ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

# Enhancing Echo State Networks with Gradient-based Explainability Methods

Francesco Spinnato[1,2], Andrea Cossu[1], Riccardo Guidotti[1,2], Andrea Ceni[1], Claudio Gallicchio[1], and Davide Bacciu[1] *

1 - University of Pisa, Italy

2 - ISTI-CNR Pisa, Italy

**Abstract**. Recurrent Neural Networks are effective for analyzing temporal data, such as time series, but they often require costly and time-intensive training. Echo State Networks simplify the training process by using a fixed recurrent layer, the reservoir, and a trainable output layer, the readout. In sequence classification problems, the readout typically receives only the final state of the reservoir. However, averaging all states can sometimes be beneficial. In this work, we assess whether a weighted average of hidden states can enhance the Echo State Network performance. To this end, we propose a gradient-based, explainable technique to guide the contribution of each hidden state towards the final prediction. We show that our approach outperforms the naive average, as well as other baselines, in time series classification, particularly on noisy data.

## 1 Introduction

Deep Neural Networks (DNNs) have emerged in various fields like medical diagnosis and financial forecasting, yet their adoption is limited by their opacity, which affects the trust in their outputs and understanding of their decision processes. To address these issues, Explainable Artificial Intelligence (XAI) aims to make the workings of complex DNNs transparent and understandable [1]. XAI methods help clarify how DNNs operate, fostering transparency, accountability, and compliance, which is vital in sensitive applications [2]. While traditionally focused on improving model transparency and interpretability, recent studies suggest XAI can also enhance learning processes in these models [3].

This work focuses on Recurrent Neural Networks (RNNs), which are effective for learning from temporal data, like time series [4]. Unfortunately, the practical deployment of RNNs often entails significant computational resources and time investments for training, rendering them less appealing in scenarios where efficiency is paramount. Echo State Networks (ESNs) are reservoir computing models that offer a more sustainable alternative for processing temporal data [5], with a simplified training, characterized by a fixed recurrent component called *reservoir* and a trained output layer called *readout*. Typically, in time-series classification applications, the ESN's readout layer exploits only the reservoir's final

*Work supported by: EU EIC project EMERGE (Grant No. 101070918), EU NextGenerationEU programme under the funding schemes PNRR-PE-AI FAIR (Future Artificial Intelligence Research), PNRR-SoBigData.it - Strengthening the Italian RI for Social Mining and Big Data Analytics - Prot. IR13, H2020-INFRAIA-2019-1: Res. Infr. G.A. 871042 *SoBigData++*, G.A. 761758 *Humane AI*, G.A. 952215 *TAILOR*, and ERC-2018-ADG G.A. 834756 *XAI*.

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

state to generate predictions, or uses information from the entire reservoir state trajectory in the state space, e.g., by averaging all its states. Both approaches have limitations and can result in performance degradation, particularly in noisy scenarios, where the relevant information is neither at the end of the time-series nor distributed uniformly across the entire signal.

In this paper, we investigate the usage of gradient-based, post-hoc explainability methods for enhancing the performance of ESNs in time-series classification tasks. Typically, gradient-based XAI methods are employed to determine the relevance of each point in the time series for the classification outcome [6]. We adopt them in this work to produce a weighted average of the hidden states of the ESN, so that the readout receives more relevant information about the driving input signal. We benchmark our proposal on 30 binary classification datasets from the time series UCR repository, employing various XAI techniques to weigh the hidden states. We assess the performance on the original datasets and also in the presence of increasingly noisier data.

## 2 Setting The Stage

Given a time series dataset $\mathbf{X} \in \mathbb{R}^{N \times T}$, containing $N$ univariate time series, $\mathbf{x} = [x_1, \ldots, x_T]$, of length $T$, and respective labels, $\mathbf{y} \in \{0, 1\}^N$, we tackle the problem of binary classification. The objective is to train a model $f$ that maps the input time series to a predicted class label, $\hat{y} = f(\mathbf{x})$.

