inherent ordering (rank) is preserved. We argue that HVGs convey the same type of information captured by the Spearman correlation. Hence, the latter should be preferred to Pearson correlation to characterize the characteristics of the vertices and related topological properties in HVGs. This fact justifies the higher agreement with memory capacity achieved by means of δ^{DG} when configured with κ^{SC}, which accounts for Spearman correlations between sequences of vertex degrees in the HVGs related to the input signal and the neuron activations.

Modeling ESN dynamics through a multiplex network allowed us to connect two seemingly different research fields, thus fostering multidisciplinary research in the context of recurrent neural networks. By converting a temporal problem into a topological one, we handled temporal dependencies introduced by ESNs (as well as by other types of RNNs), hence overcoming technical limitations of statistical approaches that require independence of samples. We performed and discussed several experiments that provided empirical evidence that our methodology achieves performance higher than other unsupervised methods and comparable to cross-validation techniques. These results suggest to allocate efforts to further improve the effectiveness of unsupervised learning methods in the context of ESNs and RNNs. Finally, we would like to stress that, while this paper is primarily focused on network structures in machine learning, our results might suggest new ideas for theoretical understanding of recurrent structures in biological models of neuronal networks[7,65,66]. In particular, we believe that it is possible to identify emergent structural patterns in the developed graph-based representations of network dynamics. This would allow to further explore and analyze the route to chaos in input-driven neural models by exploiting the language of graph theory, which is an already established framework within the neuroscience field[67].

