Deep Reservoir Computing

Claudio Gallicchio and Alessio Micheli

Abstract This chapter surveys the recent advancements on the extension of Reservoir Computing toward deep architectures, which is gaining increasing research attention in the neural networks community. Within this context, we focus on describing the major features of Deep Echo State Networks based on hierarchical composition of multiple reservoirs. The intent is to provide a useful reference to guide applications and further developments of this efficient and effective class of approaches to deal with times-series and more complex data within a unified description and analysis.

Key words: Deep Reservoir Computing, Deep Echo State Networks, Deep Recurrent Neural Networks

1 Introduction

In recent years, the study of deep neural network architectures for temporal data has been an attractive area of research in the neural networks community [2, 32, 54]. Investigations in the field of hierarchically organized Recurrent Neural Networks (RNNs) showed that deep RNNs are able to develop internal states that are multiple time-scale representations of the temporal information. This is a much desired feature, e.g., when approaching complex tasks especially in domains related to human cognition, like speech and text processing [33, 35]. In addition, the interest in studying hierarchical RNN

Claudio Gallicchio
Department of Computer Science, University of Pisa, Largo Bruno Pontecorvo 3, 56127 Pisa (Italy), e-mail: gallicch@di.unipi.it

Alessio Micheli
Department of Computer Science, University of Pisa, Largo Bruno Pontecorvo 3, 56127 Pisa (Italy), e-mail: micheli@di.unipi.it

models finds strong motivation also from the different, but related, perspective of computational neuroscience, from which we know that a "deep" hierarchical organization of recurrent neural units is a major pattern in the neocortex (e.g. [31, 4]). In this sense, information processing in deep RNN architectures has a strong biological motivation.

Recently, within the umbrella of randomized neural network approaches [30, 12, 29, 52], the Reservoir Computing (RC) [47, 60] paradigm offered a novel perspective to the analysis and design of deep RNNs. In particular, in the context of discrete time reservoirs, the introduction of the Deep Echo State Network (DeepESN) model [23, 14] has allowed study of the properties of layered RNN architectures separately from the learning aspects. Remarkably, such studies pointed out that the structured state space organization with multiple time-scale dynamics in deep RNNs is *intrinsic* to the nature of compositionality of recurrent neural models. The interest in the study of the DeepESN model is hence twofold. On the one hand, sheds light on the intrinsic properties of state dynamics of layered RNNs [15, 28, 19]. On the other hand, it enables the design of efficiently trained deep neural networks for temporal data, capable of improving on previous state-of-the-art results in complex tasks [24].

From a historical perspective, before the explicit introduction of the DeepESN model in [23], preliminary studies on hierarchical RC models targeted ad-hoc constructed architectures, where different modules were trained for discovery of temporal features at different scales on synthetic data [39]. Moreover, ad-hoc constructed modular networks made up of multiple ESN modules have also been investigated in the speech processing area [58, 59]. More recently, the advantages of multi-layered RC networks have been experimentally studied on time-series benchmarks in the RC area [49]. Differently from the above mentioned works, the studies on DeepESN considered in the following aim to address some fundamental questions pertaining to the true nature of layering as a factor of architectural RNN design [19]. Such basic questions can be essentially summarized as follows:
(i) Why stacking layers of recurrent units?
(ii) What is the inherent architectural effect of layering in RNNs (independently from learning)?
(iii) Can we extend the advantages of depth in RNN design using efficiently trained RC approaches?
(iv) Can we exploit the insights from such analysis to address the automatic design of deep recurrent models (including fundamental parameters such as the architectural form, the number of layers, the number of units in each layer, etc.)?

This chapter is intended both to draw a line of recent developments in response to the above mentioned key research questions and to provide an up-to-date overview on the progress and on the perspectives in the studies of DeepESNs. The rest of this contribution is organized as follows. The DeepESN model is introduced and discussed in Section 2, both from the ar-

chitectural and the dynamical system viewpoints. The progressive advances in the study of DeepESN field are summarized in Section 3, while further developments related to other hierarchical reservoir models are recalled in Section 4. Finally, conclusions are drawn in Section 5.

2 Deep Echo State Network

This section is intended to provide an introduction to the major characteristics of deep RC models. In particular, we focus on discrete-time reservoir systems, i.e., we frame our analysis adopting the formalism of Echo State Networks (ESNs) [40, 37]. In this context, we illustrate the main properties of deep reservoir architectures in Section 2.1, while in Section 2.2 we analyze the behavior of deep reservoirs from the point of view of dynamical systems.

2.1 Architecture

As for the standard shallow ESN model [37, 40], a DeepESN [23] is composed by a dynamical *reservoir* system, which embeds the input history into a rich state representation, and by a feed-forward *readout* part, which exploits the state encoding provided by the reservoir to compute the output. Crucially, the reservoir of a DeepESN is organized into a *hierarchy of stacked recurrent layers*, where the output of each layer acts as input for the next one. At each time step t, the state computation proceeds by following the pipeline of recurrent layers, from the first one, which is directly fed by the external input, up to the highest one in the reservoir architecture (i.e., the farthest one from the external input). The layered reservoir architecture of a DeepESN is illustrated in Figure 1. In our notation, we use N_U to denote the external input dimension, N_L to indicate the number of reservoir layers, and we assume, for the only sake of simplicity, that each reservoir layer has N_R recurrent units. Moreover, we use $\mathbf{u}(t) \in \mathbb{R}^{N_U}$ to denote the external input at time step t, while $\mathbf{x}^{(i)}(t) \in \mathbb{R}^{N_R}$ is the state of the reservoir layer i at time step t. In general, we use the superscript (i) to indicate that an item is related to the i-th reservoir in the stack. At each time step t, the composition of the states in all the reservoir layers, i.e. $\mathbf{x}(t) = \left(\mathbf{x}^{(1)}(t), \ldots, \mathbf{x}^{(N_L)}(t)\right) \in \mathbb{R}^{N_R\, N_L}$, gives the global state of the network.

