UCLouvain

epl

École polytechnique de Louvain

Hardware-compatible and energy-efficient neural network for solving machine learning tasks

Author: **Aurian David**
Supervisor: **Araujo Abreu Flavio**
Readers: **Anatole Moureaux, Martin Andraud**
Academic year 2023–2024
Master [120] in Electrical Engineering

Acknowledgment

I would like to express my deepest gratitude to my supervisor, Professor Flavio Abreu Araujo, as well as my mentor Anatole Moureaux, for their invaluable guidance and support throughout this project. I am especially thankful to Anatole for his unwavering availability and willingness to help whenever I had questions, which significantly contributed to the progress and completion of this work.

I would also like to thank Simon de Wergifosse for the insightful discussions and knowledge he shared regarding the workings of Spin Torque Vortex Oscillators (STVOs). His expertise greatly enhanced my understanding of the subject.

Finally, I want to extend my heartfelt thanks to my family for their constant encouragement and support, which has been a source of strength for me throughout my studies.

Abstract

This study explores the fusion of machine learning, particularly reservoir computing networks (RCNs), with spintronics to achieve energy-efficient hardware implementations. The focus is on utilizing Spin Torque Vortex Oscillators (STVOs) as "neurons" within a reservoir computing network. By leveraging the unique properties of STVOs, the research aims to assess their performance in comparison to other hardware-implemented and conventional reservoir computing networks. The investigation includes a comprehensive analysis of STVO-specific parameters and their impact on the overall system's efficiency and computational capability. This work contributes to the advancement of energy-efficient machine learning hardware, demonstrating the potential of STVOs in enhancing the performance of reservoir computing networks.

Abbreviations List

Abbreviation	Meaning
DC	Direct Current
ESN	Echo State Network
LSM	Liquid State Machines
MC	Memory Capacity
NARMA	Nonlinear Autoregressive-Moving Average
NMSE	Normalized Mean Square Error
RC(N)	Reservoir Computing (Network)
RNN	Recurent Neural Network
STM(C)	Short Term Memory (Capacity)
STT	Spin Torque Transfer
STVO	Spin Torque Vortex Oscillator
SVM	Support Vector Machine

Table of Contents

Introduction

The high current energy cost of machine learning, driven by the exponential growth in data and model complexity, has become a significant concern for both researchers and industry practitioners. Training and deploying sophisticated machine learning models, particularly deep neural networks, require substantial computational resources, leading to increased energy consumption and operational costs. This challenge has spurred interest in developing hardware-implemented machine learning solutions aimed at improving energy efficiency without compromising performance.

Reservoir computing (RC), which includes Echo State Networks (ESNs) and Liquid State Machines (LSMs), offers a promising approach for energy-efficient machine learning. RC is particularly well-suited for temporal information processing and pattern recognition tasks. One of the key advantages of reservoir computing networks (RCNs) is their simplicity in training. Unlike traditional recurrent neural networks (RNNs) that require complex and computationally intensive training algorithms like backpropagation through time, RCNs only require a linear regression on the output of the network. This is because the reservoir, which is the core of the network, remains fixed after random initialization. Only the readout layer, which maps the high-dimensional reservoir states to the desired output, is trained. This simplification not only reduces the computational burden but also makes the hardware implementation of RCNs more feasible and efficient.

This study investigates the integration of STVOs into reservoir computing networks, aiming to harness their unique properties for energy-efficient machine learning. By employing STVOs as "neurons" in a reservoir computing architecture, we seek to evaluate their performance against conventional reservoir computing implementations. This includes a detailed examination of STVO-specific parameters and their influence on the overall system's effectiveness.

The paper is structured as follows: We begin with an overview of the state-of-the-art in reservoir computing, highlighting key developments and existing challenges. This is followed by an introduction to spintronics and the role of STVOs in computational applications. We then present the architecture of our proposed model, detailing how STVOs are integrated into the reservoir computing framework.

Subsequently, perform a short analysis of the qualitative effects expressed by the different STVO-parameters before exploring the concept of short-term memory (STM) capacity, a critical metric for evaluating the temporal processing capabilities of reser-

voir computing networks. We provide benchmarks on well-known tasks such as the NARMA2 and Mackey-Glass time series, comparing the performance of STVO-based networks with other hardware and software implementations. To further illustrate the practical applications of our approach, we apply our model to the prediction of sunspot numbers, a complex and nonlinear time series problem.

The findings from our study underscore the potential of STVOs in enhancing the energy efficiency and computational prowess of reservoir computing networks, paving the way for more sustainable and scalable machine learning solutions. The paper concludes with a discussion of the implications of our results and directions for future research.

It is important to note that many of our observations are supported by detailed data presented in the annex. We provide references to the annex throughout the document whenever applicable. This approach is intended to avoid encumbering the main document with a large number of graphs and tables, ensuring that the main text remains clear and focused. By including the detailed data in the annex, we provide the necessary evidence to support our findings while maintaining the readability of the main document.

Finally, we need to inform you that, generative AI was used as an aid to write some of the content of this document, especially to speed up the introductions to some of the chapters.

State of the art and theoretical background

2.1 Reservoir computing

2.1.1 Introduction

Reservoir computing is a computational framework designed to harness the dynamic properties of recurrent neural networks for time-series data processing and other tasks involving temporal dependencies. Introduced in the early 2000s, RC has gained significant attention due to its simplicity in training and its ability to perform complex temporal tasks efficiently. The key components of RC include Echo State Networks and Liquid State Machines, both of which leverage the inherent dynamics of randomly connected networks to project input data into high-dimensional spaces.

2.1.2 Historical Background and Foundations

The concept of reservoir computing emerged independently from two different perspectives: echo state networks introduced by Herbert Jaeger in 2001 [1] and liquid state machines proposed by Wolfgang Maass, Thomas Natschläger, and Henry Markram in 2002 [2]. Both models share a common principle: using a fixed, randomly connected recurrent network (the reservoir) to transform the input signal into a high-dimensional representation, while only training the output weights to perform the desired task.

- **Echo State Networks**:
  ESNs utilize a reservoir with fixed weights and a simple linear readout layer. The primary requirement for an ESN is the echo state property, which ensures that the influence of previous inputs on the reservoir state diminishes over time, allowing the network to forget past inputs and remain stable. The main advantages of ESNs are their computational efficiency and ease of training, as only the output weights require adjustment.@ jaeger2001echo

- **Liquid State Machines**:
  LSMs, also known as spiking neural networks, consist of a randomly connected network of spiking neurons. The network's dynamic response to input signals generates a high-dimensional temporal representation, which is then read out by a simple linear classifier. LSMs are particularly well-suited for processing real-time data streams and have applications in robotics and sensory processing.@ maass2002real

2.1.3 Theoretical Foundations

The theoretical basis for RC lies in the ability of recurrent networks to create a high-dimensional mapping of input signals, which facilitates linear separability of complex temporal patterns. Key theoretical concepts include:

- **High-dimensional Projection**: The reservoir's recurrent dynamics project the input data into a high-dimensional space, where linear regression can effectively separate different patterns. This is akin to the kernel trick used in support vector machines (SVMs), where nonlinear data is mapped to a higher-dimensional space to make it linearly separable. [3]
- **Echo State Property**: For an ESN to function correctly, it must exhibit the echo state property, which ensures that the reservoir's state is uniquely determined by the input history. This property is essential for the stability and reliability of the network. [1]
- **Separation Property**: In LSMs, the separation property states that different input signals should generate distinct trajectories in the reservoir state space. This property is crucial for the network's ability to distinguish between different input patterns. [2]

2.1.4 Mathematical Description of Reservoir Computing

The dynamics of a reservoir computing network can be described mathematically as follows:

Let $\boldsymbol{u}(t) \in \mathbb{R}^N$ be the input vector at time t, $\boldsymbol{x}(t) \in \mathbb{R}^M$ be the state of the reservoir at time t, and $\boldsymbol{y}(t) \in \mathbb{R}^K$ be the output vector at time t. The update equations for the reservoir and the output can be written as:

- **Reservoir State Update**:

$$\boldsymbol{x}(t+1) = f(\boldsymbol{W}_{\text{res}}\boldsymbol{x}(t) + \boldsymbol{W}_{\text{in}}\boldsymbol{u}(t) + \boldsymbol{b}) \tag{1}$$

  Here, $\boldsymbol{W}_{\text{res}} \in \mathbb{R}^{M \times M}$ is the recurrent weight matrix of the reservoir, $\boldsymbol{W}_{\text{in}} \in \mathbb{R}^{M \times N}$ is the input weight matrix, $\boldsymbol{b} \in \mathbb{R}^M$ is the bias vector, and $f(\cdot)$ is a nonlinear activation function, such as the hyperbolic tangent (tanh) or the sigmoid function.
  Later the parameter α was introduced as the **leaking rate** [4], which controls the

speed at which the reservoir updates its state and thus affects the memory of the system such that the reservoir state update becomes :

$$\boldsymbol{x}(t+1) = (1-\alpha)\boldsymbol{x}(t) + \alpha f(\boldsymbol{W}_{\text{res}}\boldsymbol{x}(t) + \boldsymbol{W}_{\text{in}}\boldsymbol{u}(t) + \boldsymbol{b}) \tag{2}$$

- **Spectral Radius**: The **spectral radius** $\rho(\boldsymbol{W}_{\text{res}})$ of the reservoir weight matrix $\boldsymbol{W}_{\text{res}}$ is a crucial parameter that determines the dynamic regime of the reservoir. It is defined as the largest absolute value of the eigenvalues of $\boldsymbol{W}_{\text{res}}$. A smaller spectral radius favors a more stable dynamic while a larger spectral radius favors a more chaotic behavior. To ensure the echo state property, it is typically set such that $\rho(\boldsymbol{W}_{\text{res}}) < 1$. In practice, a random search algorithm is the most robust way to find the optimal **spectral radius** for a precise task. [5]
- **Output Calculation**:

$$\boldsymbol{y}(t) = \boldsymbol{W}_{\text{out}}\boldsymbol{x}(t) \tag{3}$$

  Here, $\boldsymbol{W}_{\text{out}} \in \mathbb{R}^{K\times M}$ is the output weight matrix. In reservoir computing, only $\boldsymbol{W}_{\text{out}}$ is trained, typically using linear regression, while $\boldsymbol{W}_{\text{res}}$ and $\boldsymbol{W}_{\text{in}}$ are fixed and randomly initialized. [3]
- **Ridge Regression for Training**: Training the output weights $\boldsymbol{W}_{\text{out}}$ is typically performed using ridge regression, which is a regularized version of linear regression. Given a set of training data $(\boldsymbol{X}, \boldsymbol{Y})$, where $\boldsymbol{X} = [\boldsymbol{x}(1), \boldsymbol{x}(2), ..., \boldsymbol{x}(T)]^T$ and $\boldsymbol{Y} = [\boldsymbol{y}(1), \boldsymbol{y}(2), ..., \boldsymbol{y}(T)]^T$, the ridge regression solution for $\boldsymbol{W}_{\text{out}}$ is given by:

$$\boldsymbol{W}_{\text{out}} = \left(\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I}\right)^{-1}\boldsymbol{X}^T\boldsymbol{Y} \tag{4}$$

  Here, λ is the regularization parameter, and $\boldsymbol{I}$ is the identity matrix. The regularization term $\lambda\boldsymbol{I}$ helps to prevent overfitting by penalizing large weights. [6]

2.1.5 Applications of Reservoir Computing

Reservoir computing has been applied successfully across a wide range of domains, including:

- **Time-Series Prediction**: ESNs have been widely used for tasks such as financial forecasting, climate modeling, and speech recognition. Their ability to capture temporal dependencies makes them suitable for predicting future values based on past data. [7]
- **Robotics**: LSMs have been employed in robotic control systems to process sensory inputs and generate appropriate motor responses. Their real-time processing capabilities make them ideal for dynamic environments.[2]
- **Neuromorphic Computing**: Reservoir computing models, particularly LSMs, have been integrated into neuromorphic hardware to mimic biological neural systems. This approach leverages the energy efficiency and parallel processing capabilities of spiking neural networks.

- **Pattern Recognition**: ESNs and LSMs have been used for various pattern recognition tasks, including handwriting recognition, image classification, and anomaly detection. The high-dimensional representations generated by the reservoir facilitate the discrimination of complex patterns. [8]

2.1.6 Advances and Challenges

Despite its success, reservoir computing faces several challenges and areas for improvement:

- **Optimal Reservoir Design**: Designing the reservoir, including choosing the appropriate size, connectivity, and spectral radius, remains an open research question. Methods for optimizing these parameters are actively being explored. [3]
- **Scalability**: Scaling up reservoir computing models to handle large datasets and complex tasks requires efficient algorithms and hardware implementations. Advances in parallel computing and specialized hardware, such as neuromorphic chips, are promising directions.
- **Nonlinear Dynamics**: While reservoirs are effective at capturing linear relationships, enhancing their ability to model nonlinear dynamics is an ongoing research area. Techniques such as reservoir regularization and the inclusion of nonlinear components are being investigated. [4]
- **Hybrid Models**: Combining reservoir computing with other machine learning models, such as deep learning, has shown potential for improving performance on complex tasks. Hybrid models leverage the strengths of both approaches, enabling more robust and accurate predictions.

Conclusion Reservoir computing represents a powerful and flexible framework for processing temporal data, with applications spanning various fields from robotics to finance. Its unique approach of using fixed, randomly connected networks simplifies the training process and enables efficient computation. Ongoing research aims to address current challenges and further enhance the capabilities of reservoir computing models, making them even more versatile and effective for real-world applications.

2.2 Spintronics and model of STVO

In this chapter, we will provide a simplified explanation of how Spin Torque Vortex Oscillators work. Understanding the fundamental operation of STVOs is crucial for appreciating their role in our reservoir computing network. We will break down the complex physics behind STVOs into easily digestible concepts, focusing on their unique properties and how they can be leveraged for efficient machine learning.

To understand the functioning of STVOs, it is essential to grasp two key physical effects: magnetoresistance and spin transfer torque (STT). These effects combine in STVOs to enable their unique capabilities.

2.2.1.1 Magnetoresistance

Magnetoresistance refers to the change in electrical resistance of a material due to the alignment of magnetic moments within the material. As illustrated in Fig. 1, devices utilizing magnetoresistance are made of a stack of multiple separated magnetic layers. One layer is the polarizer which ensure that the current, by passing through the polarizer, becomes polarized in spin. That is to say, that the current is mostly composed of electrons with spin aligned with the polarizer (the portion of the current which is not aligned in spin experiences scattering at a much higher rate).

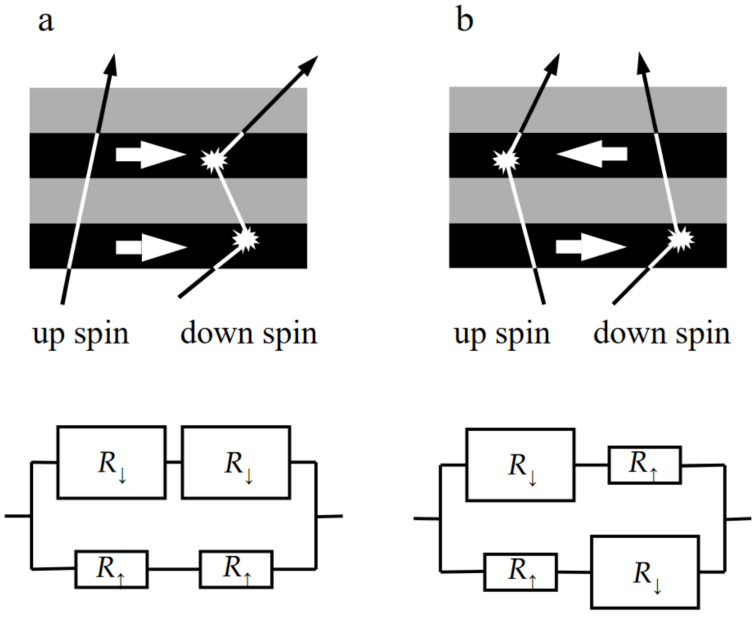


Fig. 1: Two-current model for a GMR stack. On the left, the system is in a parallel state. The spin up current is favored because its scattering probability is lower. It encounters successive small resistance states compared to the spin down channel. On the right, the system is in an antiparallel state. Spin up and spin down currents encounter successively high and low resistance layers. No spin channel is favored. Reproduced from [9].

After that, the current polarized in spin has to go through a second polarized layer. On one hand, if the magnetic moment (the polarization) of the second layer is the same as the one in the first (which we call parallel alignment) the resistance is low

(the polarized current experiences little scattering in the second layer). On the other hand, if the magnetic moment of the second layer is in the opposite direction (which we call antiparallel) the resistance is high (the polarized current experiences considerable scattering in the second layer). As we can see in Fig. 1.

Simply put, when the magnetic moments are aligned parallel, resistance is lower, and when they are antiparallel, resistance is higher. This change in resistance allows the device to convert magnetic information into electrical signals.

2.2.1.2 Spin Transfer Torque effect (STT)

Spin Transfer Torque (STT) is a phenomenon where the spin of electrons, when passed through a magnetic material, can exert a torque on the magnetic moments of that material. When a spin-polarized current is injected into a magnetic layer, it can transfer angular momentum to the magnetic moments, causing them to rotate or oscillate as illustrated in Fig. 2.

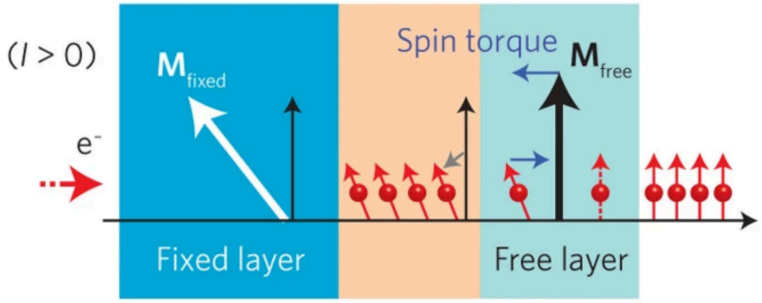


Fig. 2: Schematic illustration of a multilayered structure made of a thick ferromagnet, a nonmagnetic material and a thin ferromagnet. The fixed layer acts as a spin polarizer. When passing through the free layer, the electrons align their spin with its magnetization. This will generate a spin-transfer torque by conservation of angular momentum in the free layer. Reproduced from [10].

2.2.1.3 Combining Magnetoresistance and STT in STVOs

In an STVO, these two effects combine to create a device that can generate and detect high-frequency oscillations:

- Spin Polarization: A direct current is passed through a magnetic layer, polarizing the spins of the electrons.
- Spin Transfer Torque: This spin-polarized current enters another magnetic layer, exerting a torque on the magnetic moments and causing a magnetic vortex to oscillate.
- Oscillation Radius: The amplitude of the DC current affects the radius at which the vortex oscillates. A higher current leads to a larger oscillation radius, which in turn influences the magnetoresistance.

- Magnetoresistance Detection: As the magnetic vortex oscillates with varying radii, the resistance of the magnetic layers changes due to the magnetoresistance effect. This change in resistance creates an oscillating voltage signal.
- Output Voltage Amplitude: The amplitude of the oscillating voltage signal depends on the radius of the vortex oscillation. Larger oscillation radii, induced by higher DC currents, result in greater changes in resistance and thus higher output voltage amplitudes.

By leveraging the STT effect to induce oscillations and magnetoresistance to detect these oscillations, an STVO can efficiently convert a DC current into a high-frequency oscillating voltage of varying amplitudes. As we will see in Section 3.1.1, this study will disregard the oscillation frequency and focus on the amplitude of the resulting signal.

2.2.2 Conclusion

To conclude, a Spin Torque Vortex Oscillator is a nanoscale device that exploits the spintronic phenomenon, where the intrinsic spin of electrons and their associated magnetic moments are used to generate oscillations. STVOs are capable of converting a direct current (DC) into a high-frequency oscillating voltage of varying amplitudes. This property makes them suitable for applications in communication, sensing, and, as explored in this study, machine learning.