References

1. Barzel, B. & Barabási, A.-L. Universality in network dynamics. *Nature Physics* **9,** 673–681 (2013).
2. Hammer, B., Micheli, A., Sperduti, A. & Strickert, M. Recursive self-organizing network models. *Neural Networks* **17,** 1061–1085 (2004).
3. Maass, W., Joshi, P. & Sontag, E. D. Computational aspects of feedback in neural circuits. *PLoS Computational Biology* **3,** e165 (2007).
4. Reinhart, R. F. & Steil, J. J. Regularization and stability in reservoir networks with output feedback. *Neurocomputing* **90,** 96–105 (2012).
5. Jordan, M. I. Serial order: A parallel distributed processing approach. In Donahoe, J. W. & Dorsel, V. P. (eds) *Neural-Network Models of Cognition: Biobehavioral Foundations*, vol. 121 of *Advances in Psychology*, 471–495 (North-Holland, 1997).
6. Amit, D. J., Gutfreund, H. & Sompolinsky, H. Spin-glass models of neural networks. *Physical Review A* **32,** 1007–1018 (1985).
7. Enel, P., Procyk, E., Quilodran, R. & Dominey, P. F. Reservoir computing properties of neural dynamics in prefrontal cortex. *PLoS Computational Biology* **12,** e1004967 (2016).
8. Barak, O., Sussillo, D., Romo, R., Tsodyks, M. & Abbott, L. F. From fixed points to chaos: three models of delayed discrimination. *Progress in Neurobiology* **103,** 214–222 (2013).
9. Rajan, K., Abbott, L. F. & Sompolinsky, H. Stimulus-dependent suppression of chaos in recurrent neural networks. *Physical Review E* **82,** 011903 (2010).
10. Fusi, S., Miller, E. K. & Rigotti, M. Why neurons mix: high dimensionality for higher cognition. *Current Opinion in Neurobiology* **37,** 66–74 (2016).
11. Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Networks* **61,** 85 –117 (2015).
12. Barra, A., Bernacchia, A., Santucci, E. & Contucci, P. On the equivalence of Hopfield networks and Boltzmann machines. *Neural Networks* **34,** 1–9 (2012).
13. Charles, A., Yin, D. & Rozell, C. Distributed sequence memory of multidimensional inputs in recurrent networks. *arXiv preprint arXiv:1605.08346* (2016).
14. Tiňo, P. & Rodan, A. Short term memory in input-driven linear dynamical systems. *Neurocomputing* **112,** 58–63 (2013).
15. Legenstein, R. & Maass, W. What makes a dynamical system computationally powerful? In Haykin, S., Principe, J., Sejnowski & McWhirter (eds) *New Directions in Statistical Signal Processing: From Systems to Brain*, 127–154 (MIT Press, Cambridge, 2007).
16. Yildiz, I. B., Jaeger, H. & Kiebel, S. J. Re-visiting the echo state property. *Neural Networks* **35,** 1–9 (2012).
17. Dambre, J., Verstraeten, D., Schrauwen, B. & Massar, S. Information processing capacity of dynamical systems. *Scientific Reports* **2** (2012).
18. Lajoie, G., Lin, K. K., Thivierge, J.-P. & Shea-Brown, E. Encoding in balanced networks: Revisiting spike patterns and chaos in stimulus-driven systems. *PLoS Computational Biology* **12,** 1–30 (2016).
19. Livi, L., Bianchi, F. M. & Alippi, C. Determination of the edge of criticality in echo state networks through Fisher information maximization. *IEEE Transactions on Neural Networks and Learning Systems* 1–12, doi: 10.1109/TNNLS.2016.2644268 (2017).
20. Mayer, N. M. Input-anticipating critical reservoirs show power law forgetting of unexpected input events. *Neural Computation* **27,** 1102–1119 (2015).
21. Moretti, P. & Muñoz, M. A. Griffiths phases and the stretching of criticality in brain networks. *Nature Communications* **4** (2013).
22. Mora, T. & Bialek, W. Are biological systems poised at criticality? *Journal of Statistical Physics* **144,** 268–302 (2011).
23. Scheffer, M. *et al.* Anticipating critical transitions. *Science* **338,** 344–348 (2012).
24. Wang, X., Lizier, J. & Prokopenko, M. Fisher information at the edge of chaos in random Boolean networks. *Artificial Life* **17,** 315–329 (2011).
25. Tkačik, G. *et al.* Thermodynamics and signatures of criticality in a network of neurons. *Proceedings of the National Academy of Sciences* **112,** 11508–11513 (2015).
26. Marković, D. & Gros, C. Power laws and self-organized criticality in theory and nature. *Physics Reports* **536,** 41–74 (2014).
27. Lukoševičius, M. & Jaeger, H. Reservoir computing approaches to recurrent neural network training. *Computer Science Review* **3,** 127–149 (2009).
28. Grigoryeva, L., Henriques, J., Larger, L. & Ortega, J.-P. Optimal nonlinear information processing capacity in delay-based reservoir computers. *Scientific Reports* **5** (2015).
29. Bianchi, F. M., Scardapane, S., Uncini, A., Rizzi, A. & Sadeghian, A. Prediction of telephone calls load using echo state network with exogenous variables. *Neural Networks* **71,** 204–213 (2015).
30. Bianchi, F. M., Livi, L. & Alippi, C. Investigating echo state networks dynamics by means of recurrence analysis. *IEEE Transactions on Neural Networks and Learning Systems* 1–13, doi: 10.1109/TNNLS.2016.2630802 (2016).
31. Boedecker, J., Obst, O., Lizier, J. T., Mayer, N. M. & Asada, M. Information processing in echo state networks at the edge of chaos. *Theory in Biosciences* **131,** 205–213 (2012).
32. Ozturk, M. C., Xu, D. & Prncipe, J. C. Analysis and design of echo state networks. *Neural Computation* **19,** 111–138 (2007).
33. Verstraeten, D. & Schrauwen, B. On the quantification of dynamics in reservoir computing. In *Artificial Neural Networks-ICANN 2009* 985–994 (Springer Berlin Heidelberg, 2009).
34. Leisch, F., Trapletti, A. & Hornik, K. Stationarity and stability of autoregressive neural network processes. In Kearns, M. J., Solla, S. A. & Cohn, D. A. (eds) *Advances in Neural Information Processing Systems* **11,** 267–273 (MIT Press, 1999).
35. Lee, K.-M., Min, B. & Goh, K.-I. Towards real-world complexity: an introduction to multiplex networks. *The European Physical Journal B* **88,** 1–20 (2015).
36. Boccaletti, S. *et al.* The structure and dynamics of multilayer networks. *Physics Reports* **544,** 1–122 (2014).
37. Menichetti, G., Remondini, D., Panzarasa, P., Mondragón, R. J. & Bianconi, G. Weighted multiplex networks. *PloS ONE* **9,** e97857 (2014).
38. Kivelä, M. *et al.* Multilayer networks. *Journal of Complex Networks* **2,** 203–271 (2014).
39. De Domenico, M. *et al.* Mathematical formulation of multilayer networks. *Physical Review X* **3,** 041022 (2013).
40. Lacasa, L., Nicosia, V. & Latora, V. Network structure of multivariate time series. *Scientific Reports* **5** (2015).