Assuming leaky integrator reservoir units [41] in each layer and omitting the bias terms for the ease of notation, the state transition functions in the reservoir layers can be described as follows. For the first layer we have that:

$$\mathbf{x}^{(1)}(t) = (1 - a^{(1)})\mathbf{x}^{(1)}(t-1) + a^{(1)}\mathbf{f}(\mathbf{W}^{(1)}\mathbf{u}(t) + \hat{\mathbf{W}}^{(1)}\mathbf{x}^{(1)}(t-1)), \quad (1)$$

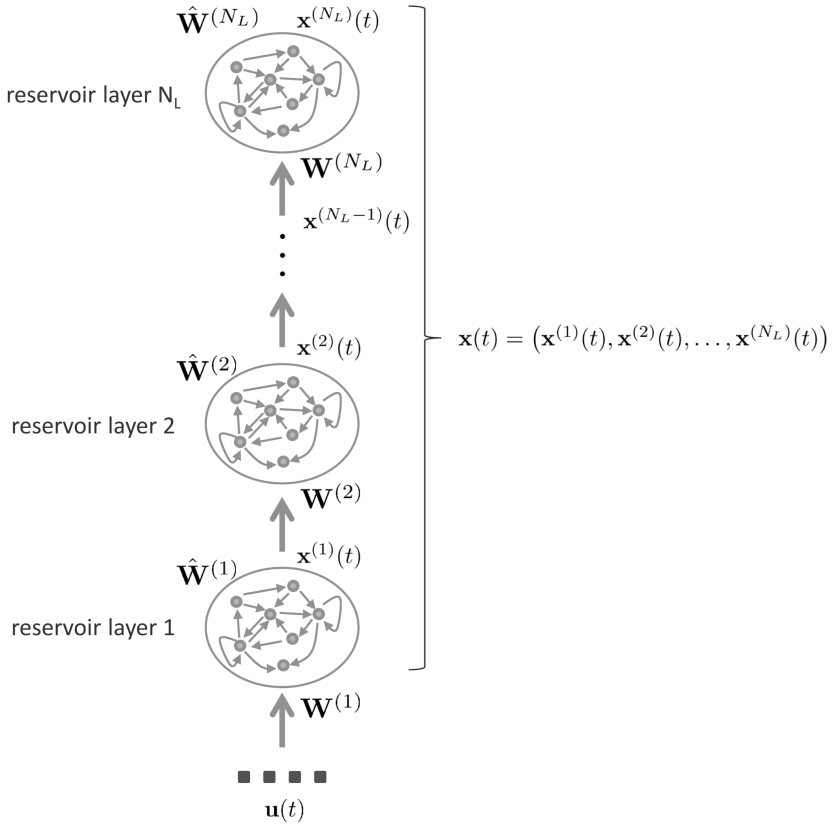

Fig. 1 Reservoir architecture of a Deep Echo State Network.

while for successive layers $i > 1$ the state update is given by:

$$\mathbf{x}^{(i)}(t) = (1 - a^{(i)})\mathbf{x}^{(i)}(t-1) + a^{(i)}\mathbf{f}(\mathbf{W}^{(i)}\mathbf{x}^{(i-1)}(t) + \hat{\mathbf{W}}^{(i)}\mathbf{x}^{(i)}(t-1)). \quad (2)$$

In the above equations 1 and 2, $\mathbf{W}^{(1)} \in \mathbb{R}^{N_R \times N_U}$ denotes the input weight matrix, $\mathbf{W}^{(i)} \in \mathbb{R}^{N_R \times N_R}$ (for $i > 1$) is the weight matrix for inter-layer connections from layer $(i-1)$ to layer i, $\hat{\mathbf{W}}^{(i)} \in \mathbb{R}^{N_R \times N_R}$ is the recurrent weight matrix for layer i, $a^{(i)} \in [0, 1]$ is the leaking rate for layer i and $\mathbf{f}$ denotes the element-wise applied activation function for the recurrent reservoir units (typically, the *tanh* non-linearity is used).

Remark 1 In light of the mathematical description introduced in equations 1 and 2, we can see that the standard (shallow) ESN model can be seen as a special case of DeepESN, obtained whenever a single reservoir layer is considered, i.e. for $N_L = 1$.

Interestingly, as graphically illustrated in Figure 2, we can observe that the reservoir architecture of a DeepESN can be characterized, with respect to the shallow counterpart, by interpreting it as a constrained version of standard shallow ESN/RNN with the same total number of recurrent units. In particular, the following constraints are applied in order to obtain a layered architecture:

- all the connections from the input layer to reservoir layers at a level higher than 1 are removed (influencing the way in which the external input information is seen by recurrent units progressively more distant from the input layer);
- all the connections from higher layers to lower ones are removed (which affects the flow of information and the dynamics of sub-parts of the network's state);
- all the connections from each layer to higher layers different from the next one in the pipeline are removed (which affects the flow of information and the dynamics of sub-parts of the network's state).

The above mentioned constraints, that graphically correspond to layering, have been explicitly and extensively discussed in our previous work in [23]. Under this point of view, the DeepESN architecture can be seen as a simplification of the corresponding single-layer ESN, leading to a reduction in the absolute number of recurrent weights which, assuming full-connected reservoirs at each layer, is quadratic in both the number of recurrent units per layer and total number of layers [28]. As detailed in the above points, however, note that this peculiar architectural organization influences the way in which the temporal information is processed by the different sub-parts of the hierarchical reservoir, composed by recurrent units that are progressively more distant from the external input.

Furthermore, differently from the case of a standard ESN/RNN, the state information transmission between consecutive layers in a DeepESN presents no temporal delays. In this respect, we can make the following considerations:

- the aspect of sequentiality between layers operation is already present and discussed in previous works in literature on deep RNN (see e.g. [35, 33, 9, 53]), which actually stimulated the investigation on the intrinsic role of layering in such hierarchically organized recurrent network architectures;
- this choice allows the model to process the temporal information at each time step in a "deep" temporal fashion, i.e., through a hierarchical composition of multiple levels of recurrent units;
- in particular, notice that the use of (hyperbolic tangent) non-linearities applied individually to each layer during the state computation does not allow to describe the DeepESN dynamics by means of an equivalent shallow system.

Based on the above observations, a major research question naturally arises and drives the motivation to the studies reported in Section 3, i.e., how and to

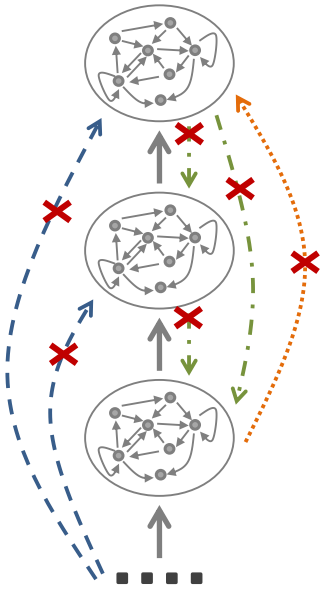

Fig. 2 The layered reservoir architecture of DeepESN as a constrained version of a shallow reservoir. Compared to the shallow case with the same total number of recurrent units, in a stacked DeepESN architecture the following connections are removed: from the input to reservoir levels at height > 1 (blue dashed arrows), from higher to lower reservoir levels (green dash dotted arrows), from each reservoir at level i to all reservoirs at levels higher than $i + 1$ (orange dotted arrows).

what extent, do the described constraints that rule the layered construction and the hierarchical representation in deep recurrent models have an influence on their dynamics?