The STVOs will be used as nodes in the network as illustrated in Fig. 3.

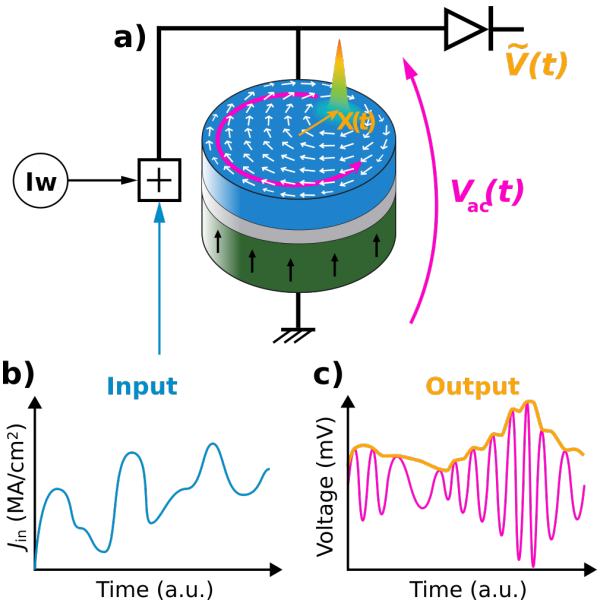


Fig. 3: Illustration of a node in the network. Adapted from [11].

Network and study introduction

In our study, we explore the integration of Spin Torque Vortex Oscillators into reservoir computing networks, aiming to leverage their unique properties for energy-efficient machine learning. This STVO-based RC network retains the fundamental structure of a conventional reservoir computing network but incorporates STVOs as the neurons within the reservoir.

3.1 The STVO-based RC network

Apart from the integration of STVOs, the overall architecture of our STVO-based RC network remains similar to that of a conventional RC network. The key difference lies in the nodes or neurons within the reservoir, which are replaced by STVOs with their specific activation functions. This replacement aims to exploit the inherent hardware advantages of STVOs, such as high-speed operation and low power consumption, for more efficient computation.

By combining the unique properties of STVOs with the established principles of reservoir computing, our study aims to evaluate the performance as well as the effect of the STVO-specific parameters.

3.1.1 Activation function

As previously discussed, an STVO is a device capable of converting a direct current into an oscillating voltage. The amplitude of this oscillating voltage is dependent on the magnitude of the DC current applied. In the context of our study, we exploit this property by using the amplitude of the oscillating voltage resulting from the reduced vortex position $s(t) = \frac{X(t)}{R}$ (the position of the vortex, $X(t)$, as seen in Fig. 3, divided by the radius of the STVO, R) as the activation function for the neurons within the reservoir, resulting in the activation function Eq. 5 seen in Fig. 4 derived from the Thiele equation approach [12].

$$s(t) = \frac{s(t-\Delta t)}{\sqrt{\left(1 + \frac{s(t-\Delta t)^{n(t)}}{\alpha(t)/\beta(t)}\right)\exp(-n(t)\alpha(t)\Delta t) - \frac{s(t-\Delta t)^{n(t)}}{\alpha(t)/\beta(t)}}} \tag{5}$$

The final position of s, i.e. when $t \to \infty$ (as illustrated in Fig. 4), depends on the input current density J:

$$S_\infty(J) = \sqrt{\frac{-\alpha(J)}{\beta(J)}} \tag{6}$$

Where $\alpha(t)$, $\beta(t)$ and $n(t)$ are dynamical parameters depending on t through $J(t)$ (the input).

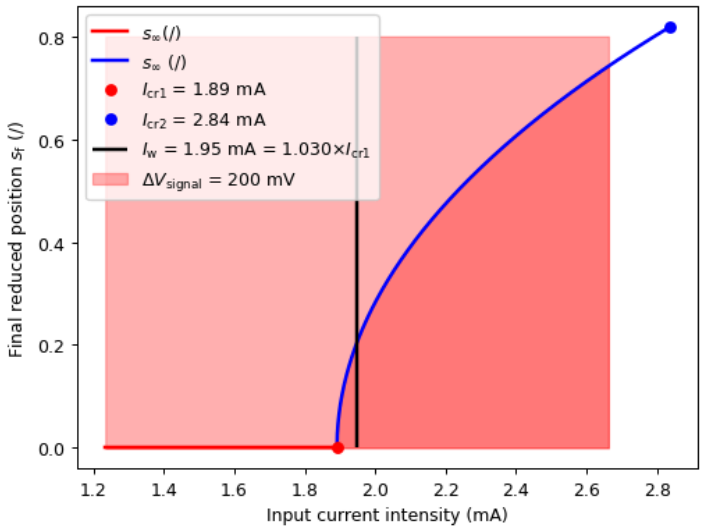


Fig. 4: Activation function of STVO representing the final ($t \to \infty$) reduced position of the vortex core for a given current which is translated into voltage.

This function represent the final (since the STVO behavior is subject to a form of "inertia" (relaxation time)) reduced position of the vortex core for a given current. The reduced position of the vortex indicates the radius from which it oscillates from the center of the device, as well as the amplitude of the resulting output voltage.

3.1.2 Network layout

As stated previously, we are using a reservoir computing network in our study. Here's a brief reminder of its basic layout as illustrated in Fig. 5:

- **Input Layer**: This layer takes the external input signals and feeds them into the reservoir. The weights connecting the input layer to the reservoir are fixed and randomly initialized. These weights are not trained, simplifying the network design and reducing computational overhead.
- **Reservoir**: The core of the RCN is the reservoir, a dynamic system composed of a numerous interconnected neurons/nodes which are illustrated in Fig. 3. The connections between these neurons are also fixed and randomly initialized. The reservoir projects the input signals into a high-dimensional space, capturing temporal patterns and relationships in the data.

- **Readout/Output Layer**: This layer produces the final output of the network. Unlike the weights in the input layer and the reservoir, the weights in the readout layer are trainable. During training, these weights are adjusted to minimize the error between predicted and actual outputs, here using ridge regression (a simple linear regression would also be an option).

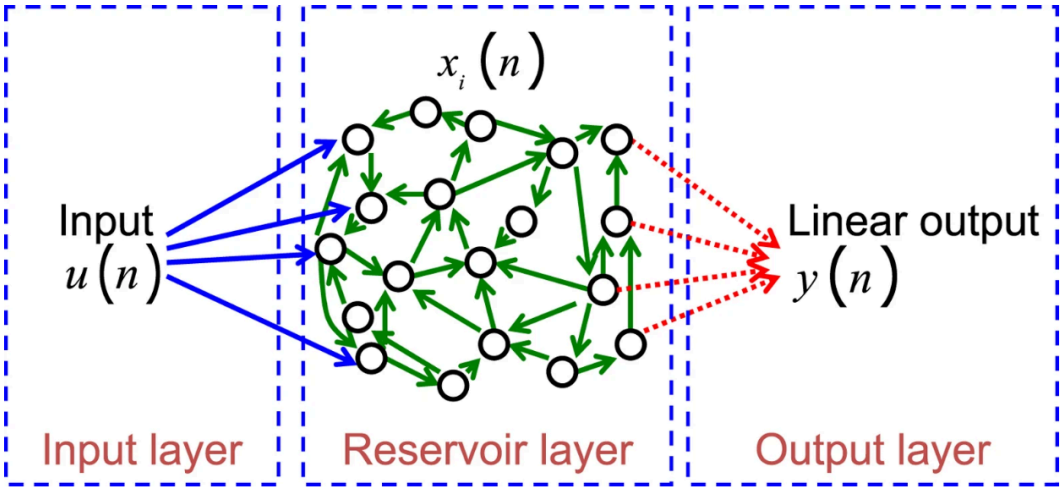


Fig. 5: Illustration of the network.

The input $u(n)$ is sent to the reservoir layer by the input layer. The reservoir layer is a nonlinear recurrent dynamical system whose internal variables are denoted $x_i(n)$. The output $y(n)$, produced in the output layer, is a linear combination of the internal variables. Only the linear output layer is adjusted during the training phase. Reproduced from [13].

3.2 Parameters

The architecture of our STVO-based RC network includes both STVO-specific parameters and traditional RC parameters. To provide a clear understanding of these parameters, we will include two tables: one listing the STVO-specific parameters and the other detailing the conventional RC parameters.

3.2.1 STVO-Specific parameters

The STVOs in our network are characterized by their unique activation functions, which differ from the typical tanh or sigmoid functions used in standard RC networks. These STVO-specific parameters play a crucial role in defining the dynamic behavior and efficiency of the network. The primary STVO parameters we considered include:

I_w (also labeled Iw in figures)	The bias current applied by default when there is no input. It shifts the center of the dynamic/of our activation function. From now on, what will be referred to as I_w will be the multiplicative factor in front of I_{cr1}[1]. So in Fig. 4 I_w would be 1.03 and not 1.95 mA, and it is represented as a black vertical line. We rapidly found that when I_w is below 1 the vortex core tends to remain stuck resulting in very poor performance[2]. Therefore, we will be working with $I_w \geq 1$ for the rest of the study.
ΔV (also labeled DeltaV in figures)	It limits the signal in a particular range around I_w. It should not be high enough to allow the vortex core to be expelled[3].
Δt (also labeled Dt in figures)	The time spent relaxing for each step (between 2 inputs). The STVOs dynamic is subject to a form of "inertia". The lower the value, the more pronounced the inertia's effect will be, hindering the dynamic. The higher the value, the more the activation function will actually look like Fig. 4 which is truly the activation function when the relaxation time between each input is infinite. It is also the parameter Δt seen in Eq. 5.

Table 1: STVO parameters

3.2.2 RC-specific parameters

Alongside the STVO parameters, our network also includes traditional RC parameters that govern the overall structure and functionality of the reservoir. These parameters include:

[1]This is the first critical current value. This is the current required for the vortex core to start oscillating, leaving the center of the STVO.

[2]As seen in Annex 10.1

[3]Which happens at I_{cr2}

N	The number of nodes in the reservoir
Spectral radius	The largest absolute value of the eigenvalues of W the reservoir's matrix.
Input connectivity	The proportion of input nodes that are connected to each reservoir node.
RC connectivity	The proportion of connections between nodes within the reservoir.
Leakage rate	Controls the update speed of the reservoir states. It defines the proportion of the previous state retained when the new state is calculated. In a way similarly to Δt, as it also adds a form of "inertia" to the system. Low leakage means increased inertia, and high leakage means decreased inertia.
Input bias	A constant value added to the input signal (it is always used (True) in this study)
Input scaling	The process of scaling the input signals before feeding them into the reservoir. (it is fixed to 1 for this study)
Regularization coefficient	The regularization coefficient used by the ridge regression. (It is fixed to 1e-7 for this study)

Table 2: RC parameters

3.2.3 Other details

The STVO-based RC network was created by building upon ReservoirPy [14]. The washout, which refers to the amount of step the network is allowed to take freely before it starts to be evaluated[4], is always of 100 steps.

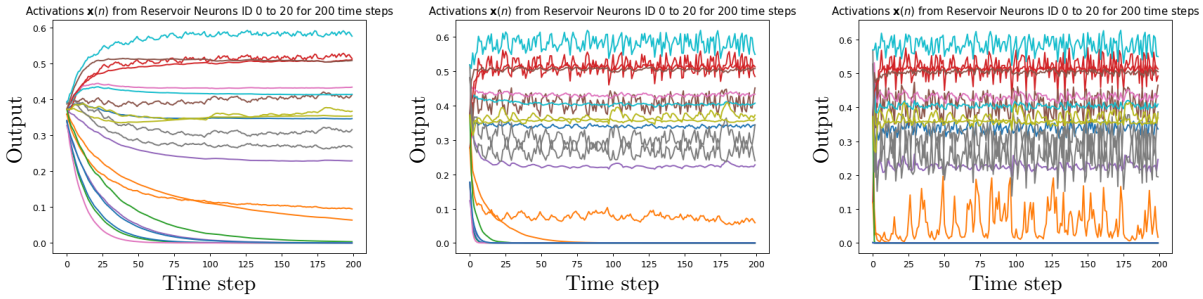
The data is always normalized between $[0, 1]$ which is then scaled once more based on ΔV.

[4]At the beginning of a time series processing task, the reservoir states are often influenced by arbitrary initial values. The washout period allows these initial influences to dissipate, ensuring that the reservoir states reflect the dynamics driven by the input signal rather than initial transients.

Qualitative study of the STVO-parameters effects

We will now quickly and qualitatively analyze the effect of each of the STVO parameters by fixing all the parameters except the one we want to study and comparing the results. By isolating each parameter, we can observe its specific impact on the behavior of the STVO-based reservoir computing network. The parameters under investigation include I_{w}, ΔV and Δt. For each parameter, we will compare the outputs of 20 neurons in the reservoir over 200 time steps. This analysis will provide insights into how each parameter influences the network's dynamics and overall performance.

4.1 Effect of Δt



a) $\Delta t = 10$ b) $\Delta t = 100$ c) $\Delta t = 1000$

Fig. 6: Illustration of the effect of Δt on the state of the reservoir for a Δt of 10, 100 and 1000

The effect of Δt is probably the most noticeable. When Δt gets lower, we can observe that the response slows down. It is expected since lowering Δt increases the inertia's effect on the system, which has a "calming effect" over it, reducing high frequency responses. As we can see in Fig. 7, its effect is similar to the leaking rate's effect, which also adds a form of inertia.

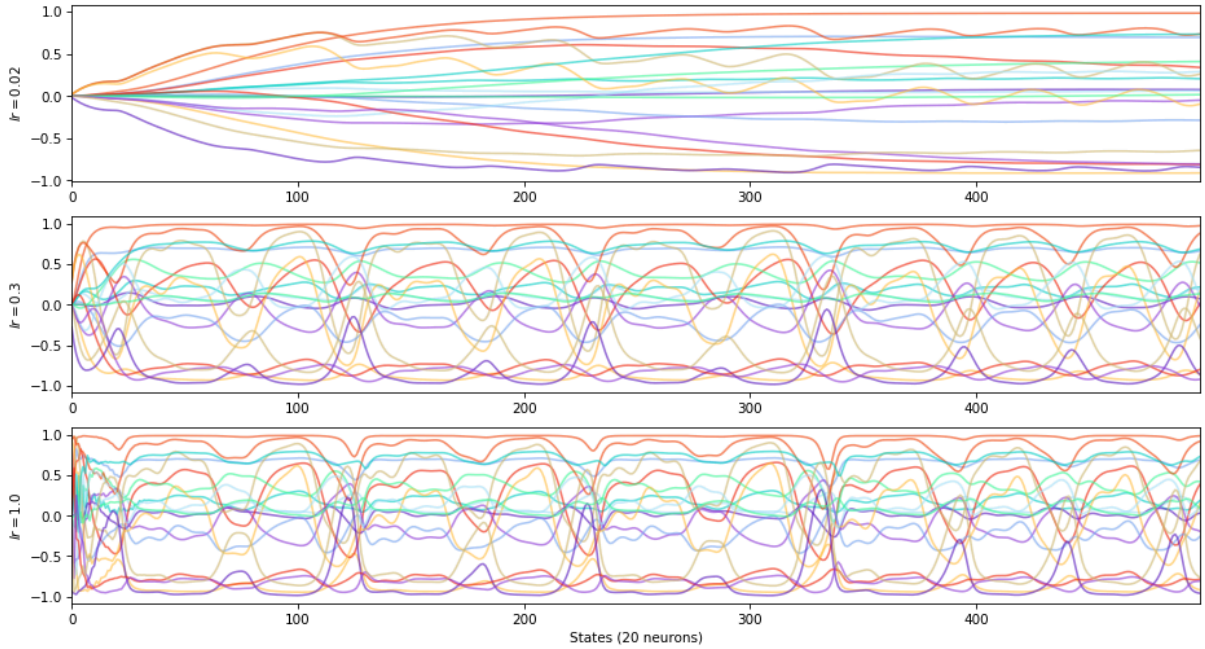
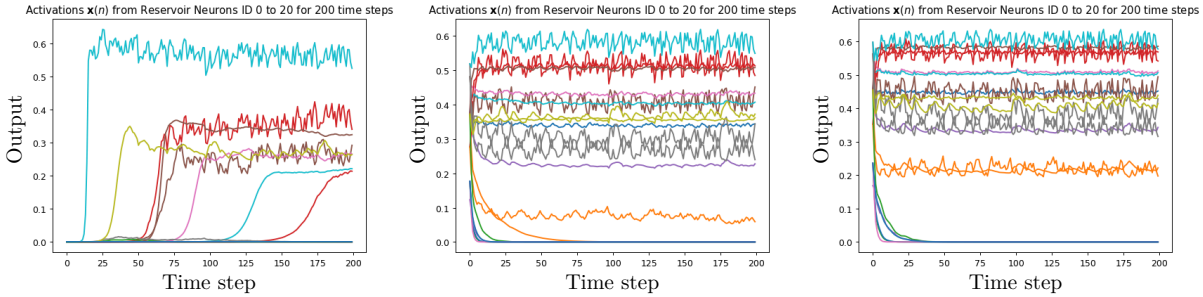


Fig. 7: Illustration of the effect of the leaking rate on the state of the reservoir. The input and the model (tanh-based) is not the same as the other figures of this chapter. Taken from [14].

4.2 Effect of I_{w}



a) $I_{\mathrm{w}} = 1.00$ b) $I_{\mathrm{w}} = 1.10$ c) $I_{\mathrm{w}} = 1.15$

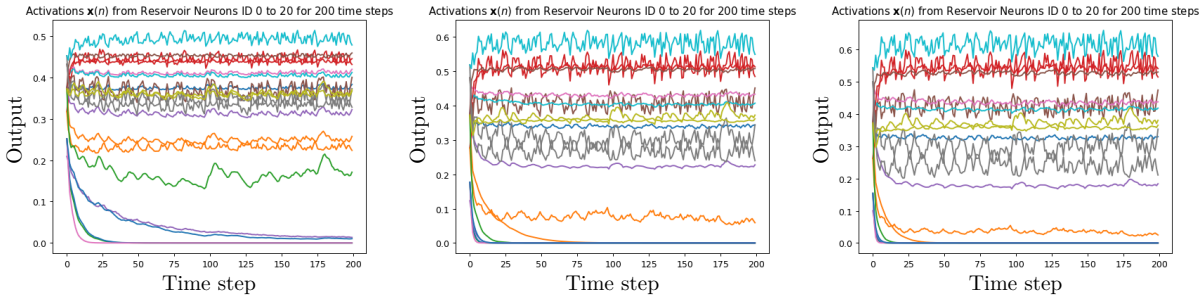
Fig. 8: Illustration of the effect of I_{w} on the state of the reservoir for a I_{w} of 1.00, 1.10 and 1.15

The second most noticeable effect is the one coming from I_{w}. I_{w} indicates the center of the dynamic as seen in Fig. 4. The first noticeable effect is at the start of the graphs seen in Fig. 8. We can see that the signals in (a) start at 0 while the ones in (b) and (c) start higher than that. It is simply because, as seen previously, when I_{w} is 1 the reduced vortex position starts at $\lim_{t\to\infty} s(t) = 0$ while, when $I_w > 1$, $\lim_{t\to\infty} s(t) > 0$.

In addition to that, we can see in (a) that the system takes more time to start. It is because when starting at $s(t) = 0$ the vortex is lying in a local minimum which means that a significant amount of energy is required in order to leave $s(t) = 0$ and enter the active region of the dynamics. This effect is even more noticeable at low Δt when the inertia's effect is stronger.

Finally, lower I_ws enable the use of larger ΔVs since we are less limited by the maximum ΔV acceptable allowing a larger coverage of the STVO's dynamics, but it isn't covered in this simple qualitative analysis.