In our case, $f$ is an ESN. The ESN updates its hidden state according to $\mathbf{h_t} = \sigma(W\mathbf{h_{t-1}} + Vx_t + \mathbf{b})$, where $\sigma$ is a nonlinear activation function (e.g., hyperbolic tangent), $x_t$ is the input at time $t$, $W$ and $V$ are the recurrent and input matrixes, respectively, and $\mathbf{b}$ is the bias. Following the reservoir computing paradigm, the ESN initializes $W$ randomly and then scales its spectral radius by dividing all elements by a constant $\rho$ (treated as a hyper-parameter). Similarly, the matrix $V$ is scaled using a scalar $\nu$ (the input scaling). Usually, the last hidden state output, $\mathbf{h}_T$, is forwarded to the readout component of the ESN, which is the only trained part of the network. The readout typically involves a linear transformation followed by a nonlinear activation function, depending on the task. For binary classification $\hat{y} = \sigma(W_{\text{out}}\mathbf{h}' + \mathbf{b}_{\text{out}})$, where $W_{\text{out}}$ represents the weights of the readout layer, $\mathbf{b}_{\text{out}}$ is the bias, and $\sigma$ is a sigmoid activation function. Since the recurrent transformation processed by the reservoir is fixed, the readout can be trained in closed form (e.g., Ridge regression).

In this work, we use gradient-based XAI methods to infer the importance of points in the time series. These techniques essentially perform sensitivity analysis, i.e., they can highlight the importance of each observation in the time series. The foundation of these approaches is the analysis of the gradient of the network's output w.r.t. an input instance, as it informs about the sensitivity of the output to the input features. XAI approaches differ in how they exploit this gradient, and in other information they use [6]. However, their output is always a vector $\boldsymbol{\phi}$, containing the importance of observations in the input.

We focus on three approaches: *(i) Gradient*, *(ii) Gradient*Input*, and *(iii)*

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

*GradientSHAP*. The simplest approach, *Gradient*, computes the gradient of the output w.r.t. each observation of the input time series, i.e., $\boldsymbol{\phi} = [\frac{\partial \hat{y}}{\partial x_1}, \ldots, \frac{\partial \hat{y}}{\partial x_T}]$. Second, *Gradient*Input* [7] multiplies the gradient with the input, i.e., $\boldsymbol{\phi} = [\frac{\partial \hat{y}}{\partial x_1} x_1, \ldots, \frac{\partial \hat{y}}{\partial x_T} x_T]$. Geometrically, *Gradient*Input* is the directional derivative computed along the direction pointed by the input. For multivariate inputs, *Gradient*Input* can vanish whenever the current input points towards a direction orthogonal to the direction pointed by the gradient, even if the input itself has a large magnitude. Finally, we also use *GradientSHAP* [8], a faster, approximated version of Integrated Gradients [9], which introduces a baseline, i.e., a point of reference from which the contribution of each feature to the prediction can be measured. By comparing the model's predictions on actual input data to its predictions on baseline data, GradientSHAP can infer the impact of each feature. In this case, $\boldsymbol{\phi} = [\frac{\partial \hat{y}}{\partial x_1}(x_1 - z_1), \ldots, \frac{\partial \hat{y}}{\partial x_T}(x_T - z_T)]$, where $\mathbf{z}$ is the baseline vector, which can be user-defined or obtained through random sampling.

## 3 Enhancing Echo State Networks

To enhance the performance of ESNs, we propose weighting the reservoir output, based on the importance of each observation in $\mathbf{x}$. For this purpose, we introduce a generalized method for aggregating hidden states, formally:

$$\mathbf{h}' = \sum_{t=1}^{T} \phi_t \mathbf{h}_t, \quad \text{with} \sum_{t=1}^{T} \phi_t = 1, \tag{1}$$

where $\mathbf{h}_t$ is the hidden state at time $t$, $\phi_t$ is a scalar weight assigned to each hidden state, and $\mathbf{h}'$ is the aggregated output. Equation (1) generalizes the standard cases. Consider, for example, a weight vector $\boldsymbol{\phi}$ consisting solely of zeros except for the final element, which is one, i.e., $\boldsymbol{\phi} = [0, 0, \ldots, 0, 1]$. In this scenario, the aggregation process ignores all hidden states except for the last, $\mathbf{h}_T$. Alternatively, if $\boldsymbol{\phi} = \left[\frac{1}{T}, \frac{1}{T}, \ldots, \frac{1}{T}\right]$, i.e., if we assign an equal weight of $\frac{1}{T}$ to each hidden state, the aggregation method transitions to computing the average of all hidden states. Beyond these specific scenarios, any other configuration of the weights produces a weighted average.

Our intuition is that selecting only the hidden states corresponding to the most relevant parts of the time series can yield better performance than simpler methods such as averaging or using only the last hidden state. Consider, for example, Fig. 1. On the top left, we have a time series, $\mathbf{x}$, with some noise at the beginning and the end. Right below (bottom left), we have the respective hidden states, 100, for each time series point, colored based on their value (low values in yellow, high values in violet). Intuitively, taking only the last hidden state would not be the best choice, as it contains information from approximately 300 noisy points. At the same time, taking the average of the hidden states would dilute the information contained in the hidden states, given that approximately half of this time series is composed of Gaussian noise.