As regards the output computation, although different choices are possible for the pattern of connectivity between the reservoir layers and the output module (see e.g. [35, 51]), a typical setting consists in feeding at each time step t the state of all reservoir layers (i.e., the global state of the DeepESN) to the output layer, as illustrated in Figure 3. Note that this choice enables the readout component to give different weights to the dynamics developed at different layers, thereby allowing to exploit the potential variety of state representations in the stack of reservoirs. Under this setting, denoting by N_Y the size of the output space, in the typical case of linear readout, the output at time step t is computed as:

$$\mathbf{y}(t) = \mathbf{W}_{out}\mathbf{x}(t) = \mathbf{W}_{out}\big(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N_L)}\big), \tag{3}$$

where $\mathbf{W}_{out} \in \mathbb{R}^{N_Y \times N_R N_L}$ is the readout weight matrix that is adapted on a training set, typically in closed form through direct methods such as pseudo-inversion or ridge-regression.

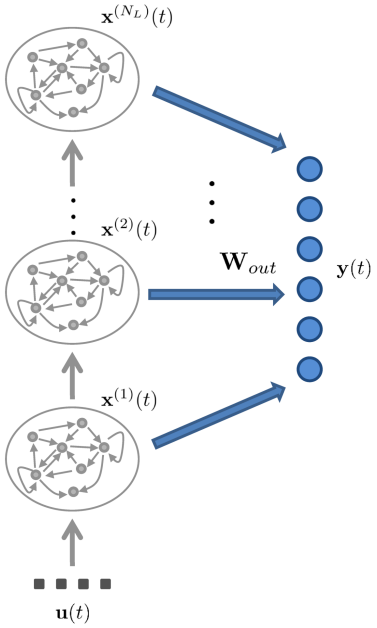

Fig. 3 Readout organization for DeepESN in which at each time step the reservoir states of all layers are used as input for the output layer.

As in the standard RC framework, all the reservoir parameters, i.e., the weights in matrices $\mathbf{W}^{(i)}$ and $\hat{\mathbf{W}}^{(i)}$, are left untrained after initialization under stability constraints given by the Echo State Property [37, 63, 15]. This aspect is related to the analysis of dynamical regimes of stacked reservoirs systems, and it is detailed in Section 2.2.

2.2 Dynamics of Deep Reservoirs and Echo State Property

The computation carried out by the stack of reservoirs of a DeepESN can be analyzed from a dynamical system viewpoint in terms of input-driven discrete-time non-linear dynamical systems. In particular, we can see that the dynamics of the first layer, driven by the external input, are ruled by a function $F^{(1)}$:

$$\begin{aligned} F^{(1)} &: \mathbb{R}^{N_R} \times \mathbb{R}^{N_U} \rightarrow \mathbb{R}^{N_R} \\ \mathbf{x}^{(1)}(t) &= F^{(1)}(\mathbf{x}^{(1)}(t-1), \mathbf{u}(t)). \end{aligned} \tag{4}$$

The dynamical behavior of each successive layer $i > 1$ is driven by the state of the previous layer in the pipeline, which determines a dependence (through multiple non-linearities) of $\mathbf{x}^{(i)}(t)$ from the states of the hierarchy computed

at the previous time step from the first layer up to level i, i.e., $\mathbf{x}^{(1)}(t-1), \ldots, \mathbf{x}^{(i)}(t-1)$, as well as from the input. This is expressed by a function $F^{(i)}$, as follows:

$$\begin{aligned} F^{(i)} &: \underbrace{\mathbb{R}^{N_R} \times \ldots \times \mathbb{R}^{N_R}}_{i \ times} \times \mathbb{R}^{N_U} \rightarrow \mathbb{R}^{N_R} \\ \mathbf{x}^{(i)}(t) &= F^{(i)}(\mathbf{x}^{(1)}(t-1), \ldots, \mathbf{x}^{(i)}(t-1), \mathbf{u}(t)). \end{aligned} \tag{5}$$

Note that for both equations 4 and 5, the specific shape of the state transition functions has been described in terms of leaky integrator reservoir units respectively in equations 1 and 2, and are parametrized by the weight values in matrices $\mathbf{W}^{(i)}$ and $\hat{\mathbf{W}}^{(i)}$, for $i = 1, \ldots, N_L$.

When we turn into considering the global state of the DeepESN as the composition of the reservoir states in all the levels of the hierarchy, i.e., $\mathbf{x}(t) = \left(\mathbf{x}^{(1)}(t), \ldots, \mathbf{x}^{(N_L)}(t)\right) \in \mathbb{R}^{N_R N_L}$, we can see that the state dynamics are ruled by a global state transition function F. This function can be defined as a composition of the layer-wise applied functions $F^{(i)}$, i.e. $F = \left(F^{(1)}, \ldots, F^{(N_L)}\right)$. At each time step t, function F computes the next state of the entire deep reservoir system based on the external input information and on the previous state of the deep reservoir, as follows:

$$\begin{aligned} F &: \underbrace{\mathbb{R}^{N_R} \times \ldots \times \mathbb{R}^{N_R}}_{N_L \ times} \times \mathbb{R}^{N_U} \rightarrow \underbrace{\mathbb{R}^{N_R} \times \ldots \times \mathbb{R}^{N_R}}_{N_L \ times} \\ \mathbf{x}(t) &= F(\mathbf{x}(t), \mathbf{u}(t)) \\ &= \left(F^{(1)}(\mathbf{x}^{(1)}(t-1), \mathbf{u}(t)), \ldots, F^{(N_L)}(\mathbf{x}^{(1)}(t-1), \ldots, \mathbf{x}^{(N_L)}(t-1), \mathbf{u}(t))\right). \end{aligned} \tag{6}$$

As in the case of standard shallow RC architectures, in order to avoid training of the reservoir connections, the state dynamics of a deep reservoir system described by equation 6 should exhibit global asymptotic (Lyapunov) stability, as prescribed by the Echo State Property (ESP) [37, 63]. This aspect has been analyzed in detail in [15], where the well known algebraic conditions for the ESP have been extended to cope with the case of deep reservoirs. Here we recall the statements of Theorems 1 and 2 in [15], which provide practical means for initialization of DeepESNs. Note that, as in the shallow case, when analyzing the behavior of deep reservoirs we shall assume that both the input space and the reservoir state spaces in all the layers are compact sets[1].