4.3 Effect of ΔV

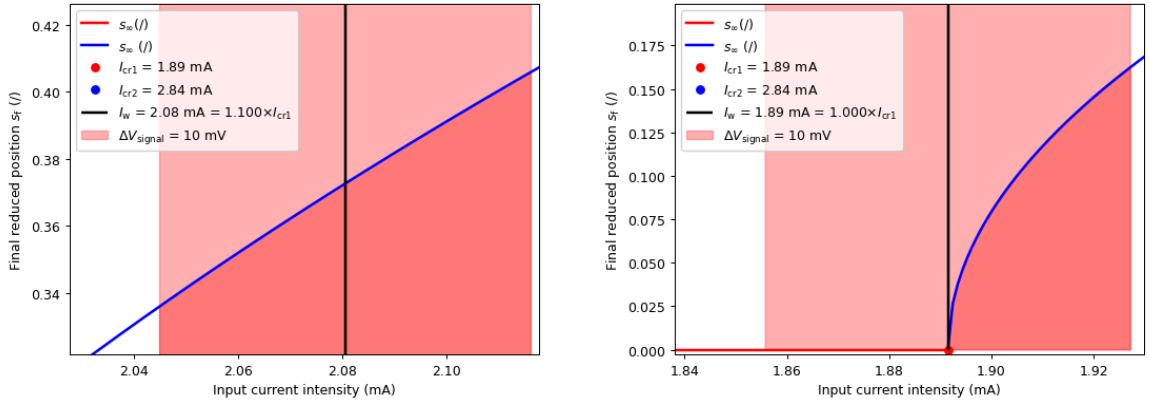


a) $\Delta V = 100$ b) $\Delta V = 180$ c) $\Delta V = 210$

Fig. 9: Illustration of the effect of ΔV on the state of the reservoir for a ΔV of 100, 180 and 210

Lastly, the least noticeable effect from a qualitative point of view results from ΔV. It is responsible for the coverage of the STVO's dynamics. Qualitatively, it reduces the amplitude of the neuron responses. We can see this effect simply in the amplitudes of the responses in Fig. 9, but this also results in some signals settling more slowly to 0 when ΔV is small.

When Δt is high and ΔV very low, the resulting activation function behave closer to a linear function (Fig. 10.a) except around $I_w = 1$ (Fig. 10.b) (where we expect it to still behave similarly to the original STVO function with a larger ΔV with this shape somewhat reminiscent of the reLU).



a) Away from $I_w = 1$ b) At $I_w = 1$

Fig. 10: Illustration of the effect of a very low ΔV

Short-term memory capacity of STVO

5.1 Introduction

We will now assess the effect of the STVO parameters on the short-term memory capacity (STMC) as defined by [15]. It is introduced to reflect the correlation between the current reservoir states (at time t) and past inputs (at time $(t-k)$) where the inputs are binary values (between 0 and 1) uniformly distributed.
The STMC is defined to be represented by the squared correlation coefficient of the target output, $y(t)$, and the predicted output, $\hat{y}(t-k)$. For each delay k tested, the output layer is fit anew using ridge regression.

The memory capacity for a certain delay k is thus given by :

$$\mathrm{MC}_k = \frac{\mathrm{cov}^2(\hat{y}(t-k), y(t))}{\sigma^2(\hat{y}(t-k))\sigma^2(y(t))} \tag{7}$$

And the total memory capacity with a maximum delay of 20 is given by :

$$\mathrm{MC} = \sum_{k=0}^{20} \mathrm{MC}_k \tag{8}$$

With this knowledge we studied the effect of three STVO parameters, I_w, ΔV and Δt by performing a grid search on these hyperparameters[5].

5.2 Observations

In Fig. 11 we can see a strong relation between MC and ΔV[6], a lesser relation with I_w and a relation with Δt which reach saturation relatively fast.

[5]The other hyperparameters are fixed, see Annex 10.2.1.

[6]Take note that here ΔV is given as a percentage of the maximum ΔV allowed by the chosen I_w. If the accumulation of I_w and ΔV exceed this maximum value the vortex gets expelled, and the oscillations do no longer occurs.

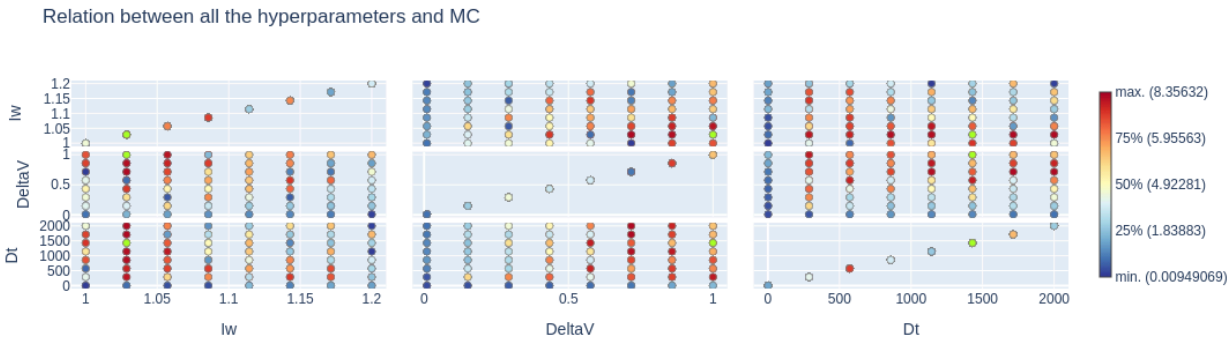


Fig. 11: Relation of hyperparameters with the target (MC) and each others

Let's now look at each of these hyperparameters one by one. In we can see a clear relation where MC increases with ΔV, in MC peak around $I_{\text{w}} = 1.03$ and in Fig. 14 we can see that MC is very low when Δt is very small then increases with Δt before saturating around $\Delta t = 860$ with no significant change after that.

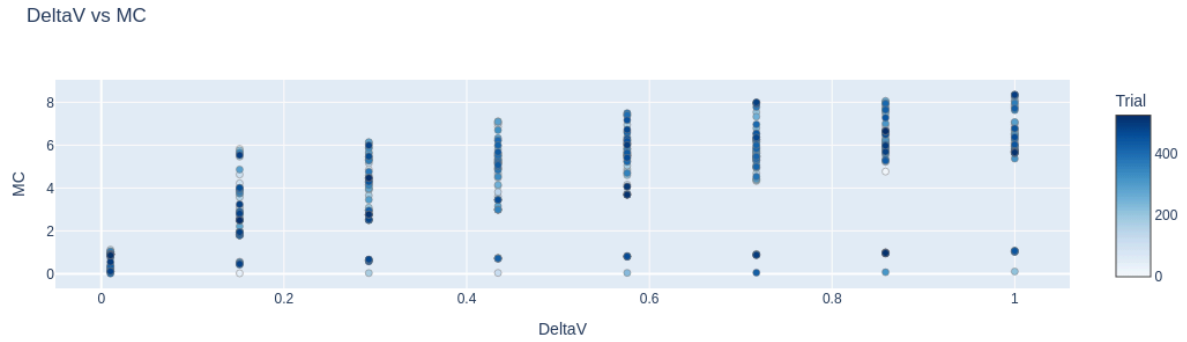


Fig. 12: Relation of ΔV(as a percentage of max ΔV) with MC

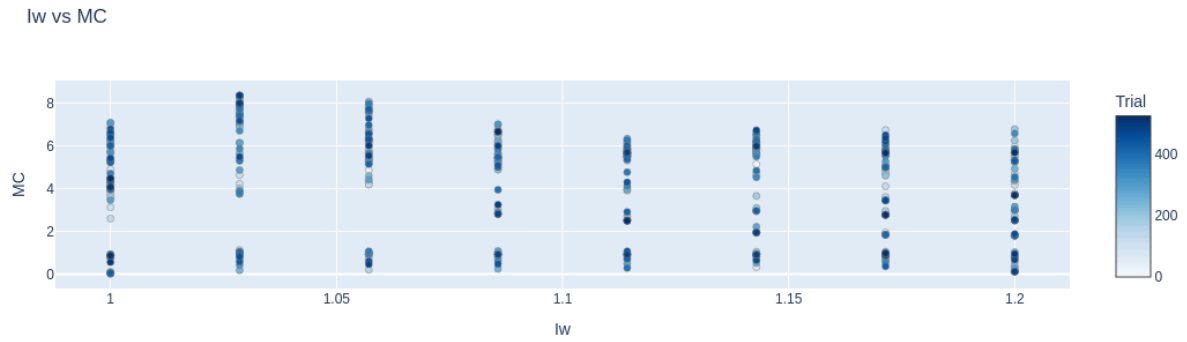


Fig. 13: Relation of I_{w} with MC

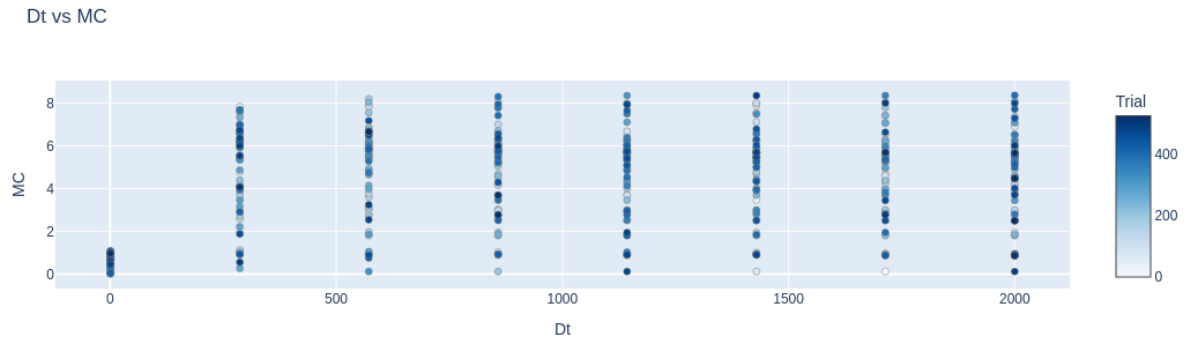


Fig. 14: Relation of Δt with MC

Diving deeper, we can see in Fig. 15 some regions forming. A larger MC seems to be favored when using a large ΔV and larger ΔVs are reachable when I_{w} is lower (which increase the maximum ΔV). Therefore, a combination of large ΔVs with small I_{w}s results in the larger MCs.

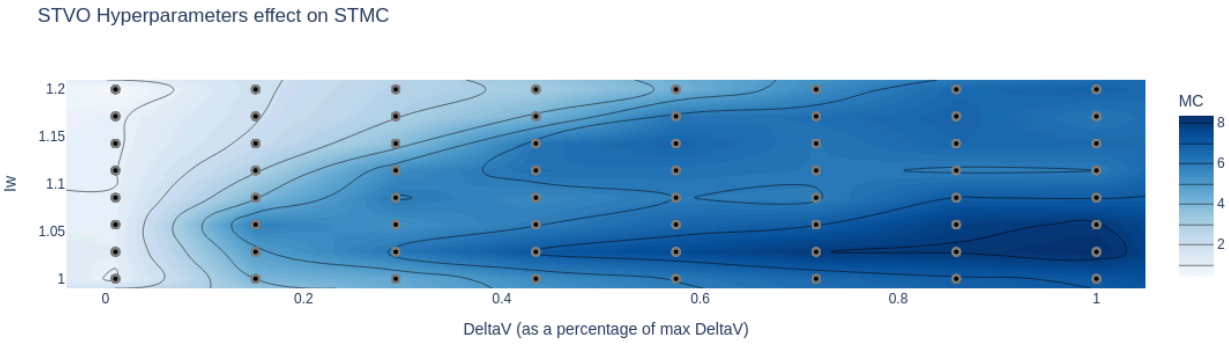


Fig. 15: Relation of I_{w} and ΔV (as a percentage of max ΔV) with MC

Finally, in Fig. 16 we can see the resulting MC_k for a given delay k where the MC is large, medium or small.

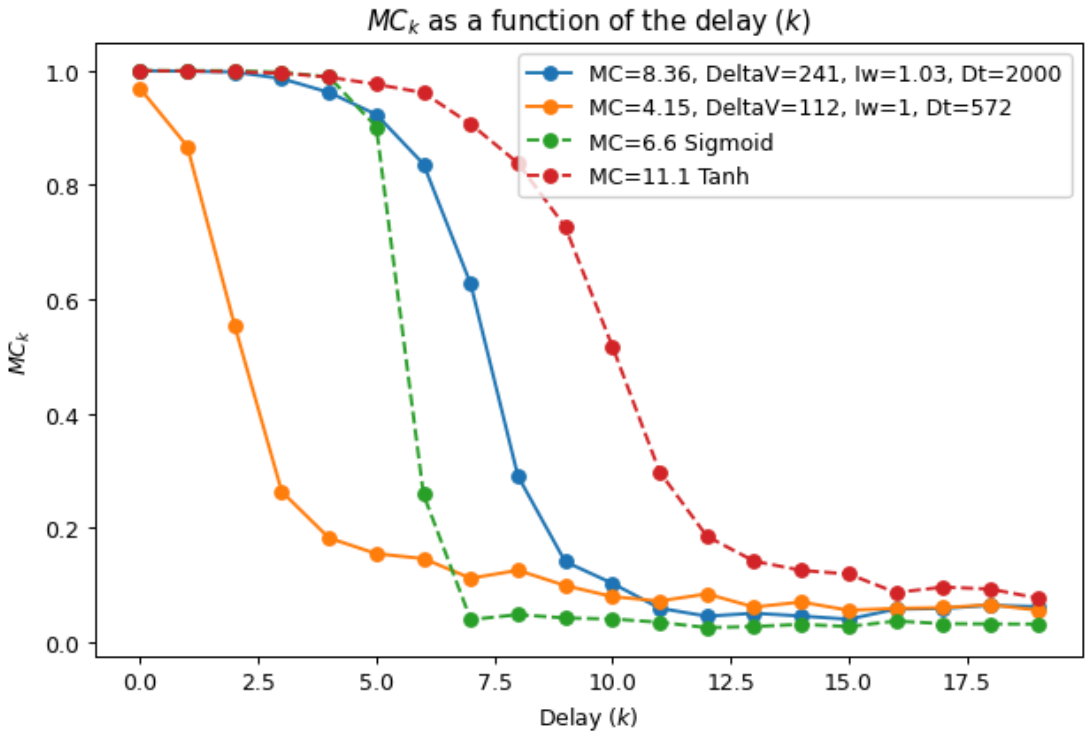


Fig. 16: MC_k for a high (blue) and low (orange) MC as well as for a sigmoid function (green) and a tanh function (red)

5.3 Interpretation

When Δt is high, we can expect almost no hysteresis effect, making the STVO activation function look like Fig. 4. In this situation, the larger ΔV results in a larger coverage of the underlying function, resulting in a richer non-linearity similar to a blend between tanh and relu when only considering $s_\infty(J)$[7]. While a lower ΔV results in something more and more linear when ΔV decreases in the active region[8] (right part in blue). Therefore, we can expect a resulting MC similar to the one obtained with functions such as tanh or sigmoid when ΔV increases.

That being said, I_w must not be too close to zero because it can easily remain stuck in the inactive region of the dynamic, especially with lower values of Δt.

Furthermore, we can see an improvement of MC with increasing Δt before reaching saturation. This saturation was to be expected, since Δt indicates the time that passed between the input at t and $t+1$. Since our system is subject to inertia, a large Δt means that the vortex core rotation radius has time to settle to its new position before the next data value is fed into the STVO. When a high enough Δt is reached, there is no significant gain to be had by increasing it, since the vortex core rotation radius is already very close to it's $t \to \infty$ value. Increasing Δt makes the activation function looks something closer to Fig. 4 by removing the inertia's effect.

[7]When the relaxation time Δt is infinite

[8]As seen in Fig. 10.

Initially we thought that the inertia's effect could potentially improve the memory capability of our reservoir, but at least in this application it doesn't seem to be the case at first glance...

5.4 Observations deeper into the effect of low Δt

When observing the effect of low Δt in more details[9], we can see in the following figures Fig. 17, Fig. 18 and Fig. 19 that a lower Δt can indeed play a positive role in the preservation of information. This potentially positive effect is more noticeable when the size of the reservoir is relatively small, like in Fig. 17 with 50 nodes. In this figure, we can see MC increasing with decreasing Δt. By giving a closer look to the graph, we can see that lowering Δt seems to exchange shorter-term memory capacity for longer-term memory capacity, specially when comparing the green curve (Δt=150) to the blue one (Δt=2000). This effect is also noticeable in Fig. 18 where $N = 100$, but the total MC now gets lower with decreasing Δt. And finally in Fig. 19 with $N = 300$ only the loss in shorter-term memory remains.

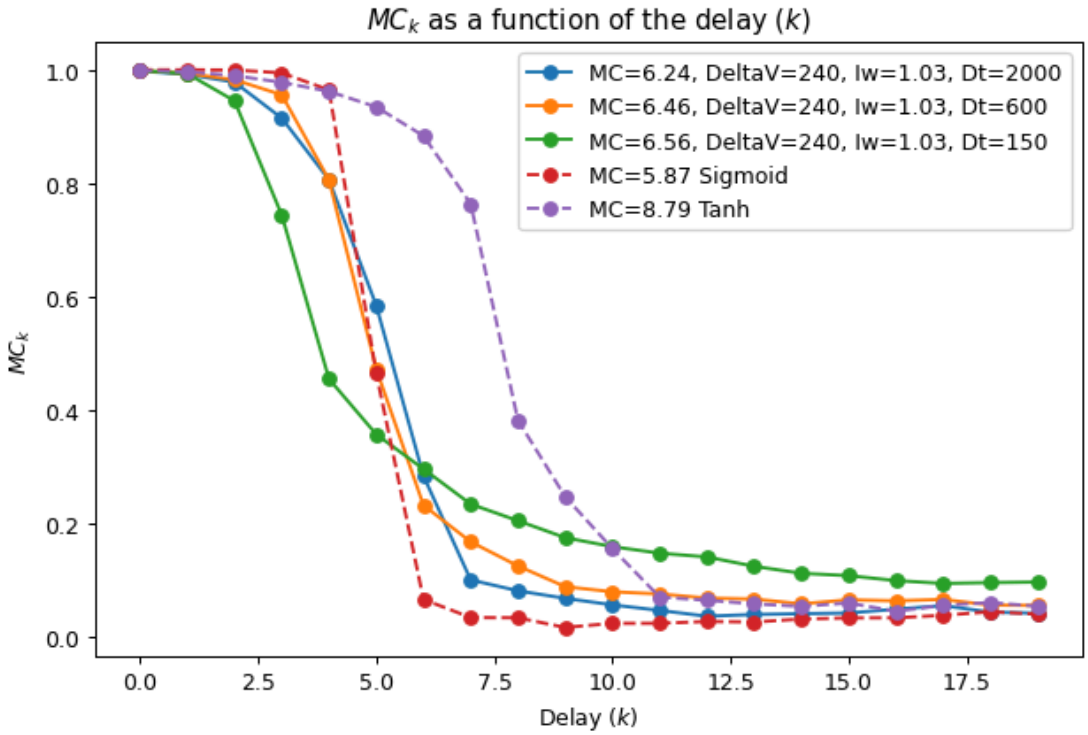


Fig. 17: MC_k for different values of Δt with $N = 50$ (number of nodes)