The issue, therefore, is how to find the weights $\phi_i$ without an iterative optimization, as that would negate the primary advantages of ESNs, i.e., that

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

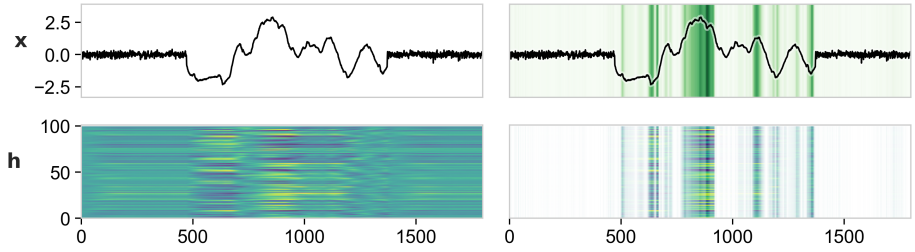


Fig. 1: Top left: a time series from the *WormsTwoClass* dataset with some Gaussian noise. Bottom left: the raw hidden states for each timestep. Top right: the importance of each timestep of the input time series from *Gradient*Input*. Bottom right: the hidden states weighted by their importance.

only the readout necessitates training. We propose inferring these weights using XAI methods in a one-shot fashion, i.e., train a *base model*, get the weights, $\boldsymbol{\phi}$, using an XAI technique from Section 2, and retrain the readout only once. Given that we are interested in the magnitude of the importance vector and not its sign, we take its absolute value and normalize it so that it sums to one, formally:

$$\boldsymbol{\phi} = \frac{[|\phi_1|, \ldots, |\phi_T|]}{\sum_{i=1}^{T} |\phi_i|}. \tag{2}$$

Fig. 1 on the top right shows the importance vector extracted by *Gradient*Input* (the more intense the color, the more important the observation), and on the bottom right, the hidden states weighted by the importance vector.

## 4 Experiments

We benchmark our approach on 30 datasets from the UCR time series classification repository[1]. As baselines, we test ESN using the last hidden states (LAST), taking the average (AVG), randomly selecting weights (RND), and calculating the weighted average using 3 XAI approaches: *Gradient* (GRAD), *Gradient*Input* [7] (GRAD*INP), and *GradientSHAP* (GRADSHAP) [8]. As absolute performance is not the focus of this work, and to have a fair comparison between models, we leave the parameters of the ESN fixed at standard values, i.e., 100 units, a spectral radius of 0.99, and an input scaling of 1, with a leak rate of 0.01 [10]. To determine the importance of the points in the time series, we need a *base model* to retrieve them. For this purpose, we use AVG, given that it is the most stable baseline against noise, as shown in the following experiments. Once $\boldsymbol{\phi}$ is retrieved, we use it to retrain the readout on the weighted average of hidden states. We use LBFGS as the readout optimizer and set the maximum number of iterations to 1000.

[1]Code and datasets are available at `https://github.com/fspinna/xai_enhanced_esn`.

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

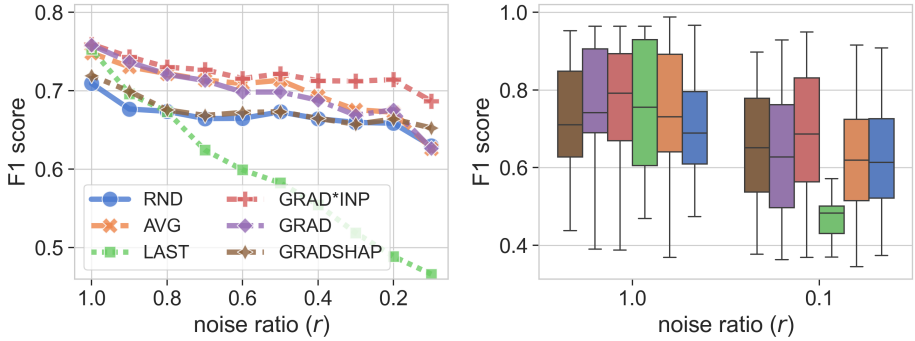


Fig. 2: Left: average F1 of the benchmarked models for all 30 UCR datasets, starting from no added noise ($r = 1.0$), to 10x noise ($r = 0.1$). Right: boxplot for the two extreme cases, $r = 1.0$ and $r = 0.1$.