Theorem 1 (Necessary Condition for the ESP of DeepESN)

Consider a DeepESN whose dynamics are ruled by equation 6, implemented in terms of leaky integrator reservoir units as in equations 1 and 2, and assume that the null sequence is an admissible input for the system. Then a

[1] The latter set of conditions is ensured when the reservoir state transition functions in equations 1 and 2 are squashing non-linearities, such is the case, e.g., of tanh.

necessary condition for the ESP to hold is provided by the following equation:

$$\max_{i=1,\dots,N_L} \rho^{(i)} = \max_{i=1,\dots,N_L} \rho\big((1-a^{(i)})\mathbf{I} + a^{(i)}\hat{\mathbf{W}}^{(i)}\big) < 1, \tag{7}$$

where $\rho(\cdot)$ denotes the spectral radius operator (i.e. the maximum absolute eigenvalue of its matrix argument), and $\mathbf{I}$ is the identity matrix of size N_R.

Theorem 2 (Sufficient Condition for the ESP of DeepESN)

Consider a DeepESN whose dynamics are ruled by equation 6, implemented in terms of leaky integrator reservoir units as in equations 1 and 2, with tanh *non-linearity as activation function. If the DeepESN is featured by globally contractive dynamics then it satisfies the ESP. Accordingly, a sufficient condition for the ESP to hold is given by the following equation:*

$$\max_{i=1,\dots,N_L} C^{(i)} < 1, \tag{8}$$

where $C^{(i)}$ denotes the Lipschitz constant of the state transition function $F^{(i)}$ of the i-th reservoir level, and it is computed as follows:

$$C^{(i)} = \begin{cases} (1-a^{(1)}) + a^{(1)}\|\hat{\mathbf{W}}^{(1)}\| & if \;\; i = 1 \\ (1-a^{(i)}) + a^{(i)}\big(C^{(i-1)}\|\mathbf{W}^{(i)}\| + \|\hat{\mathbf{W}}^{(i)}\|\big) & if \;\; i > 1, \end{cases} \tag{9}$$

where $\|\cdot\|$ is the matrix norm induced by the L_2-norm defined on the corresponding state spaces.

The proofs of both Theorems 1 and 2 are given in [15].

A simple approach to initialize the reservoir weights in DeepESN is then to randomly draw the elements in $\mathbf{W}^{(i)}$ and $\hat{\mathbf{W}}^{(i)}$, e.g., from a uniform distribution in $[-1, 1]$, and then rescale them in order to meet one of the conditions expressed by Theorems 1 or 2. As for standard shallow reservoirs, the sufficient condition is often too restrictive in practice, and the necessary one is commonly adopted in DeepESN applications.

Remark 2 The necessary and the sufficient conditions for the ESP of DeepESN expressed by Theorems 1 and 2 generalize the corresponding conditions for shallow reservoirs given in standard RC literature [37, 63], respectively obtained from equations 7 and 9 by considering reservoir architectures with just one layer, i.e. for $N_L = 1$.

An interesting insight that we can get from the formulations of the conditions in Theorems 1 and 2, is that adding progressively more reservoir layers to the architecture of a DeepESN can never lower either the degree of stability (max operator in equation 7) or the Lipschitz constant (max operator in

equation 8) of the global deep reservoir system. This essentially translates into a propensity of deeper recurrent neural systems to show longer memory spans even in the absence of training of the recurrent connections (as observed by several numerical simulations in [23, 15]). This insight is also confirmed by more in-depth studies on local Lyapunov exponents of DeepESN states, reported in [28]. In particular, the results of the analysis given in [28] indicate that in conditions of equal number of recurrent units, deeper reservoir architectures are more easily shifted nearby the edge of stability (or criticality), a dynamical regime close to a stable-unstable transition where recurrent neural systems are known to develop richer temporal representations of their driving input signals [44, 43].

3 Advances

In this section we briefly survey the recent advances in the study of the DeepESN model. The works described in the following, by addressing the key questions summarized in the Introduction, provide general support to the significance of the DeepESN, and also critically discuss its advantages and drawbacks. An updated overview on the advancements in DeepESN research is also available in [17].

Multiple time-scale representation. The DeepESN model has been introduced in [23], which extends the preliminary work in [14]. The analysis provided in these papers revealed, through empirical investigations, the hierarchical structure of temporal data representations developed by the layered reservoir architecture of a DeepESN. Specifically, the stacked composition of recurrent reservoir layers was shown to enable a *multiple time-scale representation* of the temporal information, naturally ordered along the network's hierarchy. Besides, in [23] layering proved effective also as a way to enhance the effect of known RC factors of network design, including unsupervised reservoir adaptation by means of Intrinsic Plasticity [55]. The resulting effects have been analyzed also in terms of state entropy and memory.

Multiple frequency representation. The hierarchically structured state representation in DeepESNs has been investigated by means of *frequency analysis* in [26], which specifically considered the case of recurrent units with *linear* activation functions. Results pointed out the intrinsic multiple frequency representation in DeepESN states, where, even in the simplified linear setting, progressively higher layers focus on progressively lower frequencies. In [26] the potentiality of the deep RC approach has also been exploited in predictive experiments, showing that DeepESNs outperform state-of-the-art results on the class of Multiple Superimposed Oscillator

(MSO) tasks by several orders of magnitude.

Echo State Property. The fundamental RC conditions related to the *Echo State Property* (ESP) have been generalized to the case of deep RC networks in [15]. Specifically, through the study of stability and contractivity of nested dynamical systems, the theoretical analysis in [15] gives a sufficient condition and a necessary condition for the Echo State Property to hold in case of deep RNN architectures. Remarkably, the work in [15] provides a relevant conceptual and practical tool for the definition, validity and usage of DeepESN in an "autonomous" (i.e., self-sufficient) way with respect to the standard ESN model.