[9]We limited the grid-search to $\Delta t \in [1,300]$

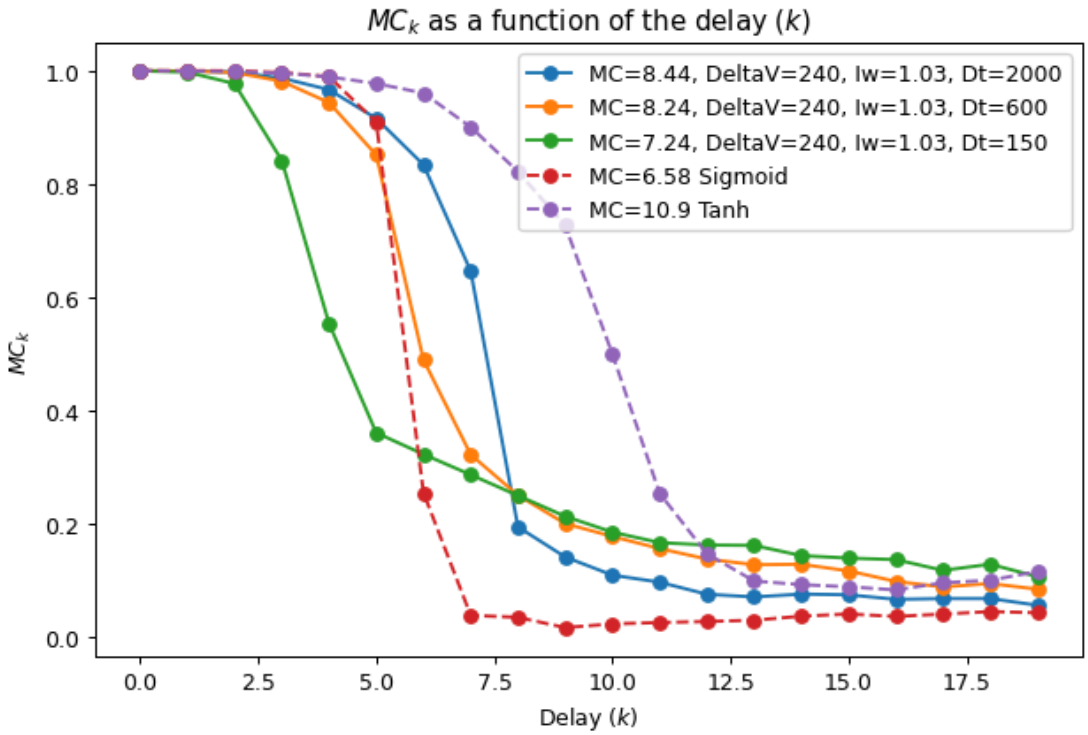


Fig. 18: MC_k for different values of Δt with $N = 100$ (number of nodes)

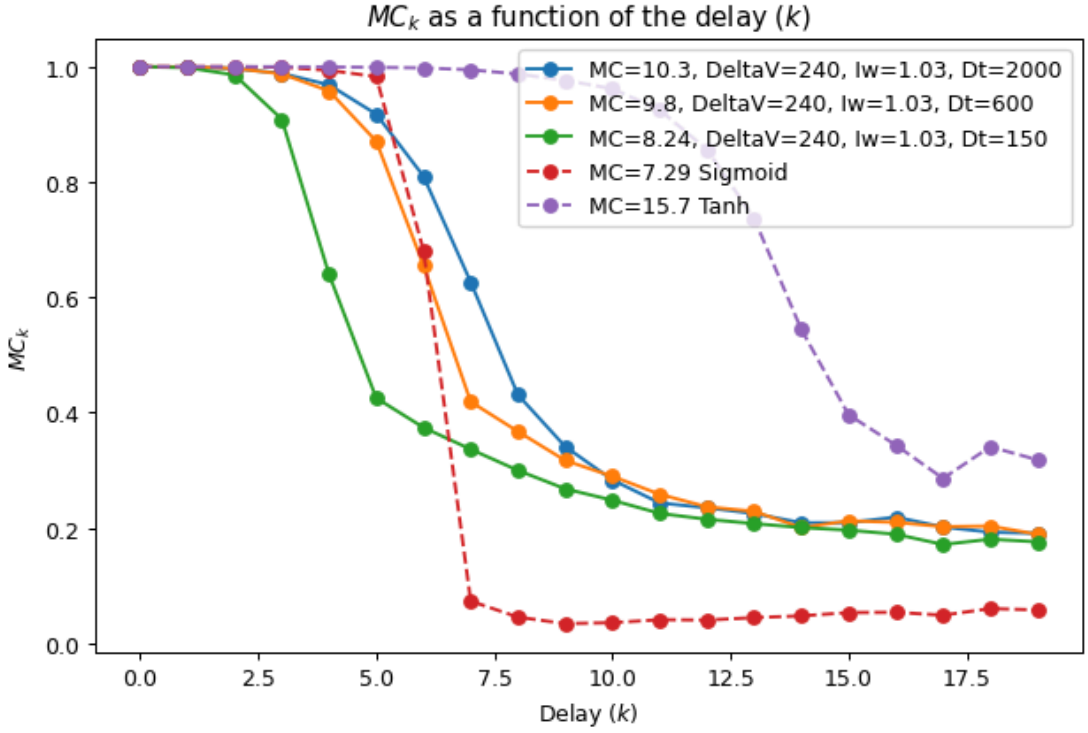


Fig. 19: MC_k for different values of Δt with $N = 300$ (number of nodes)

We can also compare Fig. 20 with Fig. 15 and see a similar picture except for a reduced MC when I_w is close to 1.

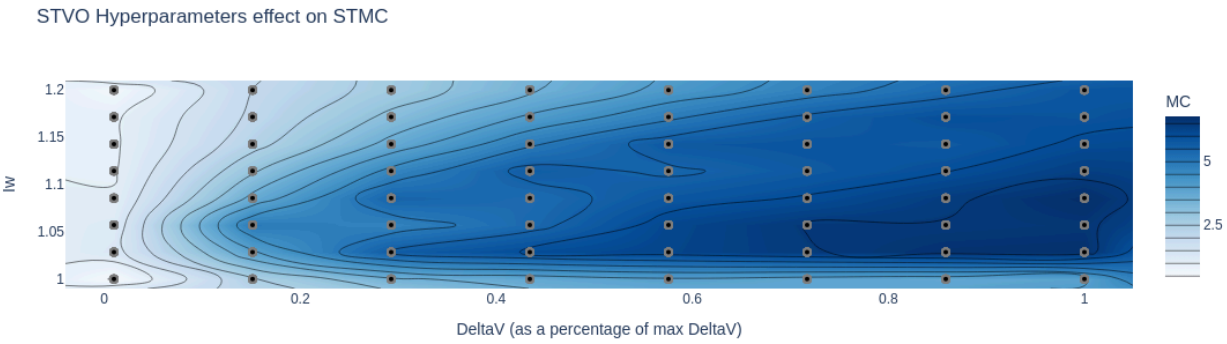


Fig. 20: Relation of I_w and ΔV (as a percentage of max ΔV) with MC (With smaller Δts)

We can also observe in Fig. 21 that for a given I_w close to 1 MC increases as Δt increases.

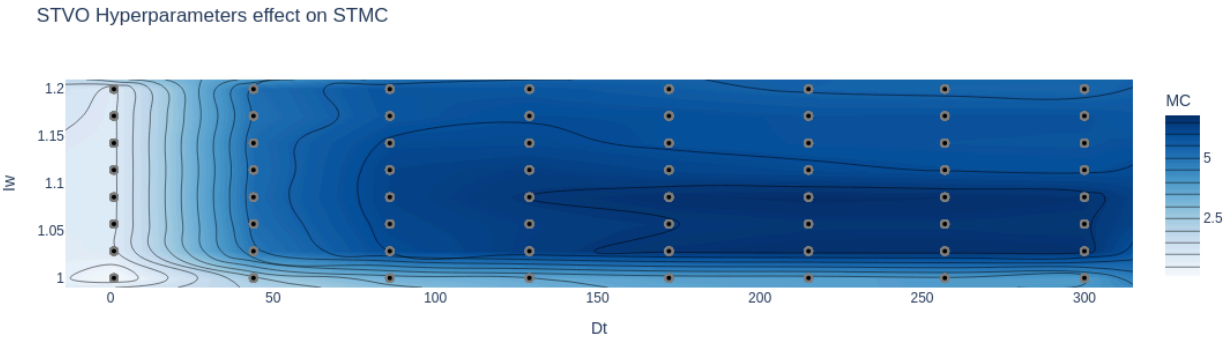


Fig. 21: Relation of I_w and Δt with MC

5.5 Interpretation of the effect of low Δt

In the end, as we thought, Δt seems to have an impact improving on the STMC, but only when the size of the reservoir is relatively low. Under these circumstances, there is a gain in longer-term memory at the cost of shorter-term memory. It is because information can be retrieved thanks to the inertia acting as a memory. The inertia's effect blends old inputs with more recent inputs, explaining the observed effect where there is a simultaneous gain in longer-term memory and loss in shorter-term memory.

In Fig. 20 (and in Fig. 21), the loss in MC at low I_w when Δt is low was expected (as compared to Fig. 15 and the results with higher Δts). Indeed, when Δt is low, the inertia's effect is stronger which can be an issue when the radius of the vortex core reaches 0 (in the inactive zone, in red in Fig. 4) because it can stay stuck there maintaining the output to or very close to 0 decreasing MC and generally the performance of any RC network[10].

[10]As seen in Annex 10.1

5.6 Conclusion

With a Δt sufficiently large, larger ΔVs (with lower I_{w}s) should be considered for tasks where long-term information is important, whereas lower ΔVs (with larger I_{w}s) could possibly be considered if the task does not require long-term memory.

In the case of smaller reservoirs, the previous statement still applies, but one must be more careful about values of I_{w} close to 1 as they could result in a dynamic stuck at 0 lowering MC and the general performance of any RC network.

Furthermore, in these smaller reservoirs, lowering Δt can increase the STMC of the reservoir, exchanging shorter-term memory capacity for longer-term memory capacity outperforming the longer-term memory of the tanh-based model. With larger reservoirs, this property is lost and MC can only decrease (longer-term memory is no longer improving) with decreasing Δt.

We believe that when the reservoir size increases, the longer-term information can be more reliably (with less transformation through the inertia's effect) stored amongst the larger amount of nodes, which would explain the observed behavior.

NARMA2 benchmark

6.1 Introduction

The NARMA2 (Nonlinear Auto Regressive Moving Average) benchmark is a widely used test in the field of time series prediction and reservoir computing. It serves as a challenging benchmark due to its nonlinear and dynamic nature. The NARMA2 model is designed to assess the memory capacity and predictive power of computational models, particularly those used in recurrent neural networks and reservoir computing. It also tests the "universal function approximation" property of our RC network.

NARMA2 generates a time series according to the following nonlinear equation:

$$y(t+1) = 0.4y(t) + 0.4y(t)y(t-1) + 0.6u(t) \tag{9}$$

where :

- $y(t)$ is the output at time t
- $u(t)$ is a random input at time t. It typically comes from a uniform distribution $u(t) \in [0, 0.5]$

In Fig. 22 you can see an illustration of the resulting time series.

The measure used to assess our model performance will be a normalized mean square error in which we normalize the mean square error by using the variance such that :

$$\text{NMSE} = \frac{1}{N} \frac{\sum_{t=1}^{N} (\hat{y}(t) - y(t))}{\sigma^2(y(t))} \tag{10}$$

where :

- $y(t)$ is the NARMA2 output at time t
- $\hat{y}(t)$ is the model's prediction for time t
- N is the number of time steps

6.2 Benchmark

We performed a random search to optimize the parameters of our STVO-based RC network[11]. Through this process, we observed several key trends:

- STVO Parameters:
  - I_{w}: Higher values of I_{w} yielded better results, up to the maximum value we set for the search.

- ΔV: Optimal performance was observed when ΔV was around 70% of the maximum ΔV.
- Δt: Very low Δt values (below 300) decreased accuracy. Increasing Δt improved accuracy up to a certain point, beyond which it no longer had a strong relationship with performance. This trend of diminishing returns with higher Δt values became a recurring theme.

- Reservoir Computing Parameters:
  - **Spectral Radius**: A lower spectral radius around 0.5 was preferred.
  - **Leaking Rate**: The optimal leaking rate was found to be around 1.3.
  - **Input and RC Connectivity**: No strong relationship was found between these parameters and accuracy.

It is important to note that while the spectral radius and leaking rate have a significant impact on the performance of tanh and sigmoid-based RC networks, this effect is much less pronounced in the STVO-based RC network[11]. This suggests that the intrinsic properties of STVOs may offer robustness to variations in these parameters, potentially simplifying the tuning process for hardware implementations.

In Table 3 are the optimal hyperparameters found through the random search.

Number of nodes	I_w	ΔV	Δt	Spectral radius	RC connectivity	Input connectivity	Leaking rate
300	1.173	93 (0.574 maxDV)	1939	0.895	0.154	0.452	0.816

Table 3: Optimal hyperparameters used for STVO reservoir

After 10 runs with the same parameters but different seeds, the results are :

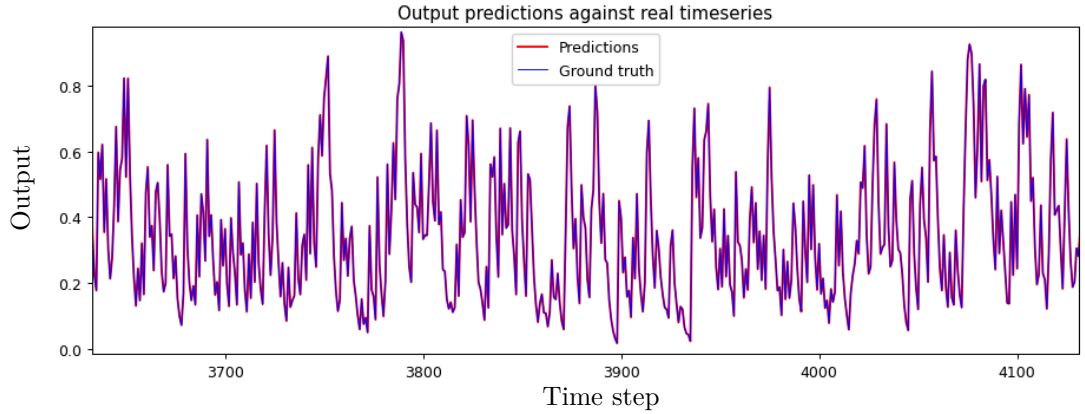


Fig. 22: Results of our model on the test set when $y(t)$ is fed from the ground truth

[11]More details on the benchmarking process and hyperparameter tuning in Annex 10.3.1

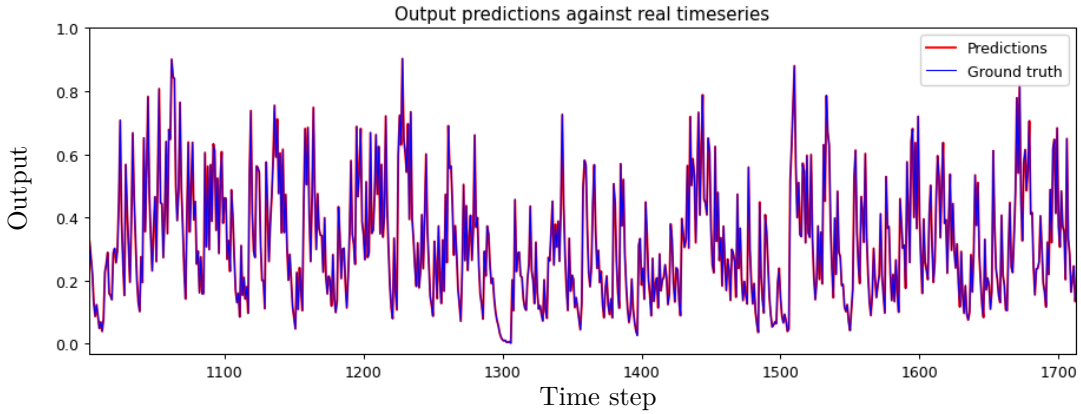


Fig. 23: Results of our model on the validation set when $\hat{y}(t)$ is fed from the model's previous prediction

	NMSE mean	NMSE variance
Using $y(t)$[12]	1.360e-04	8.982e-9
Using $\hat{y}(t)$[13]	1.840e-04	1.652e-8

Table 4: Results of STVO reservoir.

	NMSE mean	NMSE variance
Using $y(t)$	1.237e-07	3.562e-16
Using $\hat{y}(t)$	1.905e-07	1.134e-15

Table 5: Results of tanh reservoir[14].

6.3 Discussion and conclusion

One thing to note first is that, as we have seen in Section 5, high memory capacity (MC) was linked to low I_w and large ΔV. Here the optimal performance was reached at a high I_w and the ΔV wasn't that high. This would suggest that high MC is not required to perform the NARMA2 task. Furthermore, the tanh based network favored a very low spectral radius[14] (0.2) which is linked to a decrease in STMC[15]. That is to say, both STVO based and tanh based RC network were favoring low STMC networks when performing the NARMA2 task.

In order to perform the NARMA2 task the two coefficients MC_0 and MC_1 play the most important roles as the NARMA2 task only involves the current input $u(t)$ and $y(t)$ as well as the previous input $y(t-1)$ where long memory abilities is not tested (this is different for higher order NARMA tasks such as NARMA10). Therefore, we believe that reducing the memory capacity could be a way to focus on recent inputs which are more relevant, disregarding long term memory in the process.

In [16], they state that large STMC results in better NARMA2 performance but as we have seen here it doesn't seem to be the case except when the STMC is so low that MC_1 or even MC_0 gets affected.

[12] from Fig. 22
[13] from Fig. 23
[14] Details in Annex 10.3.3
[15] Details in Section 10.2.2

6.3.1 Comparison with state of the art

It is challenging to make an apple-to-apple comparison between different reservoir computing (RC) models due to variations in the number of nodes and network architecture across implementations. Additionally, comparing fully physically implemented RC networks with their fully simulated counterparts introduces further complexity. Fully physically implemented RC networks are expected to have reduced performance compared to their simulated counterparts due to physical limitations, such as noise, precision of the components, and variations in manufacturing. These factors can lead to discrepancies in accuracy and generalizability. Consequently, while simulated networks can be fine-tuned with precise parameters and perfect consistency, physical implementations must contend with real-world imperfections, which can impact overall performance.

Furthermore, many studies do not disclose their results on a validation set, reporting only on their test set in closed loop. In our case, we will use the metrics obtained in closed loop on our validation set.

	Type	Number of Nodes	NMSE
Our STVO-based model	Simulation	300	1.84e-04
Diode-based model[16]	Simulation	300	1.6e-02[16]
Deep-IGR (Ion-gating reservoirs) [17]	Physical	Deep 200+400	9.08e-03
EDL-IGR[18]	Simulation (COMSOL)	8×10	2e-02
Redox-IGR[19]	Physical	40	1.63e-01
Spin-Waves[20]	Physical	100	1.81e-02
Electrochemical[21]	Physical	112	6.24e-02
CNT-Net[22]	Physical	60	2.3e-02
Our tanh based model	Pure mathematics	300	1.905e-07

Table 6: Performance comparison with state of the art

Finally, for what is comparable, we can see in that our model performs very well against the others, but remember that this study is purely model based and didn't consider the effect of introducing noise into the network. That being said, this demonstrates the ability of our STVO-based model at performing the NARMA2 benchmark.

[16]Additionally significant difference between test set (1.6e-2) and validation (7.7e-2)

Mackey–Glass time series

7.1 Introduction

The Mackey-Glass benchmark is a popular test in the field of time series prediction and reservoir computing, known for its chaotic nature and complex temporal dependencies. Originally developed to model physiological systems, the Mackey-Glass equation generates a time series that poses significant challenges for predictive models, making it an excellent benchmark for evaluating their performance.

The Mackey-Glass equation is a time-delay differential equation defined as:

$$\frac{dy(t)}{dt} = \beta \frac{y(t-\tau)}{1 + y(t-\tau)^n} - \gamma y(t) \tag{11}$$

where:

- $y(t)$ is the output at time
- τ is the delay parameter, which introduces complexity and temporal dependencies.
- β, γ and n are constants that control the behavior of the system.