In Table 1 and Fig. 2, we report the performance in terms of the F1-score for all approaches, first on the original datasets and then by adding random Gaussian noise to each time series. The amount of noise is controlled by the noise ratio $0 \leq r \leq 1$, and the total amount of added noise is computed as $T^{noise} = \frac{T}{r} - T$. In our tests, we varied $r$ from 0.1 (noise that is 10x the length of the original time series) to 1 (no noise) in 0.1 increments. Time series are standardized through Z-score normalization. Each run is repeated 3 times, and the average on all the datasets with the standard deviation is reported[2].

In general, the best approach is GRAD*INP, both on the original datasets without noise, i.e., $r = 1.0$, and on noisy datasets. On the original datasets, AVG, LAST, and GRAD are comparable or slightly worse than GRAD*INP. However, LAST's performance drops quickly as the noise increases, while AVG and GRAD follow a less steep trend. Interestingly, their performances become comparable to random weighting as $r$ approaches 0.1 (10x noise). Although GRADSHAP is a more sophisticated approach, it performs worse than AVG and GRAD. This is likely due to its random baseline extraction, which could be subpar on time series data. In summary, GRAD*INP is more robust, both on the original data, and in the presence of noise, while its computational overhead is minimal.

Given the results of these experiments, an interesting insight is that the machine learning model seems to benefit from a form of *self-awareness*, i.e., using its own explanation in its learning process. In other words, the ESN can correct itself without human supervision but using a tool originally meant to be used by human users. In a way, the ESN exploits insights into its decision-making, enabling better adaptation to noisy data and improving its overall performance.

[2]System: Lenovo SD650 nodes, with Intel Xeon Platinum 8268 CPUs, 64GB Memory.

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

| $r$ | RND | AVG | LAST | GRAD | GRAD*INP | GRADSHAP |
|---|---|---|---|---|---|---|
| 1.0 | .71 ± .14 | .75 ± .16 | .75 ± .16 | **.76 ± .16** | **.76 ± .16** | .72 ± .15 |
| 0.5 | .67 ± .14 | .71 ± .15 | .58 ± .12 | .70 ± .14 | **.72 ± .14** | .67 ± .13 |
| 0.1 | .63 ± .16 | .63 ± .17 | .47 ± .05 | .63 ± .16 | **.69 ± .17** | .65 ± .15 |

Table 1: Mean F1 and standard deviation of each method on all the UCR datasets, with no noise ($r = 1.0$), 2x noise ($r = 0.5$), and 10x noise ($r = 0.1$). Higher is better, best values in bold.

## 5 Conclusion

In this work, we have proposed weighting the hidden state of an ESN with post-hoc gradient-based XAI methods to enhance accuracy and noise robustness. We have pursued a paradigm shift in the role of XAI from being purely interpretative to actively contributing to the learning process of a model. This is promising for advancing the capabilities of AI systems as it connects the model and the explanations it generates, leading to enhanced learning, adaptability, and, ultimately, better performance in real-world applications. A limitation of this approach is that, for now, it works only for binary classification. For future work, we plan on tackling the multiclass problem. This is not a naive endeavor, as it requires considering the gradient of multiple output nodes with respect to the time series input. Finally, we plan on exploring different tasks, such as time series regression, to assess if this approach extends across other problems.

## References

[1] Riccardo Guidotti et al. A survey of methods for explaining black box models. *ACM CSUR*, 51(5):93:1–93:42, 2019.

[2] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artif. Intell.*, 267:1–38, 2019.

[3] Andrea Apicella et al. Strategies to exploit XAI to improve classification systems. In *xAI (1)*, volume 1901 of *CCIS*, pages 147–159. Springer, 2023.

[4] Larry R Medsker, Lakhmi Jain, et al. Recurrent neural networks. *Design and Applications*, 5(64-67):2, 2001.

[5] Filippo Maria Bianchi, Simone Scardapane, Sigurd Løkse, and Robert Jenssen. Reservoir computing approaches for representation and classification of multivariate time series. *IEEE transactions on neural networks and learning systems*, 32(5):2169–2179, 2020.

[6] Andreas Theissler et al. Explainable AI for time series classification: A review, taxonomy and research directions. *IEEE Access*, 10:100700–100724, 2022.

[7] Avanti Shrikumar et al. Not just a black box: Learning important features through propagating activation differences. *CoRR*, abs/1605.01713, 2016.

[8] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. pages 4765–4774, 2017.

[9] Mukund Sundararajan et al. Axiomatic attribution for deep networks. In *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328. PMLR, 2017.

[10] Mantas Lukoševičius. A practical guide to applying echo state networks. In *Neural Networks: Tricks of the Trade: Second Edition*, pages 659–686. Springer, 2012.