Local Lyapunov Exponents. The study of DeepESN dynamics under a dynamical system perspective has been pursued in [28, 27], which provide a theoretical and practical framework for the study of stability of layered recurrent dynamics in terms of *local Lyapunov exponents*. The analysis along this research direction provided interesting insights in terms of the quality of the developed system dynamics, showing the natural beneficial effects due to layering. This aspect is graphically illustrated in Figure 4, which shows the maximum local Lyapunov exponent (MLLE) of reservoir systems at the increase of the total number of recurrent units, for the case of deep architectures (where the available recurrent units are arranged in layers of 10 units each) and shallow ones (where all the available recurrent units form a single layer). The MLLE is important to characterize the regime of dynamical stability of reservoir systems: values smaller than 0 denote a stable behavior, values greater than 0 indicate instability, and 0 identifies the *edge of stability* (or criticality) condition where dynamical recurrent systems are known to show richer representations of temporal data [44, 43]. As Figure 4 indicates, compared to shallow ESN settings in condition of equal number of recurrent units, DeepESNs consistently show higher values of the MLLE, closer to the edge of stability, i.e., richer dynamics. The natural enrichment of reservoir quality in deep reservoir settings has an interesting aftereffect in terms of short-term memory ability of the networks. Figure 5 shows the Memory Capacity (MC) score (as defined in [38]) achieved by DeepESN and shallow ESN under the same conditions considered for Figure 4, indicating the consistent improvement brought by the layered setting. The interested reader can find a more in-depth discussion on the MLLE, MC, and their relation in the context of DeepESNs in [28].

Design of Deep Recurrent Neural Networks. The study of the frequency spectrum of deep reservoirs enabled to address one of the fundamental open issues in deep learning, namely *how to choose the number of layers in a deep RNN architecture*. Starting from the analysis of the intrinsic differentiation of the filtering effects of successive levels in a stacked RNN architecture, the work in [24] proposed an automatic method for the

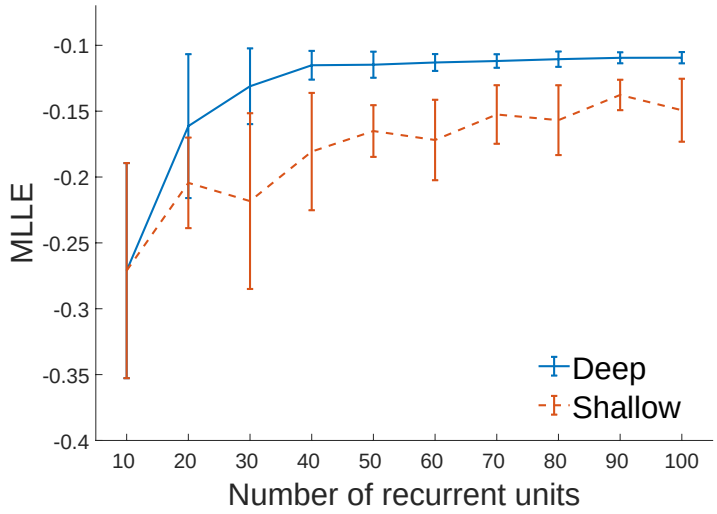

Fig. 4 MLLE of deep and shallow reservoir systems for increasing number of total recurrent units. In the deep case, the available reservoir units are organized into a layered architecture, with 10 units in each layer. In the shallow case, the available reservoir units are arranged into a shallow architecture with a single layer. The values of MLLE are computed as described in [28], using the same input time-series where individual elements were drawn from a uniform distribution in $[-0.8, 0.8]$. Results correspond to a simple network setting, with input scaling $\|\mathbf{W}^{(1)}\| = 1$, and in which all the reservoir layers share the same hyper-parametrization: spectral radius $\rho^{(i)} = 0.9$ (for all layers i) and inter-layer scaling $\|\mathbf{W}^{(i)}\| = 0.5$ (for layers $i > 1$). For each configuration, results are averaged (and standard deviation is computed) over 10 network guesses (with the same values of hyper-parameters, but different seed for random weights initialization).

design of DeepESNs. Noticeably, the proposed approach allows to tailor the DeepESN architecture to the characteristics of the input signals, consistently relieving the cost of the model selection process, and leading to new state-of-the-art results in speech and music processing tasks.

Deep Reservoirs for Structured Domains. A first extension of the deep RC framework for *learning in structured domains* has been presented in [20, 18], which introduced the Deep Tree Echo State Network (DeepTESN) model. The new model points out that it is possible to combine the concepts of deep learning, learning for trees and RC training efficiency, taking advantages from the layered architectural organization and from the compositionality of the structured representations both in terms of efficiency and in terms of effectiveness. In particular, experimental results in [20, 18] concretely demonstrate that deep RC models for trees can outperform the accuracy achieved by state-of-the-art approaches for learning in structured

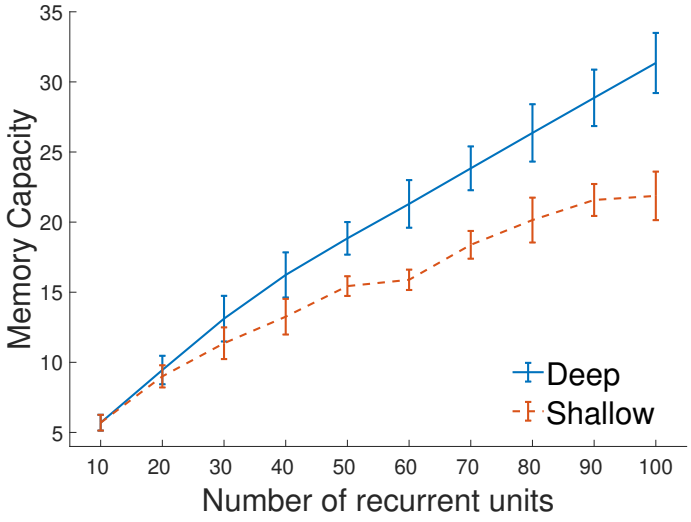

Fig. 5 Average test MC achieved by deep and shallow reservoir systems for increasing number of total recurrent units (the higher the better). In the deep case, the available reservoir units are organized into a layered architecture, with 10 units in each layer. In the shallow case, the available reservoir units are arranged into a shallow architecture with a single layer. The values of MC are computed using the same task settings reported in [28], and the same experimental settings considered for Figure 4.

domains in challenging problems in the areas of document processing and computational biology, at the same time being extremely advantageous in terms of required training times. Overall, DeepTESN provides a first instance of an extremely efficient approach for the design of deep neural networks for learning in cases where the input data is represented in the form of tree structures. Besides, from a theoretical perspective, the work in [20] also provides an in-depth analysis of asymptotic stability of untrained (non-linear) state transition systems operating on discrete tree structures. This results into a generalization of the ESP of conventional reservoirs, proposed under the name of Tree Echo State Property [20].