We choose a common set of parameters as such :

- $\beta = 0.2$
- $\gamma = 0.1$
- $n = 10$
- $\tau = 17$

7.2 Benchmark

Similar to NARMA2, we performed a random search to optimize the parameters of our STVO-based RC network[17]. Through this process, we observed several key trends:

- STVO Parameters:
  - I_{w}: No strong relationship was found between I_{w} and accuracy except at low I_{w} (below 1.05) where the results gets more spread out.
  - ΔV: Higher values of ΔV yielded better results (close to the maximum ΔV).

[17]Details in Annex 10.4.1

- Δt: Again, very low Δt values (below 300) decreased accuracy. Increasing Δt improved accuracy up to a certain point, beyond which it no longer had a strong relationship with performance.

- Reservoir Computing Parameters:
  - **Spectral Radius**: A lower spectral radius (below 1) was linked to a decrease in performance.
  - **Leaking Rate**: The optimal leaking rate was found to be around 1.3.
  - **Input and RC Connectivity**: No strong relationship was found between these parameters and accuracy.

Like for NARMA2, when it came to the spectral radius and leaking rate influence on performance, a stark contrast was observed comparing our STVO based network with tanh and sigmoid-based RC networks[17] where the STVO based network was less sensitive to changes in these parameters.

In Table 7 are the optimal hyperparameters found through the random search.

Number of nodes	I_w	ΔV	Δt	Spectral radius	RC connectivity	Input connectivity	Leaking rate
300	1.195	146 (0.967 maxDV)	254	1.976	0.487	0.668	1.368

Table 7: Optimal hyperparameters used for STVO reservoir

While for the same set of hyperparameters the chosen seed had little effect on the ability of the model to perform "imitation[18]" it had a noticeable effect on the ability of the model to stay stable when using its own output as input in closed loop. This behavior was also observed in typical RC networks with tanh activation functions, though they tend to remain in sync for longer[19].

After 10 runs with the same parameters but different seeds, the results are :

NMSE mean	NMSE variance	Stability when the output of the model is fed as input[20]
2.525e-06	3.168e-13	10%

Table 8: Results of STVO reservoir when $y(t)$ is fed from the ground truth as well as the percentage of free run that remained stable when $\hat{y}(t)$ is fed from the model's previous prediction.

[18]As in performing well when the input $y(t)$ comes from the ground truth and not from the model previous prediction $\hat{y}(t)$ in closed loop.

[19]Details in Annex 10.4.2.

[20]which means that the output remained stable during 8000 steps

NMSE mean	NMSE variance	Stability when the output of the model is fed as input[20]
8.178e-08	3.98e-16	30%

Table 9: Results of typical RC network with tanh activation function[19] when $y(t)$ is fed from the ground truth as well as the percentage of free run that remained stable when $\hat{y}(t)$ is fed from the model's previous prediction.

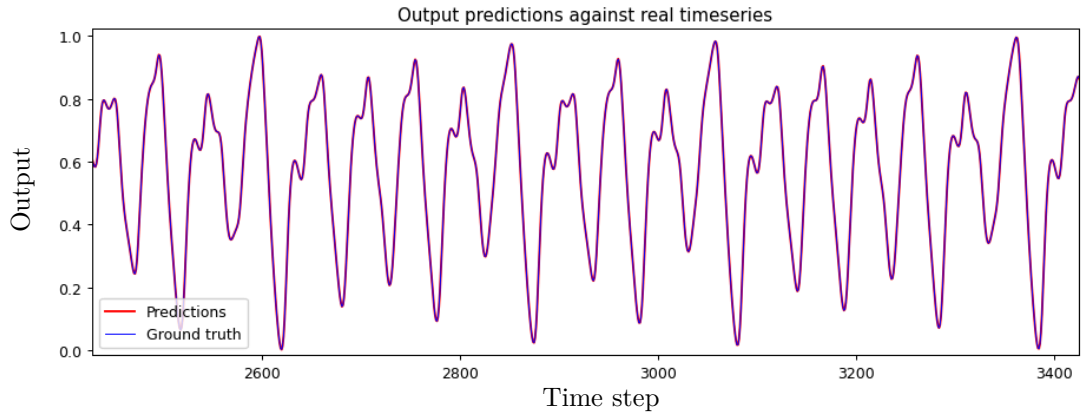


Fig. 24: A sample result of our STVO model on the test set when $y(t)$ is fed from the ground truth

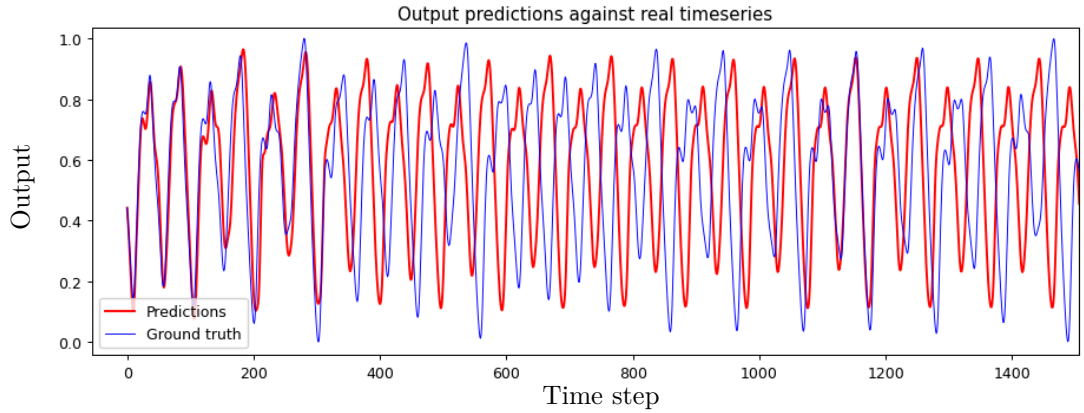


Fig. 25: A sample result of our STVO model on the validation set when $\hat{y}(t)$ is fed from the model's previous prediction

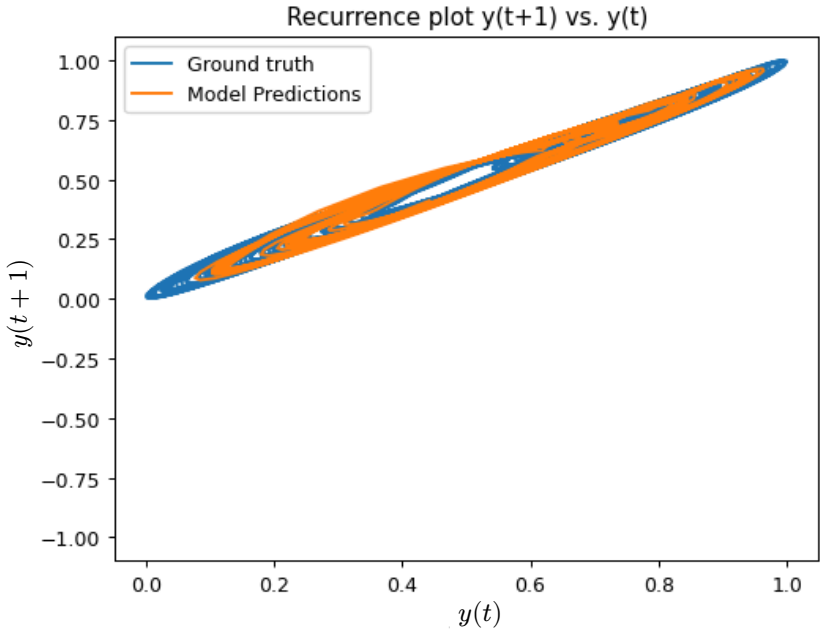


Fig. 26: The recurrence plot of the result obtained in Fig. 25 comparing the prediction with the ground truth.

7.3 Discussion and conclusion

These results are not as good as the ones obtained using a regular tanh activation function with a reservoir of the same size[19]. That being said, the STVO outperformed the sigmoid when it came to prediction in closed loop. Nevertheless, these results remain satisfactory. Even when the model output goes out of synchronization with the ground truth, we can see in Fig. 26 that the general dynamic of the system is preserved, even resynchronizing at moments.

Following a random search, our STVO-based model showed a preference for larger ΔV values, up to a point of saturation. Interestingly, the model was somewhat insensitive to variations of I_{w}.

We know that, in the tanh-based model, a spectral radius of 1.5, which enables more chaotic behavior, was found to be optimal. Although the exact relationship between ΔV and chaotic behavior in the STVO-based network is not fully understood, we hypothesize that ΔV may influence the dynamics in a manner similar to how the spectral radius affects the tanh-based model. This is supported by the observation that the spectral radius had a very limited effect on the performance of the STVO-based network, suggesting that another parameter (potentially ΔV) might be responsible for driving the network's dynamics, comparable to the role of the spectral radius in a tanh-based model.

Exploiting a larger portion of the dynamic through a larger ΔV could result in a more complex input output dynamics (remember that our activation function had a more complex behavior than a simple tanh or sigmoid), which in some circumstances could be mimicking the more complex dynamics resulting from a larger interconnection between

neurons in the tanh-based network. That being said, the relation is definitely not one to one, as the means by which we achieve this dynamics are quite different. In addition to that, large ΔVs were linked to an increase in MC where large spectral radii (above 1) were linked to very variable MCs[21]. We believe that further studies would be required to experimentally test this hypothesis.

In addition to that, further studies could also be conducted in order to get more insight into the effect of the STVO-parameters on closed loop stability.

[21] As seen in Annex 10.2.2.

Real world application on sunspot number prediction

8.1 Introduction

The Sunspot Number dataset [23] is a time series that records the number of sunspots observed on the surface of the sun over a given period. Sunspots are temporary phenomena on the Sun's photosphere that appear as spots darker than the surrounding areas. The dataset is crucial in studying solar activity, which has significant implications for space weather, climate studies, and understanding solar cycles.

8.2 Results

Again, we performed a random search to optimize the parameters of our STVO-based RC network[22]. Through this process, we observed several key trends:

- STVO Parameters:
  - I_w: While the best results were obtained at low values of I_w the results were more spread out. The opposite is true for high I_w models, which performed very similarly to each other but without reaching the performance of the best low I_w models.
  - ΔV: Similarly, the variability of the results increases when ΔV increases. Very low ΔVs were linked to slightly reduced performance before quickly saturating.
  - Δt: Here we have a different result than in the previous benchmarks. The best performing models were observed at relatively low Δt (below 250). That being said, when Δt was too low (reaching single digit) the results worsened suddenly.
- Reservoir Computing Parameters:
  - **Spectral Radius**: A lower spectral radius around 0.5 yields better results, which are less spread out as well.
  - **Leaking Rate**: Very slight decrease in performance when the leaking rate increases.

[22]Details in Annex 10.5.1.

- **Input and RC Connectivity**: No strong relationship was found between these parameters and accuracy.

This time again, tanh and sigmoid-based RC networks are significantly more sensitive to the spectral radius as well as the leaking rate than the STVO-based RC networks[22].

In Table 10 are the optimal hyperparameters found through the random search.

Number of nodes	I_w	ΔV	Δt	Spectral radius	RC connectivity	Input connectivity	Leaking rate
300	1.039	56 (0.239 maxDV)	245	1.092	0.127	0.22	0.552

Table 10: Optimal hyperparameters used for STVO reservoir

After 10 runs with the same parameters but different seeds, the results are[23] :

NMSE mean	NMSE variance
0.117	7.482e-06

Table 11: Results of STVO reservoir when $y(t)$ is fed from the ground truth as well as the percentage of free run that remained stable when $\hat{y}(t)$ is fed from the model's previous prediction.

NMSE mean	NMSE variance
0.118	6.967e-07

Table 12: Results of tanh reservoir when $y(t)$ is fed from the ground truth as well as the percentage of free run that remained stable when $\hat{y}(t)$ is fed from the model's previous prediction.

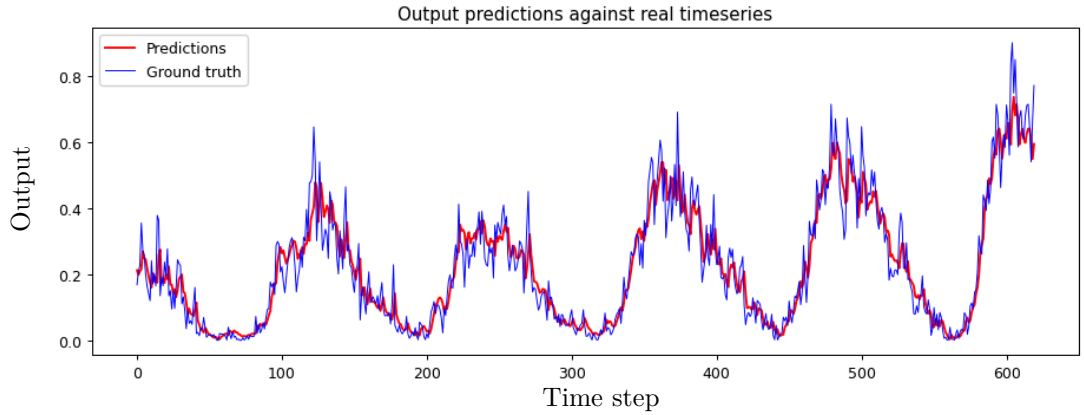


Fig. 27: A sample result of our model on the test set when $y(t)$ is fed from the ground truth

[23]All the tested models were stable.

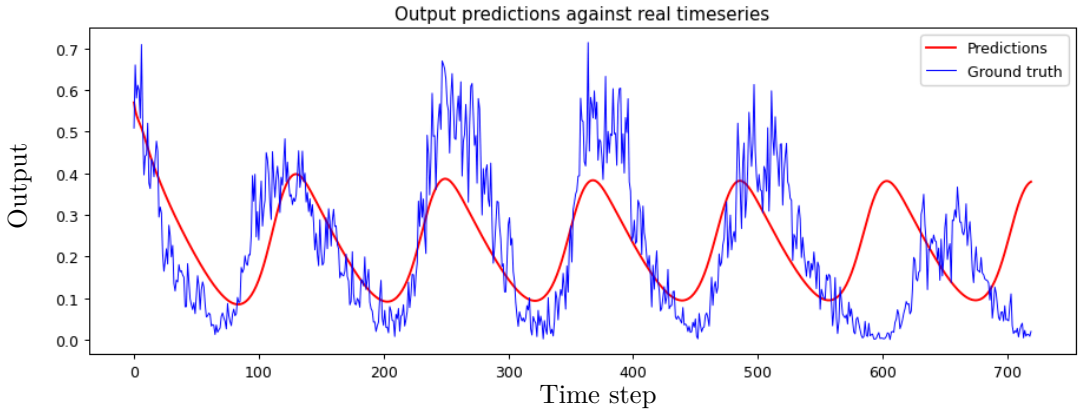


Fig. 28: A sample result of our model on the validation set when $\hat{y}(t)$ is fed from the model's previous prediction

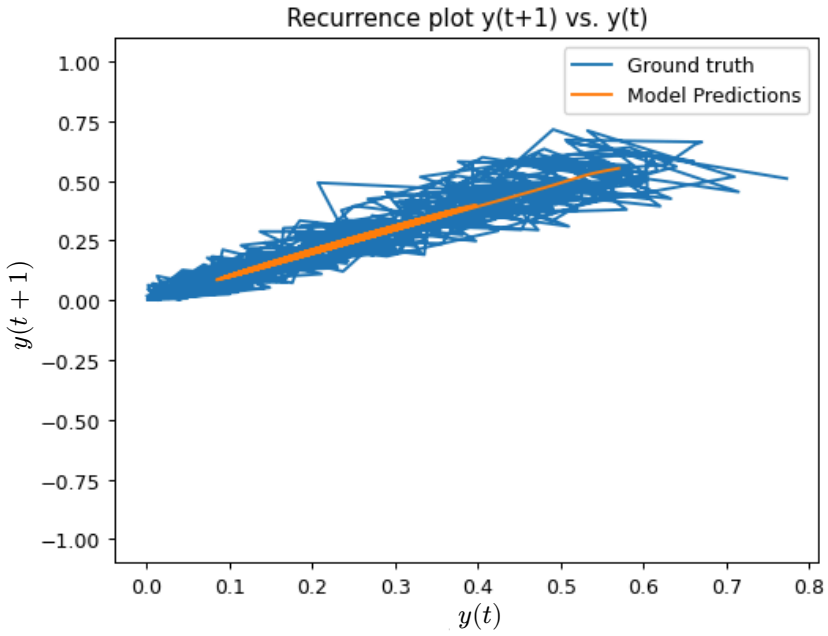


Fig. 29: The recurrence plot of the result obtained in Fig. 28 comparing the prediction with the ground truth.

8.3 Discussion and conclusion

In contrast with previous benchmarks, when it came to the sunspot number prediction task, the STVO-based network performed similarly to the tanh-based network. We hypothesize that this could be due to the nature of the noise in the sunspot prediction task. In the previous benchmarks, the noise was either an explicit input, as with $u(t)$ in the NARMA2 task, or effectively non-existent, as with the Mackey-Glass dataset. Under these conditions, the ability to fit the training set with high accuracy was rewarded, which favored the tanh-based network. However, in the sunspot prediction task, the tanh-based network's ability to closely fit the training set led to overfitting on the noisy training data. This overfitting resulted in poorer generalization to the test set, reducing the overall relative performance of the tanh-based model for this task.

Another observation contrasting with previous benchmarks was the improved performance of the STVO-based model at low Δt, even at very low Δt[24], which until now was associated with a decrease in performance. We believe that this is also related to the noise in the dataset. Indeed, low Δt increases the effect of inertia on the dynamics of the system, which allows the system to average out the subsequent inputs. This reduced the model's ability to fit the noise, leading to a more robust and generalized model. This effect enabled the STVO-based model to better handle the inherent noise in the sunspot dataset.

This time, low I_{w} were favored while ΔV had only a limited effect[24], meaning that a larger MC is not necessary and this isn't the reason why I_{w} is low. We hypothesize that these results could be explained by the fact that low I_{w} networks tend to have many nodes where the output is stuck at 0 when noise is supplied (not enough momentum in a specific direction), further reducing the overfitting on high frequency noise.

We can see that the STVO-based model demonstrated the ability to capture the low-frequency characteristics of the sunspot number time series. However, in several trials[25] using identical hyperparameters, the model did not match the frequency of the actual sunspot cycles for the first 4 (but remained stable for all trials). This inconsistency in fitting the low-frequency component accurately was also observed in the tanh-based model (and was a slightly stronger).

[24]More details in Annex 10.5.1.

[25]6 out of 10 for the STVO-based model and 8 out of 10 for the tanh-based model.

Conclusion

Using models and simulations, we explored the potential of STVOs in the context of energy efficient, hardware implementation of reservoir computing with a focus on time series prediction. We evaluated the short-term memory capacity of such STVO based RC network with emphasis on the STVO parameters. We benchmarked this model against NARMA2, followed by Mackey-glass. To finally apply it to a real world system.

We learned about the relationship between the STVO-parameters and the STMC of the network, which helped us understand some of the results from the subsequent benchmarks. We've also seen that STVO-based networks are less sensitive to typical RC parameters. This suggests that the intrinsic properties of STVOs may offer robustness to variations in these parameters, potentially simplifying the tuning process for hardware implementations.

While the STVO-based model performed well against other physical RC as seen in the NARMA2 benchmark, it generally did not achieve the same level of accuracy and performance as the tanh-based model. Those results are still promising, considering the fact that it presents a significant advantage in terms of hardware efficiency as the activation function inherent to STVOs can be performed entirely in hardware, reducing the need for additional computational resources and power consumption.

This hardware-centric approach holds promise for developing more energy-efficient machine learning systems. The reduced computational overhead associated with the STVO-based model, despite its current performance limitations, suggests that further refinement and optimization could lead to practical and scalable solutions for energy-constrained environments. This highlights the potential of spintronic devices in contributing to the advancement of sustainable and efficient machine learning hardware. Future work should focus on improving the performance of STVO-based models while leveraging their inherent hardware efficiency to create more balanced and effective machine learning systems.