41. Luque, B., Lacasa, L., Ballesteros, F. & Luque, J. Horizontal visibility graphs: Exact results for random time series. *Physical Review E* **80,** 046103 (2009).
42. Luque, B., Lacasa, L., Ballesteros, F. J. & Robledo, A. Feigenbaum graphs: A complex network perspective of chaos. *PLoS One* **6,** e22411 (2011).
43. Luque, B., Cordero-Gracia, M., Gómez, M. & Robledo, A. Quasiperiodic graphs at the onset of chaos. *Physical Review E* **88,** 062918 (2013).
44. Lacasa, L. & Toral, R. Description of stochastic and chaotic series using visibility graphs. *Physical Review E* **82,** 036120 (2010).
45. Ravetti, M. G., Carpi, L. C., Gonçalves, B. A., Frery, A. C. & Rosso, O. A. Distinguishing noise from chaos: Objective versus subjective criteria using horizontal visibility graph. *PloS ONE* **9,** e108004 (2014).
46. Luque, B., Lacasa, L., Ballesteros, F. J. & Robledo, A. Analytical properties of horizontal visibility graphs in the Feigenbaum scenario. *Chaos: An Interdisciplinary Journal of Nonlinear Science* **22,** 013109 (2012).
47. Zhang, S. *et al.* Architectural complexity measures of recurrent neural networks. *arXiv preprint arXiv:1602.08210* (2016).
48. Hermans, M. & Schrauwen, B. Recurrent kernel machines: Computing with infinite echo state networks. *Neural Computation* **24,** 104–133 (2012).
49. Lukoševičius, M. *A Practical Guide to Applying Echo State Networks*, 659–686 (Springer Berlin Heidelberg, Berlin, Heidelberg, 2012).
50. Saramäki, J., Kivelä, M., Onnela, J.-P., Kaski, K. & Kertesz, J. Generalizations of the clustering coefficient to weighted complex networks. *Physical Review E* **75,** 027105 (2007).
51. Boccaletti, S., Latora, V., Moreno, Y., Chavez, M. & Hwang, D. Complex networks: Structure and dynamics. *Physics Reports* **424,** 175–308 (2006).
52. Bertschinger, N. & Natschläger, T. Real-time computation at the edge of chaos in recurrent neural networks. *Neural Computation* **16,** 1413–1436 (2004).
53. Pascanu, R. & Jaeger, H. A neurodynamical model for working memory. *Neural Networks* **24,** 199–207 (2011).
54. Jaeger, H. & Haas, H. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science* **304,** 78–80 (2004).
55. Jaeger, H. Adaptive nonlinear system identification with echo state networks. In *Advances in Neural Information Processing Systems*, 593–600 (MIT Press, 2002).
56. Butcher, J., Verstraeten, D., Schrauwen, B., Day, C. & Haycock, P. Reservoir computing and extreme learning machines for non-linear time-series data analysis. *Neural Networks* **38,** 76 –89 (2013).
57. Blondel, V. D. *et al.* Data for Development: the D4D Challenge on Mobile Phone Data. *ArXiv preprint arXiv:1210.0137* (2012).
58. Orange d4d challenge. http://www.d4d.orange.com/en/Accueil. Accessed: 2016-09-22.
59. Xie, W.-J. & Zhou, W.-X. Horizontal visibility graphs transformed from fractional Brownian motions: Topological properties versus the Hurst index. *Physica A: Statistical Mechanics and its Applications* **390,** 3592–3601 (2011).
60. Costa, L. d. F., Rodrigues, F. A., Travieso, G. & Villas Boas, P. R. Characterization of complex networks: A survey of measurements. *Advances in Physics* **56,** 167–242 (2007).
61. Turner, A., Doxa, M., O'Sullivan, D. & Penn, A. From isovists to visibility graphs: A methodology for the analysis of architectural space. *Environment and Planning B: Planning and Design* **28,** 103–121 (2001).
62. Restrepo, J. G., Ott, E. & Hunt, B. R. Approximating the largest eigenvalue of network adjacency matrices. *Physical Review E* **76,** 056119 (2007).
63. Fioriti, V., Tofani, A. & Di Pietro, A. Discriminating chaotic time series with visibility graph eigenvalues. *Complex Systems* **21** (2012).
64. Newman, M. E. J. Assortative mixing in networks. *Physical Review Letters* **89,** 208701 (2002).
65. Marblestone, A. H., Wayne, G. & Kording, K. P. Toward an integration of deep learning and neuroscience. *Frontiers in Computational Neuroscience* **10** (2016).
66. Sussillo, D., Churchland, M. M., Kaufman, M. T. & Shenoy, K. V. A neural network that finds a naturalistic solution for the production of muscle activity. *Nature Neuroscience* **18,** 1025–1033 (2015).
67. Sporns, O. *Networks of the Brain* (MIT press, Cambridge, MA, USA, 2011).
68. Lopez-Fernandez, L., Robles, G. & Gonzalez-Barahona, J. M. Applying social network analysis to the information in CVS repositories. In *Proceedings of the International Workshop on Mining Software Repositories* 101–105 (Edinburgh, UK, 2004).

Author Contributions

L.L. outlined the research ideas. F.M.B. and L.L. conceived methods. F.M.B. performed experiments and generated figures. C.A. and R.J. contributed to the technical discussion. All authors took part to the paper writing and approved the final manuscript.

Additional Information

Competing Interests: The authors declare no competing financial interests.

How to cite this article: Bianchi, F. M. *et al.* Multiplex visibility graphs to investigate recurrent neural networks dynamics. *Sci. Rep.* **7**, 44037; doi: 10.1038/srep44037 (2017).

Publisher's note: Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.