The Deep RC approach has been proved extremely advantageous also in the case of learning in domains of graph data, enabling the development of Fast and Deep Graph Neural Networks (FDGNNs) in [21]. The concept of reservoirs operating on discrete graph structures has been first introduced in [13], and revolves around the computation of a state embedding for each vertex in an input graph. In particular, the state for a vertex v is computed as a function of the input information attached to the vertex v itself (i.e., a vector of features that takes the role of external input in the system), and of the state computed for the neighbors of v (a concept that takes the role of "previous time-step" in the case of conventional RC

systems for time-series). The stability of the resulting dynamics can be studied by generalizing the mathematical means described in Section 2.2, leading to the definition of Graph Embedding Stability (GES), a stability property for neural embedding systems on graphs introduced in [21], to which the interested reader is referred for further information. Besides the introduction of GES, the work in [21] illustrates how to design a deep RC system for graphs, where each layer builds its embedding on the basis of the state information produced in the previous layer. The FDGNN approach was shown to reach (and even outperform) state-of-the-art accuracy on known benchmarks for graph classification, comparing well with many literature approaches, especially based on convolutional neural networks and kernel for graphs. Inheriting the easy of training algorithms from the RC paradigm, the approach is also extremely faster than literature models, enabling a sensible speed-up in the required training times (up to ≈ 3 orders of magnitude in the experiments reported in [21]).

Applications. For what regards the experimental analysis in *applications*, DeepESNs were shown to bring several advantages in both cases of synthetic and real-world tasks. Specifically, DeepESNs outperformed shallow reservoir architectures (under fair conditions on the number of total recurrent units and, as such, on the number of trainable readout parameters) on the Mackey-Glass next-step prediction task [19], on the short-term MC task [23, 11], on MSO tasks [26], as well as on a Frequency Based Classification task [24], purposely designed to assess multiple-frequency representation abilities. As pertains to *real-world problems*, the DeepESN approach recently proved effective in a variety of domains, including Ambient Assisted Living (AAL) [16], medical diagnosis [22], speech and polyphonic music processing [24, 25], metereological forecasting [1, 42], solar irradiance prediction [45], energy consumption and wind power generation prediction [36], short-term traffic forecasting [7], destination prediction [56] car parking and bike-sharing in urban computing [42], financial market predictions [42], and industrial applications (for blast furnace off-gas) [8, 5].

Software. *Software implementations* for DeepESNs applications have recently been made *publicly available*, with references [23, 24] representing citation requests for the use of the developed libraries. The DeepESN software is provided in the following forms:

- `Deep Echo State Network (DeepESN) MATLAB Toolbox`
  available through the MATLAB Add-On Explorer, and directly at the link `https://it.mathworks.com/matlabcentral/fileexchange/69402-deepesn`;
- `DeepESN Numpy Library (DeepESNpy)`
  available at `https://github.com/lucapedrelli/DeepESN`.

- `DeepRC TensorFlow Library (DeepRC-TF)`
  available at `https://github.com/gallicch/DeepRC-TF`.

4 Other Hierarchical Reservoir Computing Models

In this section we briefly summarize further developments in the field of hierarchical RC architectures, successive to the introduction of the DeepESN model. The works described below further contribute to provide an integrated view on the research lines attempting at bridging the gap between the areas of RC and deep learning.

Outside the ESN formalism, hierarchical modularity in reservoir models construction has been recently explored [65] in the strictly related field of Liquid State Machines [48], where reservoirs are implemented using spiking neural networks. In particular, results in [65] indicate that information propagation across the layers might be difficult in deep organizations of spiking neural networks, and a way to overcome this difficulty might consist in using specific patterns of inter-reservoir connectivity in the form of structural mappings with topographic projections between successive layers.

Hierarchical multi-layered architectures have been explored also in the context of reservoir systems implemented as cellular automata (CA) [61, 62], which can offer some potential advantages *per se*, e.g. in terms of further reductions of computational complexity (see e.g. [64]). In this context, the preliminary work in [50] shows promising results of 2-layered architectures with CA reservoirs applied to a 5-bit memorization task.

It is known that one of the potentially more costly aspect in the design of RC models is given by the optimization of reservoir hyper-parameters to the specificity of the task at hand. This kind of difficulty, already challenging in the case of shallow reservoirs [47], can be amplified when more reservoir layers are considered. An interesting research line in this concern is thus given by the application of evolutionary algorithms for the optimization of hyper-parameter values of hierarchical reservoirs. A first attempt in this direction is described in [6], where a steady-state genetic algorithm called Microbial GA [34] is adopted.

A further line of research involves the study of hybrid neural networks architectures that exploit the composition of deep models with shallow RC networks. An instance is represented by the recently introduced Deep Belief ESN [57], in which an ESN module is stacked on top of a Deep Belief Network (DBN). The DBN essentially operates a hierarchical non-linear transformation on the input data, while the final output is given by the (shallow) ESN layer. The components of the architecture are trained individually, employing a mixture of unsupervised and supervised learning in which the DBN is trained by contrastive divergence (following a greedy layer-wise approach), and the ESN is trained by pseudo-inversion. Finally, as generally analyzed in

the context of deep RNN construction [51], depth can enter the design of recurrent neural models is several disguises. In the field of RC, we foresee that the synergy between the benefits of modular compositionality both in the recurrent part (deep reservoir) and in the output part (deep readout) could result into breakthrough application results in challenging real-world tasks. A first step along this research direction has been pursued in [3], in which a bi-directional (shallow) reservoir network is coupled with a deep readout component, showing promising results both on benchmark datasets and on a real-world task in the area of medical diagnosis.

Finally, it is worth mentioning recent works that attempt at implementing the DeepESN concept in neuromorphic hardware, especially in photonics, see e.g. [46, 10].

5 Conclusions

In this chapter we have provided a brief overview of the extension of the RC approach towards the deep learning framework. Focusing the analysis in the context of discrete-time reservoir models, we have described the salient features of the DeepESN model. Noticeably, DeepESNs enable the analysis of the intrinsic properties of state dynamics in deep RNN architectures, i.e. the study of the bias due to layering in the design of RNNs. At the same time, DeepESNs allow to transfer the striking advantages of the RC methodology to the case of deep recurrent architectures, leading to an efficient approach for designing deep neural networks for temporal data.