Annex

10.1 Results when $I_w < 1$

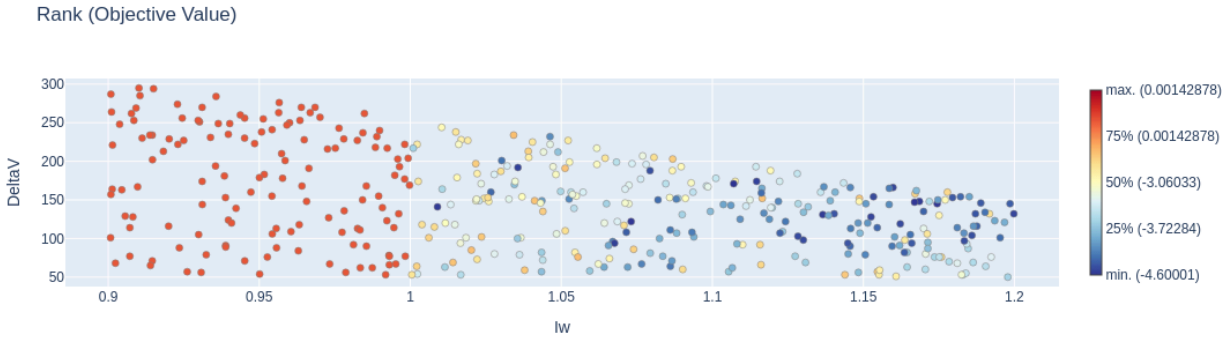


Fig. 30: Relation of I_w and ΔV with the target $\log_{10}$(NMSE) and each others

These results have been obtained on NARMA2. We can see that below a I_w of 1 all the results degrades significantly.

10.2 STMC

10.2.1 Parameters used for STMC characterization

To facilitate the study of the effect of the STVO parameters on STMC we fixed most of the reservoir parameters to :

Number of nodes	Spectral radius	RC connectivity	Input connectivity	Input scaling	Leaking rate
100	1.0	0.1	0.4	1.0	1.0

Table 13: Fixed parameters for STMC characterization

10.2.2 Effect of the spectral radius on MC

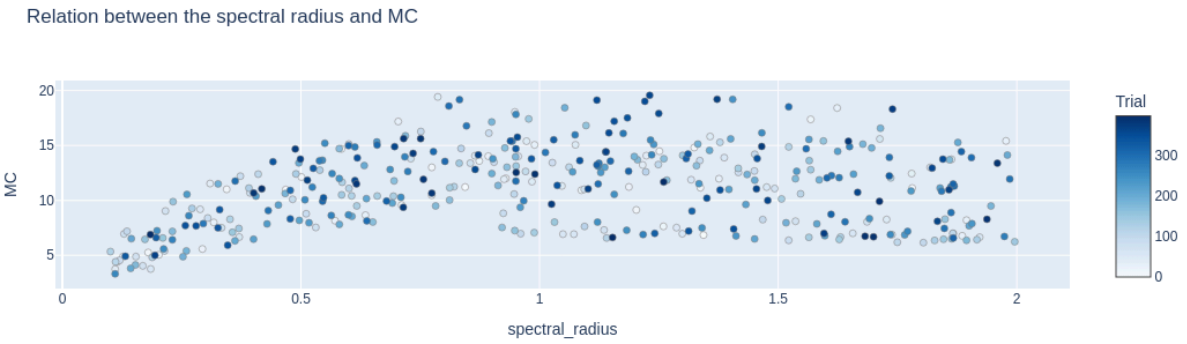


Fig. 31: Relation of the spectral radius and MC for a tanh based RC network with 300 nodes. We can see that lower spectral radii are linked to lower memory capacity.

10.3 NARMA2

10.3.1 Benchmark process

Here are the steps of our benchmarking process using NARMA2:

1. Generate a random input sequence $u(t)$ of length N.
2. Generate the output sequence $y(t)$ based on the NARMA2 equation and $u(t)$
3. Create input-output pairs, where the input to the predictive model at time t includes both $u(t)$ and y(t).
4. Train the predictive model using the generated input-output pairs.
5. Evaluate the model's performance on a test set using the Normalized Mean Squared Error

We used 8000 entries for the training set, 8000 entries for the test set and 8000 entries for the validation set where the model had to continue the time series only using $\hat{y}(t)$ predicted previously (not the ground truth) as well as $u(t)$.

10.3.2 Hyperparameter space mapping

800 models were generated and the hyperparameter space was :

Number of nodes	I_w	ΔV	Δt	Spectral radius
300	$[1, 1.2]$	$[10, \text{maxDV}^{26}]$	$[1, 2000]$	$[0.2, 2]$

RC connectivity	Input connectivity	Input scaling	Leaking rate
$[0.05, 0.8]$	$[0.05, 0.8]$	1	$[0.5, 1.5]$

Table 14: Hyperparameter space for NARMA2

[26]This is the maximum ΔV allowed which changes for a given I_w. It is determined so that the vortex core do not get expelled from the STVO.

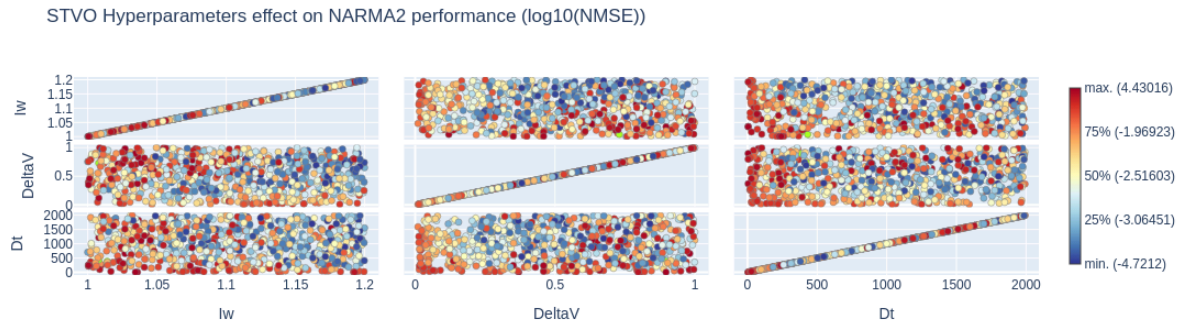


Fig. 32: Relationship between the STVO's parameters and the target. Higher I_{w} and ΔV favored.

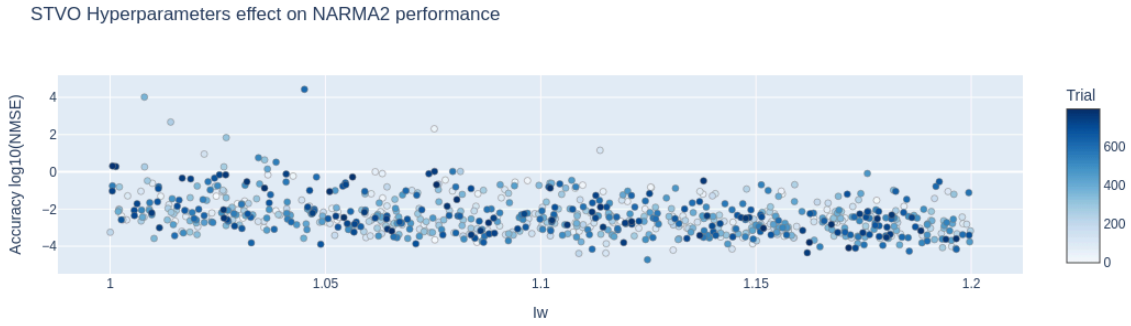


Fig. 33: Relationship between I_{w} and the target. Higher I_{w} favored.

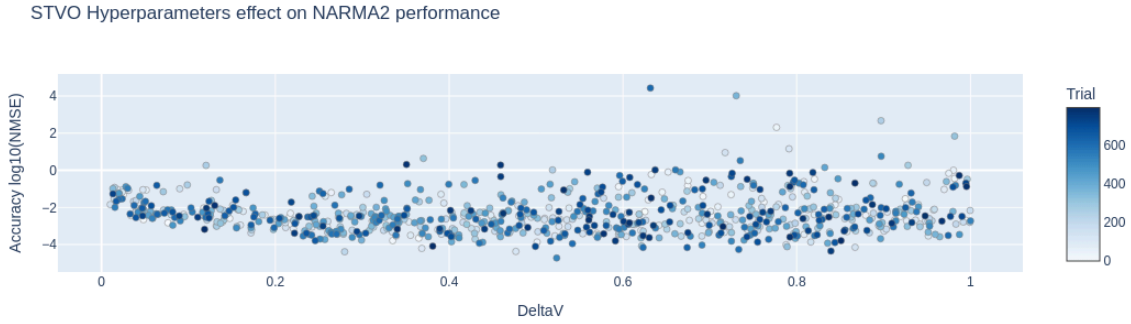


Fig. 34: Relationship between ΔV(as a percentage of maxDV) and the target. Higher ΔV favored up to a point

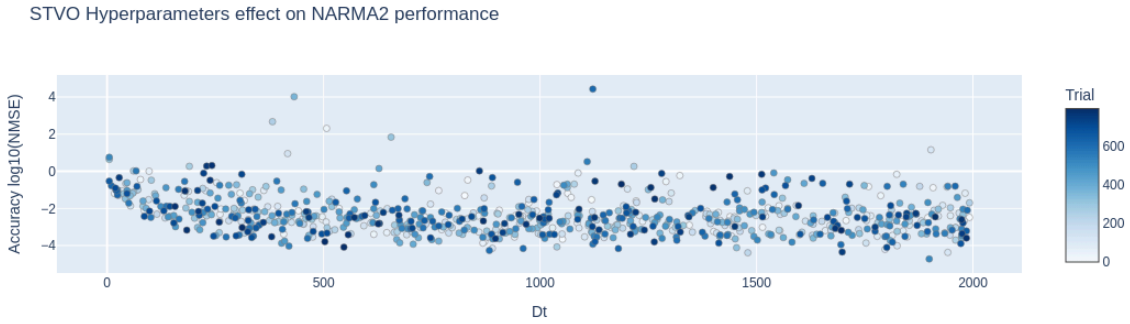


Fig. 35: Relationship between Δt and the target. Noticeable effect at very low Δt, no strong relationship after that.

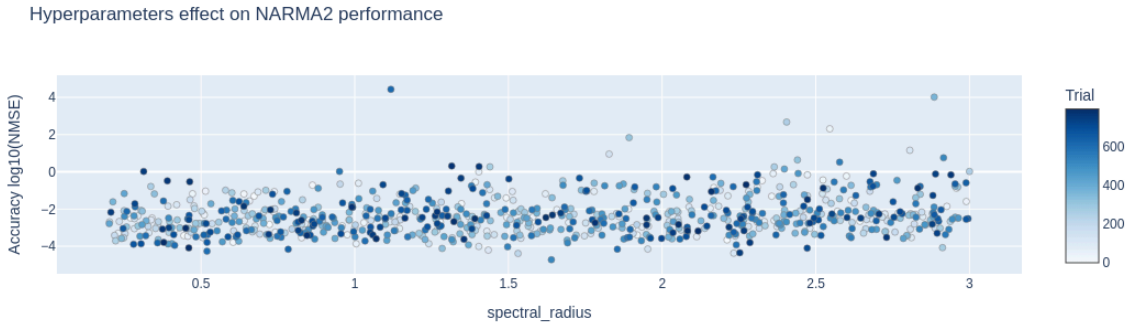


Fig. 36: Relationship between the spectral radius and the target. Very slight loss in accuracy when the spectral radius increases.

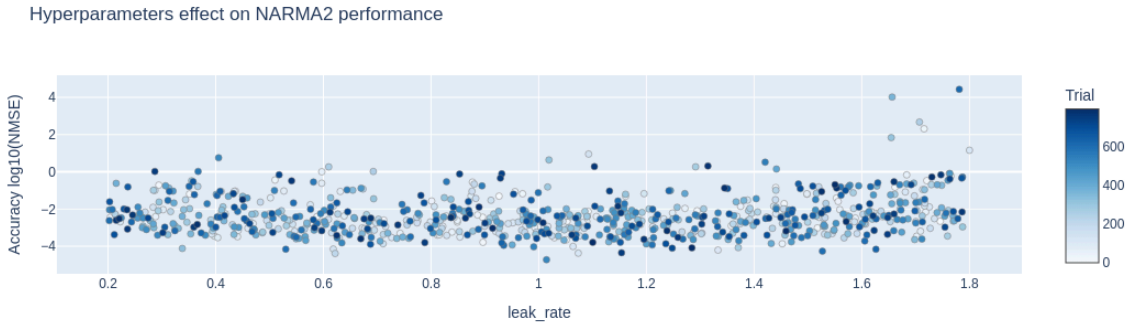


Fig. 37: Relationship between the leakage rate and the target. An optimum seems to be reached at around 1.30. After that accuracy starts to spread all over the place with the best results still slowly decreasing in accuracy.

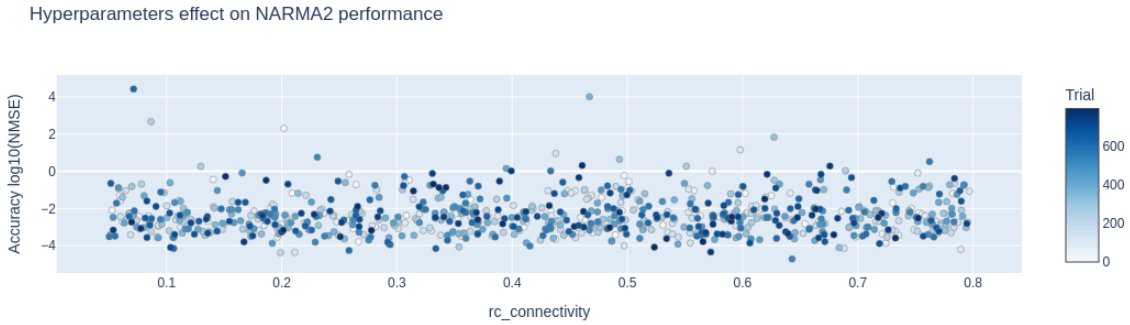


Fig. 38: Relationship between RC connectivity and the target. No strong relationship

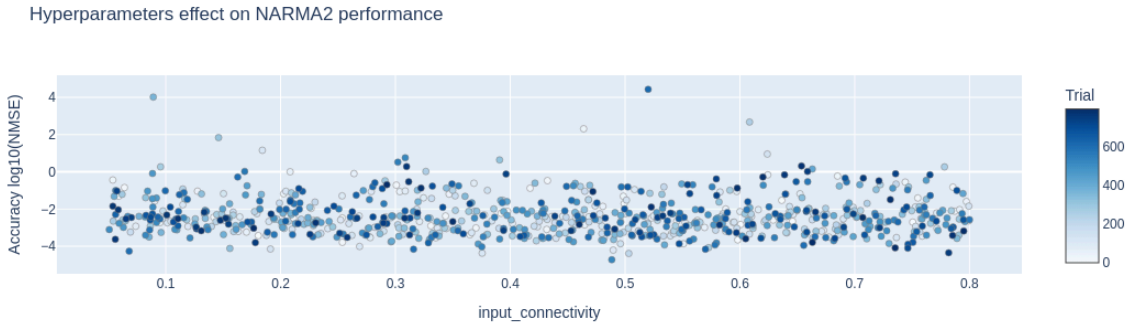


Fig. 39: Relationship between input connectivity and the target. No strong relationship

10.3.3 Results using regular tanh RC networks.

Number of nodes	Spectral radius	RC connectivity	Input connectivity	Leaking rate
300	0.2	0.274	0.672	0.981

Table 15: Optimal hyperparameters used for tanh reservoir

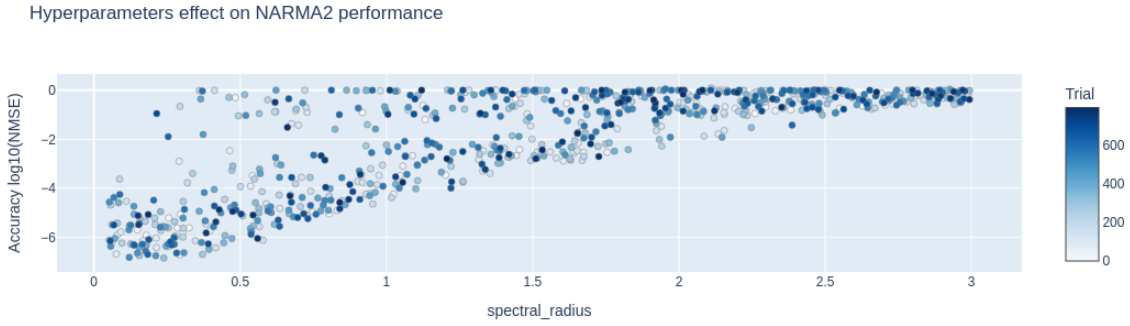


Fig. 40: Relationship between the spectral radius and the target. Very strong relation with an optimum around 0.2.

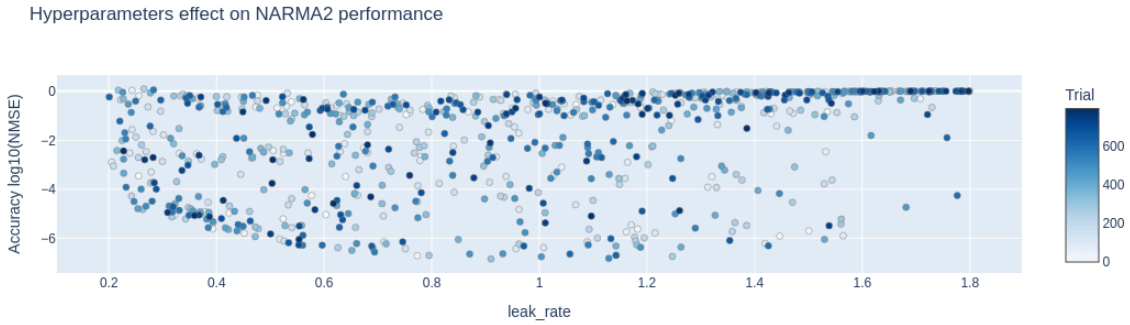


Fig. 41: Relationship between the leakage rate and the target. An optimum seems to be reached at around 1. After 1.2, accuracy starts get very bad with the best results still slowly decreasing in accuracy.

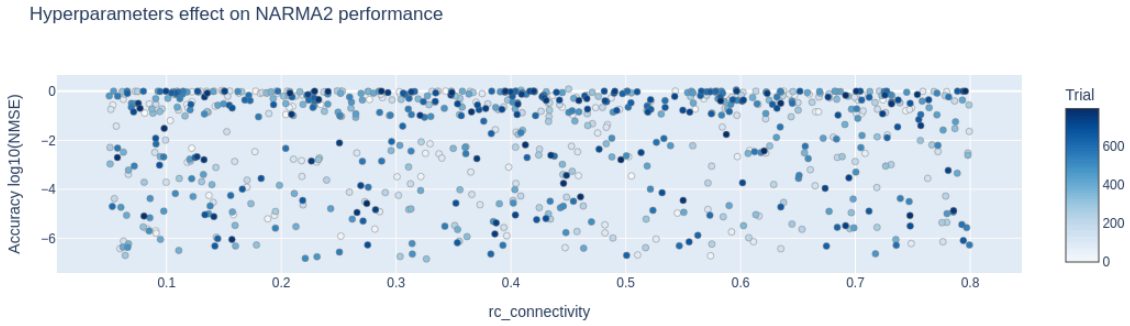


Fig. 42: Relationship between RC connectivity and the target. No strong relationship

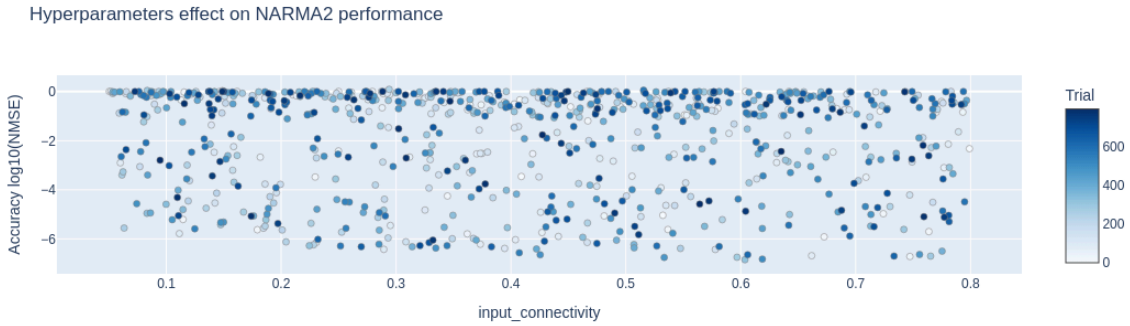


Fig. 43: Relationship between input connectivity and the target. Slight improvement in accuracy when the input connectivity increases specially noticeable at low values of it.