The analysis of the distinctive characteristics and dynamical properties of the DeepESN model has been carried out first empirically, in terms of entropy of state dynamics and system memory. Then, it has been conducted through more abstract theoretical investigations that allowed the derivation of the fundamental conditions for the ESP of deep networks, as well as the characterization of the developed dynamical regimes in terms of local Lyapunov exponents. Besides, studies on the frequency analysis of DeepESN dynamics allowed the development of a grounded algorithm for the automatic setup of (the number of layers of) a DeepESN. Current developments already include model variants and applications to both synthetic and real-world tasks. Besides, extensions of the deep RC approach to learning in structured domains have been introduced with DeepTESN (for trees) and FDGNN (for graphs). Finally, a glimpse on the developments in the study of hierarchical RC-based models successive to the introduction of DeepESN has been provided.

The authors hope that the development of this kind of architectures and their extensions, in particular towards learning in structured domains, can foster the spreading of RC models both in research and applicative fields.

References

1. Alizamir, M., Kim, S., Kisi, O., Zounemat-Kermani, M.: Deep echo state network: a novel machine learning approach to model dew point temperature using meteorological variables. Hydrological Sciences Journal **65**(7), 1173–1190 (2020)
2. Angelov, P., Sperduti, A.: Challenges in deep learning. In: Proceedings of the 24th European Symposium on Artificial Neural Networks (ESANN), pp. 489–495. i6doc.com (2016)
3. Bianchi, F.M., Scardapane, S., Lokse, S., Jenssen, R.: Bidirectional deep-readout echo state networks. In: Proceedings of the 26th European Symposium on Artificial Neural Networks (ESANN), pp. 425–430 (2018)
4. Churchland, P.S., Sejnowski, T.J.: The computational brain. The MIT Press (1992)
5. Colla, V., Matino, I., Dettori, S., Cateni, S., Matino, R.: Reservoir computing approaches applied to energy management in industry. In: International Conference on Engineering Applications of Neural Networks, pp. 66–79. Springer (2019)
6. Dale, M.: Neuroevolution of hierarchical reservoir computers. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 410–417. ACM (2018)
7. Del Ser, J., Lana, I., Manibardo, E.L., Oregi, I., Osaba, E., Lobo, J.L., Bilbao, M.N., Vlahogianni, E.I.: Deep echo state networks for short-term traffic forecasting: Performance comparison and statistical assessment. arXiv preprint arXiv:2004.08170 (2020)
8. Dettori, S., Matino, I., Colla, V., Speets, R.: Deep echo state networks in industrial applications. In: IFIP International Conference on Artificial Intelligence Applications and Innovations, pp. 53–63. Springer (2020)
9. El Hihi, S., Bengio, Y.: Hierarchical recurrent neural networks for long-term dependencies. In: Advances in neural information processing systems (NIPS), pp. 493–499 (1996)
10. Freiberger, M., Sackesyn, S., Ma, C., Katumba, A., Bienstman, P., Dambre, J.: Improving time series recognition and prediction with networks and ensembles of passive photonic reservoirs. IEEE Journal of Selected Topics in Quantum Electronics **26**(1), 1–11 (2019)
11. Gallicchio, C.: Short-term Memory of Deep RNN. In: Proceedings of the 26th European Symposium on Artificial Neural Networks (ESANN), pp. 633–638 (2018)
12. Gallicchio, C., Martin-Guerrero, J.D., Micheli, A., Soria-Olivas, E.: Randomized machine learning approaches: Recent developments and challenges. In: Proceedings of the 25th European Symposium on Artificial Neural Networks (ESANN), pp. 77–86. i6doc.com (2017)
13. Gallicchio, C., Micheli, A.: Graph echo state networks. In: The 2010 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE (2010)
14. Gallicchio, C., Micheli, A.: Deep reservoir computing: A critical analysis. In: Proceedings of the 24th European Symposium on Artificial Neural Networks (ESANN), pp. 497–502. i6doc.com (2016)
15. Gallicchio, C., Micheli, A.: Echo state property of deep reservoir computing networks. Cognitive Computation **9**(3), 337–350 (2017)
16. Gallicchio, C., Micheli, A.: Experimental analysis of deep echo state networks for ambient assisted living. In: Proceedings of the 3rd Workshop on Artificial Intelligence for Ambient Assisted Living (AI*AAL 2017), co-located with the 16th International Conference of the Italian Association for Artificial Intelligence (AI*IA 2017) (2017)
17. Gallicchio, C., Micheli, A.: Deep Echo State Network (DeepESN): A Brief Survey. arXiv preprint arXiv:1712.04323 (2018)

18. Gallicchio, C., Micheli, A.: Deep Tree Echo State Networks. In: Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), pp. 499–506. IEEE (2018)
19. Gallicchio, C., Micheli, A.: Why layering in Recurrent Neural Networks? a DeepESN survey. In: Proceedings of the 2018 IEEE International Joint Conference on Neural Networks (IJCNN), pp. 1800–1807. IEEE (2018)
20. Gallicchio, C., Micheli, A.: Deep reservoir neural networks for trees. Information Sciences pp. 174–193 (2019)
21. Gallicchio, C., Micheli, A.: Fast and deep graph neural networks. In: Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20), pp. 3898–3905 (2020)
22. Gallicchio, C., Micheli, A., L.Pedrelli: Deep Echo State Networks for Diagnosis of Parkinson's Disease. In: Proceedings of the 26th European Symposium on Artificial Neural Networks (ESANN), pp. 397–402 (2018)
23. Gallicchio, C., Micheli, A., Pedrelli, L.: Deep reservoir computing: A critical experimental analysis. Neurocomputing **268**, 87–99 (2017). DOI https://doi.org/10.1016/j.neucom.2016.12.089
24. Gallicchio, C., Micheli, A., Pedrelli, L.: Design of Deep Echo State Networks. Neural Networks **108**, 33–47 (2018)
25. Gallicchio, C., Micheli, A., Pedrelli, L.: Comparison between DeepESNs and gated RNNs on multivariate time-series prediction. In: Proceedings of the 27th European Symposium on Artificial Neural Networks (ESANN), pp. 619–624 (2019)
26. Gallicchio, C., Micheli, A., Pedrelli, L.: Hierarchical temporal representation in linear reservoir computing. In: A. Esposito, M. Faundez-Zanuy, F.C. Morabito, E. Pasero (eds.) Neural Advances in Processing Nonlinear Dynamic Signals, pp. 119–129. Springer International Publishing, Cham (2019). DOI 10.1007/978-3-319-95098-3_11. ArXiv preprint arXiv:1705.05782
27. Gallicchio, C., Micheli, A., Silvestri, L.: Local lyapunov exponents of deep rnn. In: Proceedings of the 25th European Symposium on Artificial Neural Networks (ESANN), pp. 559–564. i6doc.com (2017)
28. Gallicchio, C., Micheli, A., Silvestri, L.: Local lyapunov exponents of deep echo state networks. Neurocomputing **298**, 34–45 (2018)
29. Gallicchio, C., Micheli, A., Tiňo, P.: Randomized Recurrent Neural Networks. In: Proceedings of the 26th European Symposium on Artificial Neural Networks (ESANN), pp. 415–424. i6doc.com (2018)
30. Gallicchio, C., Scardapane, S.: Deep randomized neural networks. In: Recent Trends in Learning From Data, pp. 43–68. Springer (2020)
31. Gerstner, W., Kistler, W.M.: Spiking neuron models: Single neurons, populations, plasticity. Cambridge university press (2002)
32. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT press (2016)
33. Graves, A., Mohamed, A.R., Hinton, G.: Speech recognition with deep recurrent neural networks. In: IEEE International Conference on Acoustics, speech and signal processing (ICASSP), pp. 6645–6649. IEEE (2013)
34. Harvey, I.: The microbial genetic algorithm. In: European Conference on Artificial Life, pp. 126–133. Springer (2009)
35. Hermans, M., Schrauwen, B.: Training and analysing deep recurrent neural networks. In: NIPS, pp. 190–198 (2013)
36. Hu, H., Wang, L., Lv, S.X.: Forecasting energy consumption and wind power generation using deep echo state network. Renewable Energy **154**, 598–613 (2020)
37. Jaeger, H.: The "echo state" approach to analysing and training recurrent neural networks - with an erratum note. Tech. rep., GMD - German National Research Institute for Computer Science, Tech. Rep. (2001)
38. Jaeger, H.: Short term memory in echo state networks. Tech. rep., German National Research Center for Information Technology (2001)