While using tanh yield better results we noticed that an STVO based RC network was much less sensitive to the spectral radius value as well as the leakage rate.

10.3.4 Additional hyperparameter mapping for sigmoid reservoir

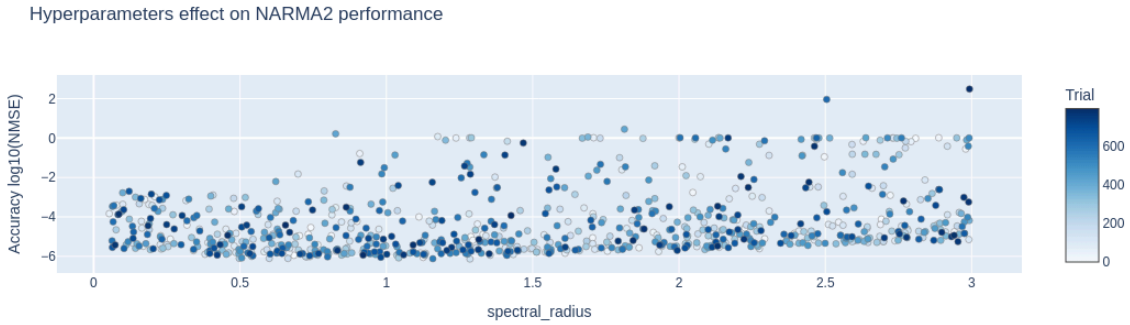


Fig. 44: Relationship between the spectral radius and the target. Noticeable relationship with an optimum around 1.

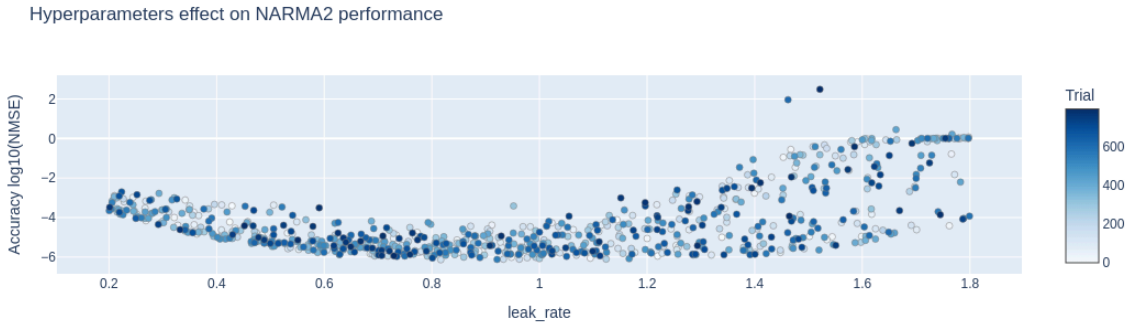


Fig. 45: Relationship between the leakage rate and the target. Very strong relation with an optimum around 0.9.

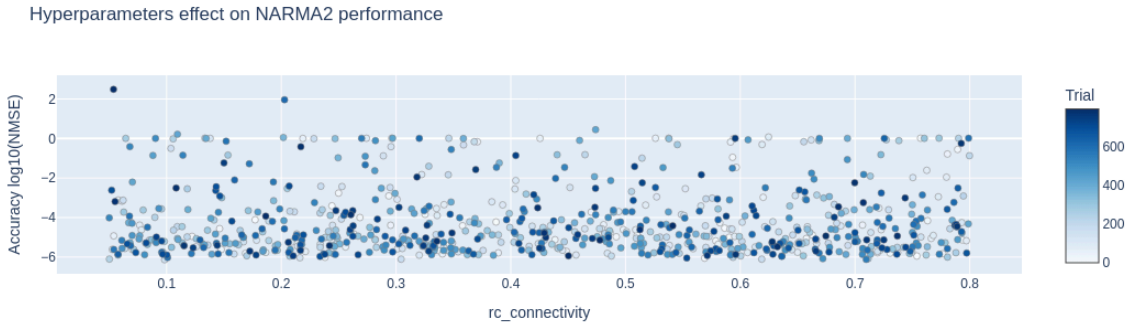


Fig. 46: Relationship between RC connectivity and the target. No strong relationship

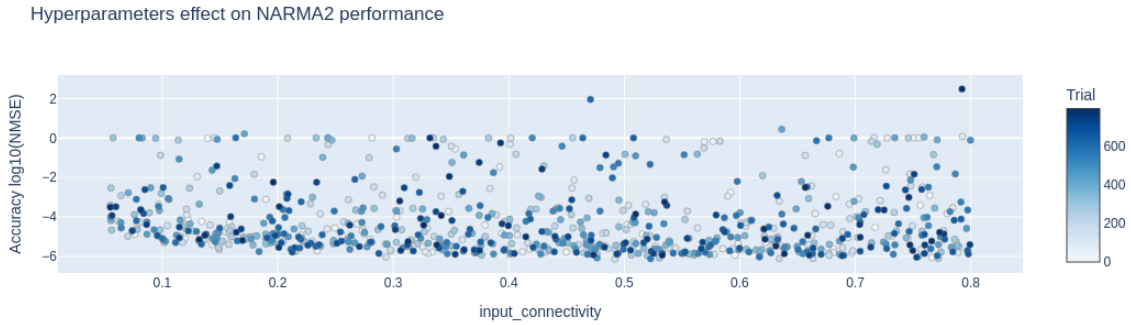


Fig. 47: Relationship between input connectivity and the target. Slight improvement in accuracy when the input connectivity increases specially noticeable when it is low.

10.4 Mackey-Glass

10.4.1 Hyperparameter space mapping

Similar to NARMA2, we used 8000 entries for the training set, 8000 entries for the test set and 8000 entries for the validation set where the model had to continue the time series only using $\hat{y}(t)$ predicted previously (not the ground truth) as well as $u(t)$.

800 models were generated and the hyperparameter space was :

Number of nodes	I_w	ΔV	Δt	Spectral radius
300	$[1, 1.2]$	$[10, \text{maxDV}^{26}]$	$[1, 2000]$	$[0.2, 3]$

RC connectivity	Input connectivity	Input scaling	Leaking rate
$[0.05, 0.8]$	$[0.05, 0.8]$	1	$[0.2, 1.8]$

Table 16: Hyperparameter space for Mackey-Glass

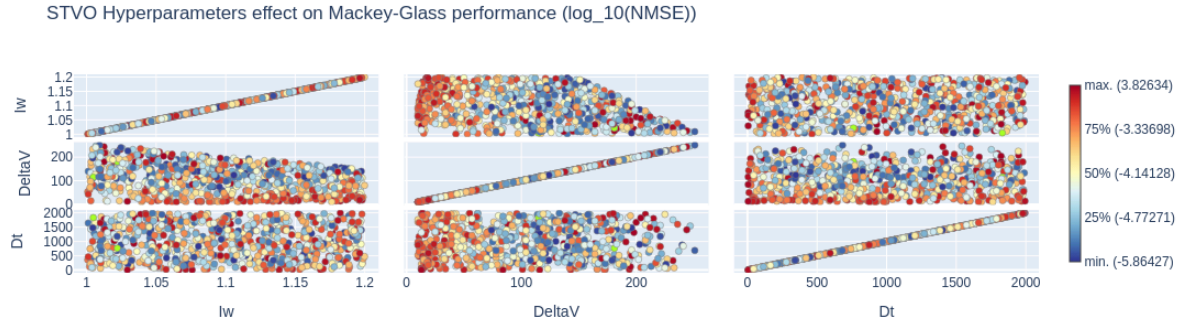


Fig. 48: Relationship between the STVO's parameters and the target. Higher ΔV favored.

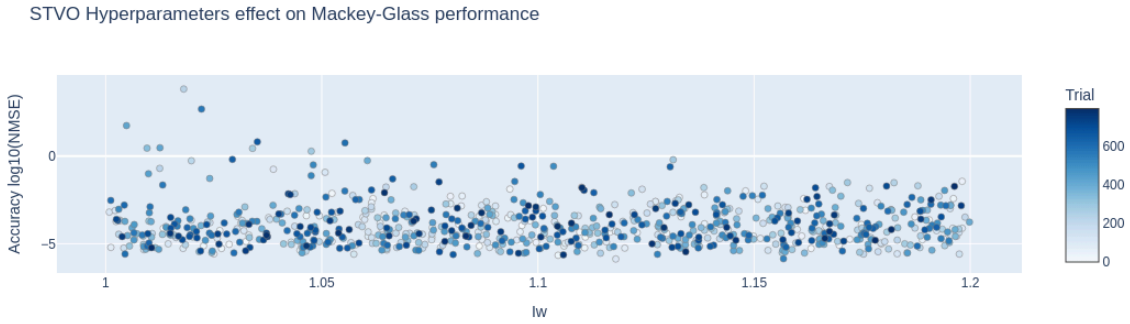


Fig. 49: Relationship between I_w and the target. No strong relationship except for very low I_w the spread in accuracy is more noticeable and very high I_w which reduce accuracy (also reducing max ΔV)

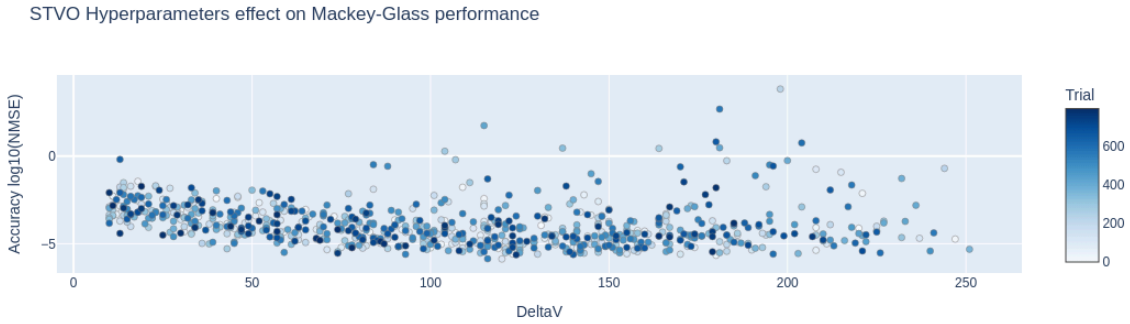


Fig. 50: Relationship between ΔV and the target. Higher ΔV favored. There are less data points at high ΔV because higher I_w limits the maximum ΔV.

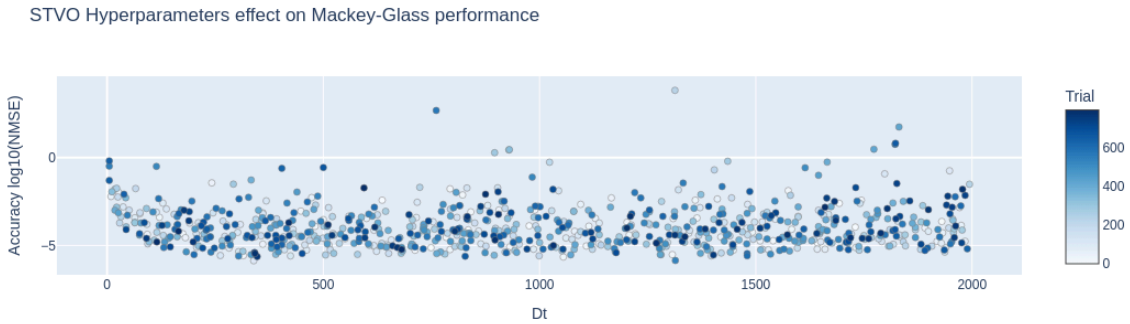


Fig. 51: Relationship between Δt and the target. No strong relationship except at very low Δt.

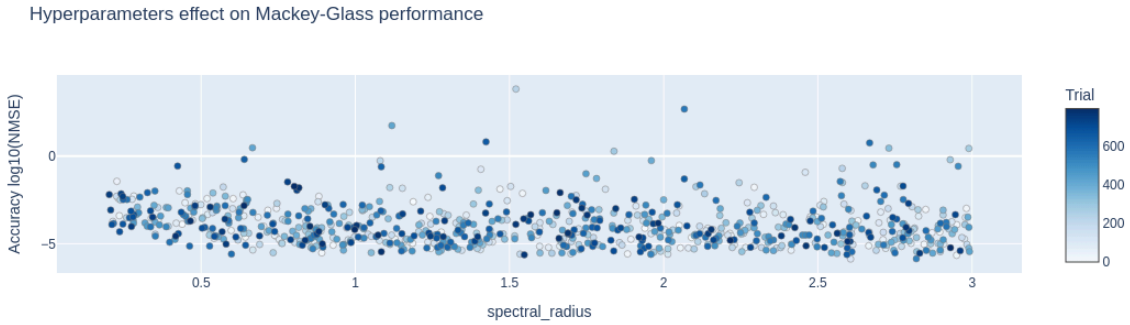


Fig. 52: Relationship between the spectral radius and the target. Loss of accuracy when the spectral radius is below approximately 1. No strong relationship after.

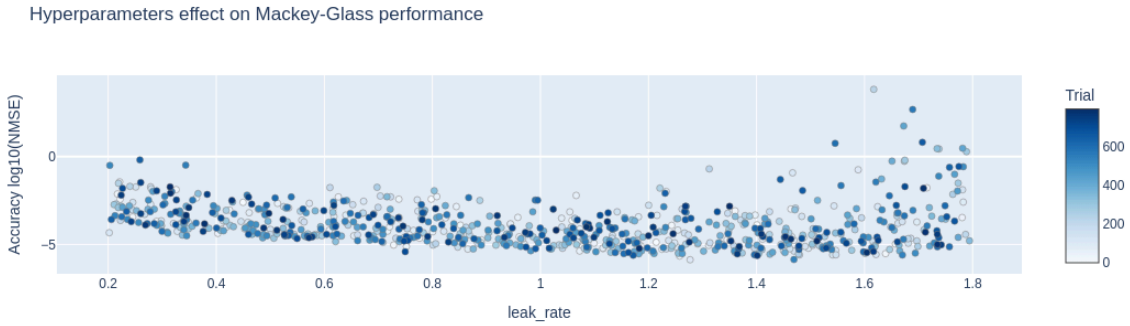


Fig. 53: Relationship between the leakage rate and the target. An optimum seems to be reached at around 1.30. After that accuracy starts to spread all over the place with the best results still slowly decreasing in accuracy.

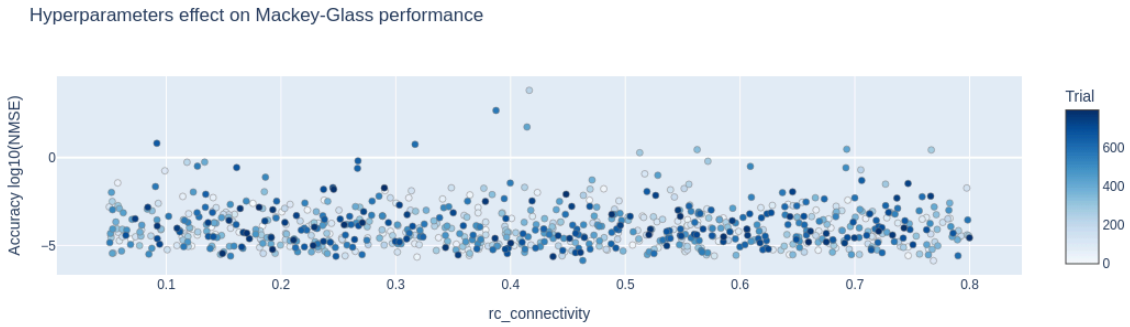


Fig. 54: Relationship between RC connectivity and the target. No strong relationship

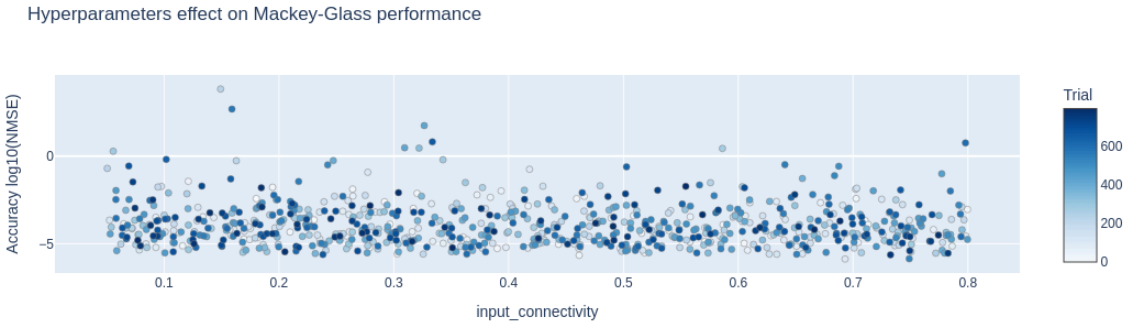


Fig. 55: Relationship between input connectivity and the target. No strong relationship

10.4.2 Results using regular (non physical) RC networks.

Number of nodes	Spectral radius	RC connectivity	Input connectivity	Leaking rate
300	1.546	0.2	0.407	0.913

Table 17: Optimal hyperparameters used for tanh reservoir

After 10 runs with the same parameters but different seeds the results are :

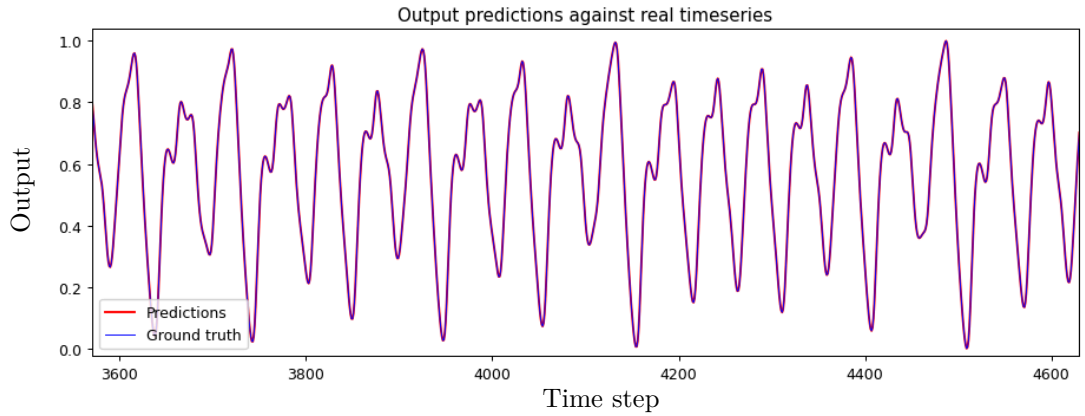


Fig. 56: A sample result of our model on the test set when $y(t)$ is fed from the ground truth