39. Jaeger, H.: Discovering multiscale dynamical features with hierarchical echo state networks. Tech. rep., Jacobs University Bremen (2007)
40. Jaeger, H., Haas, H.: Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. Science **304**(5667), 78–80 (2004)
41. Jaeger, H., Lukoševičius, M., Popovici, D., Siewert, U.: Optimization and applications of echo state networks with leaky-integrator neurons. Neural Networks **20**(3), 335–352 (2007)
42. Kim, T., King, B.R.: Time series prediction using deep echo state networks. Neural Computing and Applications pp. 1–19 (2020)
43. Legenstein, R., Maass, W.: Edge of chaos and prediction of computational performance for neural circuit models. Neural networks **20**(3), 323–334 (2007)
44. Legenstein, R., Maass, W.: What makes a dynamical system computationally powerful. New directions in statistical signal processing: From systems to brain pp. 127–154 (2007)
45. Li, Q., Wu, Z., Ling, R., Feng, L., Liu, K.: Multi-reservoir echo state computing for solar irradiance prediction: A fast yet efficient deep learning approach. Applied Soft Computing **95**, 106481 (2020)
46. Lugnan, A., Katumba, A., Laporte, F., Freiberger, M., Sackesyn, S., Ma, C., Gooskens, E., Dambre, J., Bienstman, P.: Photonic neuromorphic information processing and reservoir computing. APL Photonics **5**(2), 020901 (2020)
47. Lukoševičius, M., Jaeger, H.: Reservoir computing approaches to recurrent neural network training. Computer Science Review **3**(3), 127–149 (2009)
48. Maass, W., Natschläger, T., Markram, H.: Real-time computing without stable states: A new framework for neural computation based on perturbations. Neural computation **14**(11), 2531–2560 (2002)
49. Malik, Z.K., Hussain, A., Wu, Q.J.: Multilayered echo state machine: a novel architecture and algorithm. IEEE Transactions on cybernetics **47**(4), 946–959 (2017)
50. Nichele, S., Molund, A.: Deep learning with cellular automaton-based reservoir computing. Complex Systems **26**, 319–340 (2017)
51. Pascanu, R., Gulcehre, C., Cho, K., Bengio, Y.: How to construct deep recurrent neural networks. arXiv preprint arXiv:1312.6026v5 (2014)
52. Scardapane, S., Wang, D.: Randomness in neural networks: an overview. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **7**(2), e1200 (2017)
53. Schmidhuber, J.: Learning complex, extended sequences using the principle of history compression. Neural Computation **4**(2), 234–242 (1992)
54. Schmidhuber, J.: Deep learning in neural networks: An overview. Neural networks **61**, 85–117 (2015)
55. Schrauwen, B., Wardermann, M., Verstraeten, D., Steil, J., Stroobandt, D.: Improving reservoirs using intrinsic plasticity. Neurocomputing **71**(7), 1159–1171 (2008)
56. Song, Z., Wu, K., Shao, J.: Destination prediction using deep echo state network. Neurocomputing **406**, 343–353 (2020)
57. Sun, X., Li, T., Li, Q., Huang, Y., Li, Y.: Deep belief echo-state network and its application to time series prediction. Knowledge-Based Systems **130**, 17–29 (2017)
58. Triefenbach, F., Jalalvand, A., Demuynck, K., Martens, J.P.: Acoustic modeling with hierarchical reservoirs. IEEE Transactions on Audio, Speech, and Language Processing **21**(11), 2439–2450 (2013)
59. Triefenbach, F., Jalalvand, A., Schrauwen, B., Martens, J.P.: Phoneme recognition with large hierarchical reservoirs. In: Advances in neural information processing systems, pp. 2307–2315 (2010)

60. Verstraeten, D., Schrauwen, B., d'Haene, M., Stroobandt, D.: An experimental unification of reservoir computing methods. Neural networks **20**(3), 391–403 (2007)
61. Von Neumann, J., Burks, A.W.: Theory of self-reproducing automata. University of Illinois Press Urbana (1996)
62. Wolfram, S.: Universality and complexity in cellular automata. Physica D: Nonlinear Phenomena **10**(1-2), 1–35 (1984)
63. Yildiz, I.B., Jaeger, H., Kiebel, S.J.: Re-visiting the echo state property. Neural networks **35**, 1–9 (2012)
64. Yilmaz, O.: Reservoir computing using cellular automata. arXiv preprint arXiv:1410.0162 (2014)
65. Zajzon, B., Duartel, R., Morrison, A.: Transferring state representations in hierarchical spiking neural networks. In: Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), pp. 1785–1793. IEEE (2018)