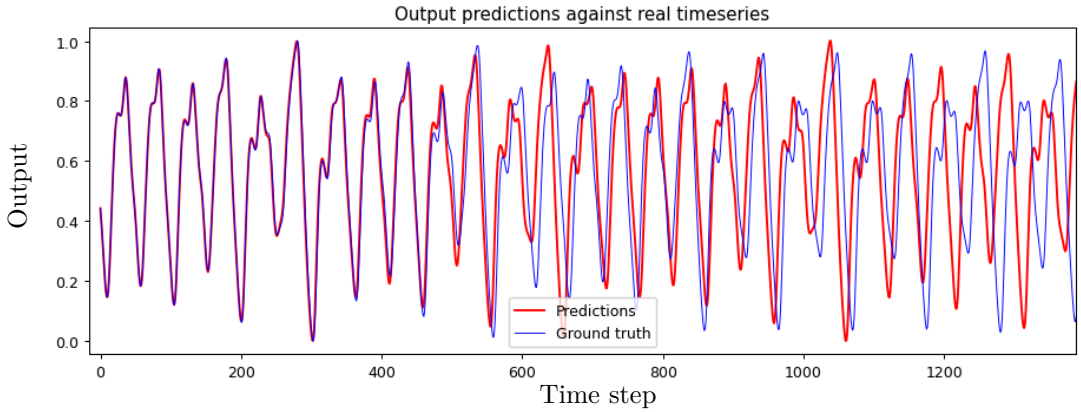


Fig. 57: A sample result of our model on the validation set when $\hat{u}(t)$ is fed from the model's previous

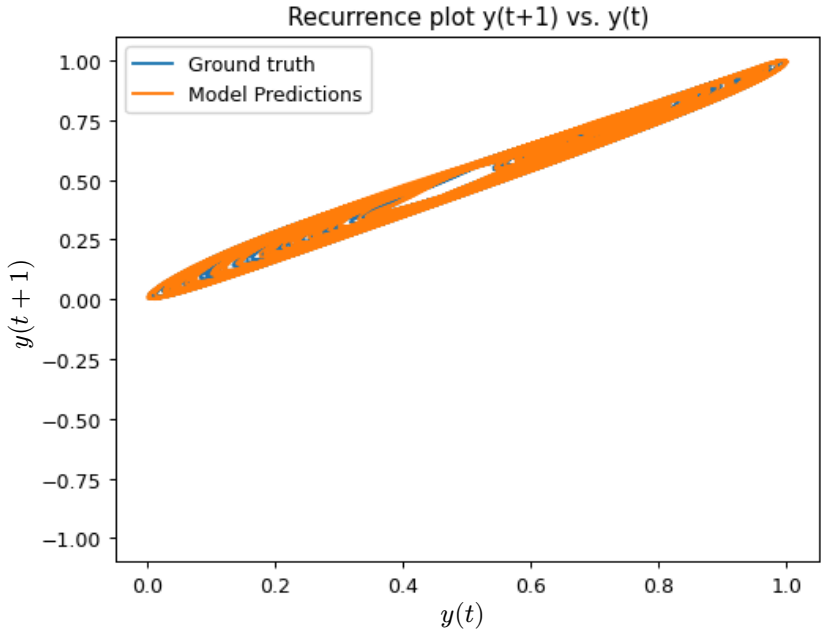


Fig. 58: The recurrence plot of the result obtained in Fig. 57 comparing the prediction with the ground truth.

We can see that using tanh results in more precise value on Fig. 56, longer coherence in Fig. 57 and a richer coverage of the recurrence plot in Fig. 58 as compared to the results obtained in Section 7.2.

Finally when it came to the sigmoid, we spent less time on it but observed that the STVO outperformed it when it came to prediction using it's output as input as the sigmoid ability to cover the recurrence plot was poor, as you can see in Fig. 59 and Fig. 60(probably due to its low MC score), and the stability was worse too. When it came to raw accuracy using the ground truth as input, the reservoir using the sigmoid was generally better (around 4.485e-07 of NMSE).

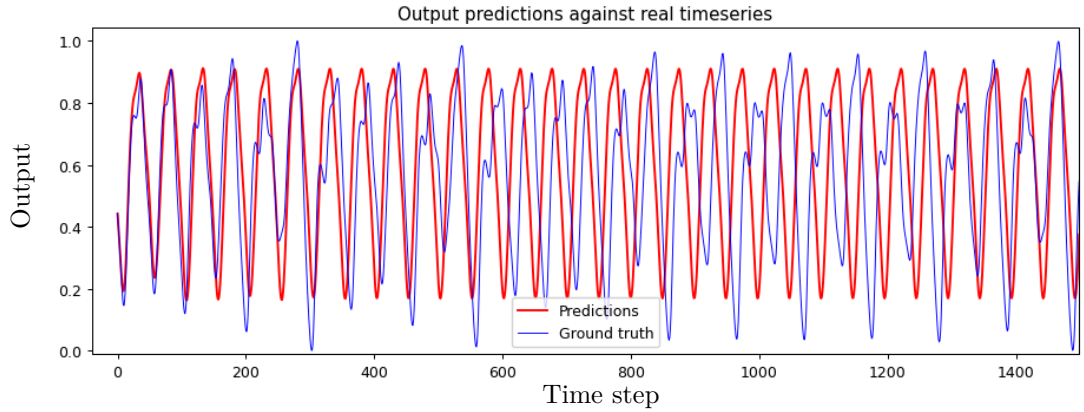


Fig. 59: A sample result of the sigmoid based model on the validation set when $\hat{u}(t)$ is fed from the

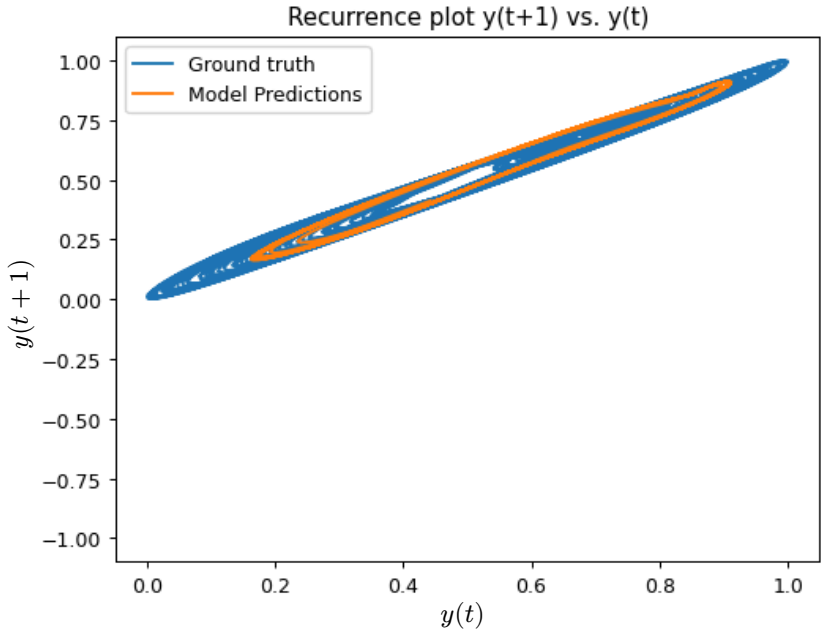


Fig. 60: The recurrence plot of the result obtained in Fig. 59 comparing the prediction with the ground truth.

10.4.3 Hyperparameter space mapping for tanh reservoir

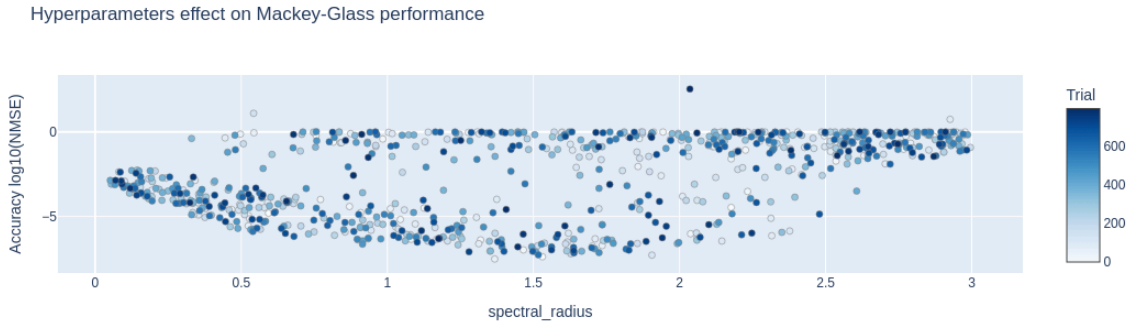


Fig. 61: Relationship between the spectral radius and the target. Very strong relationship with an optimum around 1.5.

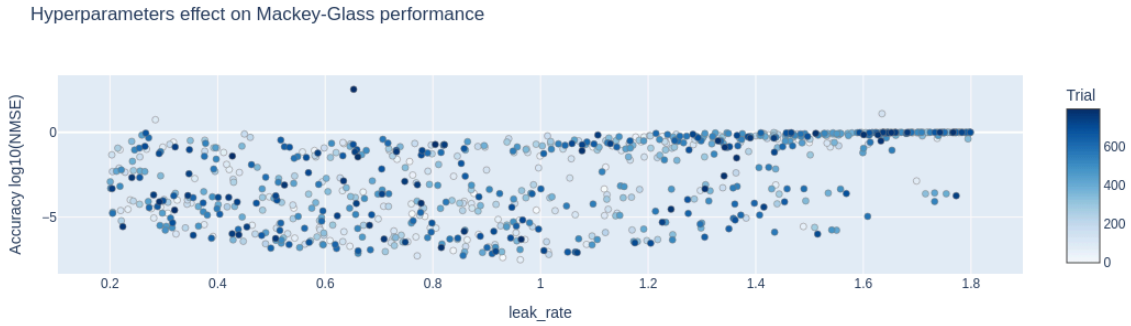


Fig. 62: Relationship between the leakage rate and the target. An optimum seems to be reached at around 1. After that accuracy starts to spread all over the place with the best results still slowly decreasing in accuracy.

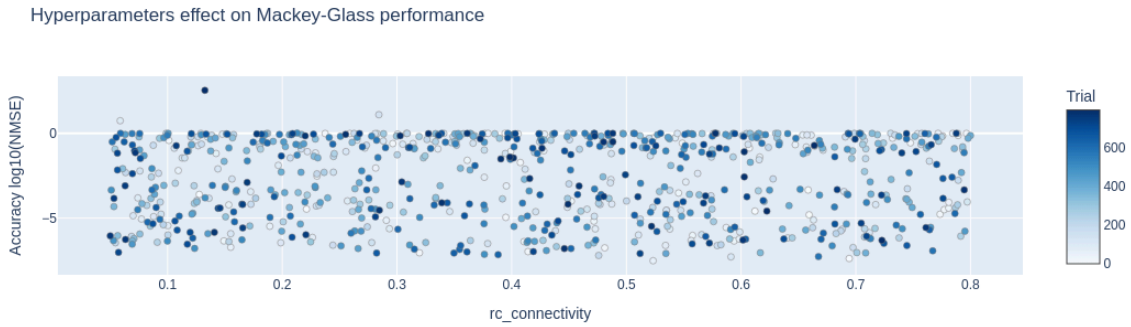


Fig. 63: Relationship between RC connectivity and the target. No strong relationship

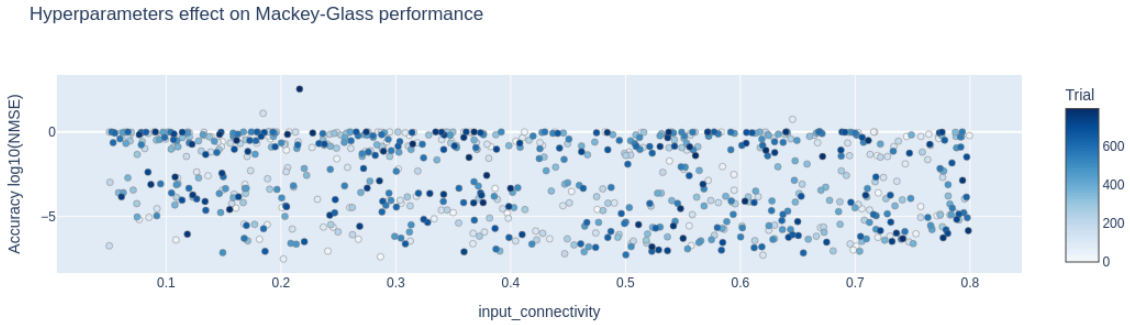


Fig. 64: Relationship between input connectivity and the target. No strong relationship

10.4.4 Additional hyperparameter mapping for sigmoid reservoir

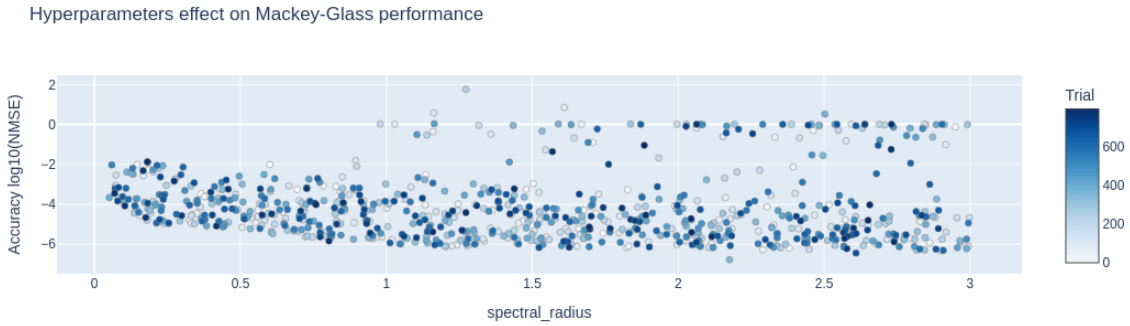


Fig. 65: Relationship between the spectral radius and the target. Noticeable relationship with an optimum around 1.5.

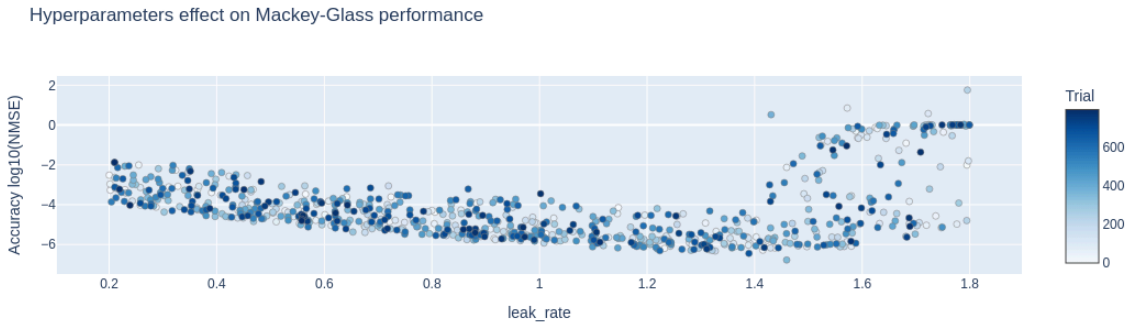


Fig. 66: Relationship between the leakage rate and the target. Very strong relation with an optimum around 0.9.

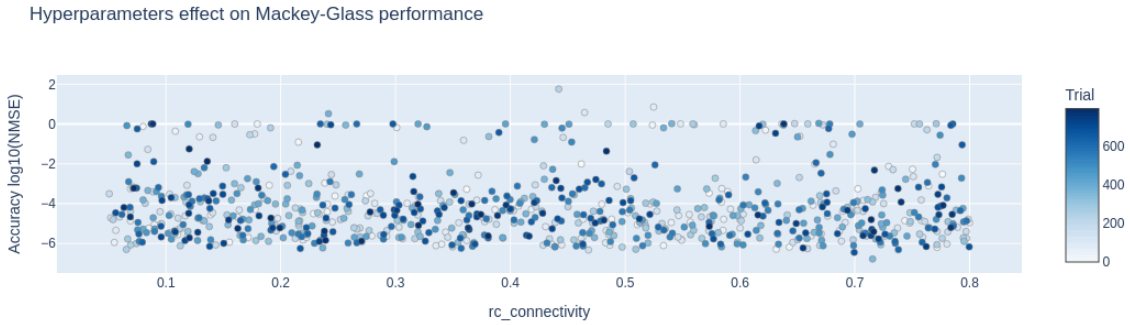


Fig. 67: Relationship between RC connectivity and the target. No strong relationship

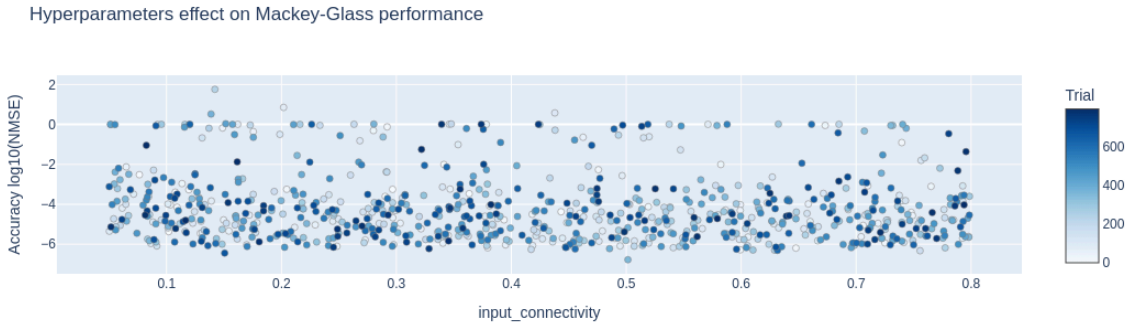


Fig. 68: Relationship between input connectivity and the target. No strong relationship

10.5 Sunspot number

10.5.1 Hyperparameter space mapping

We used 1800 entries for the training set, 700 entries for the test set and 700 entries for the validation set where the model had to continue the time series only using $\hat{y}(t)$ predicted previously (not the ground truth) as well as $u(t)$.

800 models were generated and the hyperparameter space was :

Number of nodes	I_w	ΔV	Δt	Spectral radius
300	$[1, 1.2]$	$[10, \text{maxDV}^{26}]$	$[1, 2000]$	$[0.2, 3]$

RC connectivity	Input connectivity	Input scaling	Leaking rate
$[0.05, 0.8]$	$[0.05, 0.8]$	1	$[0.2, 1.8]$

Table 18: Hyperparameter space for Mackey-Glass

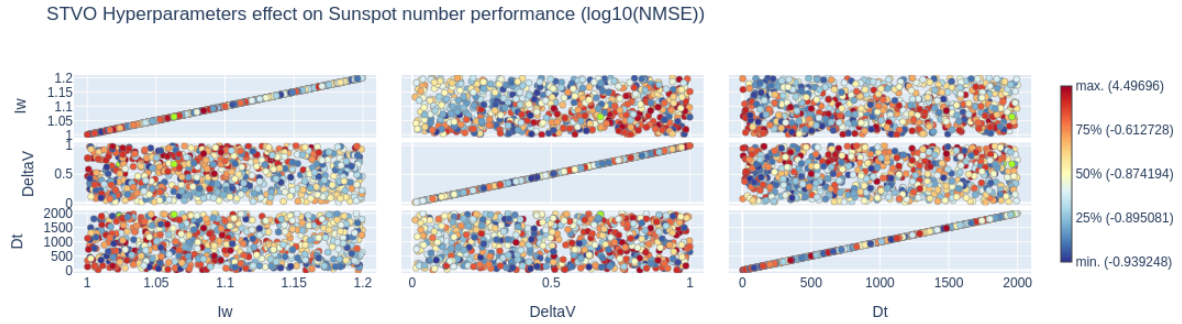


Fig. 69: Relationship between the STVO's parameters and the target. Higher ΔV favored.

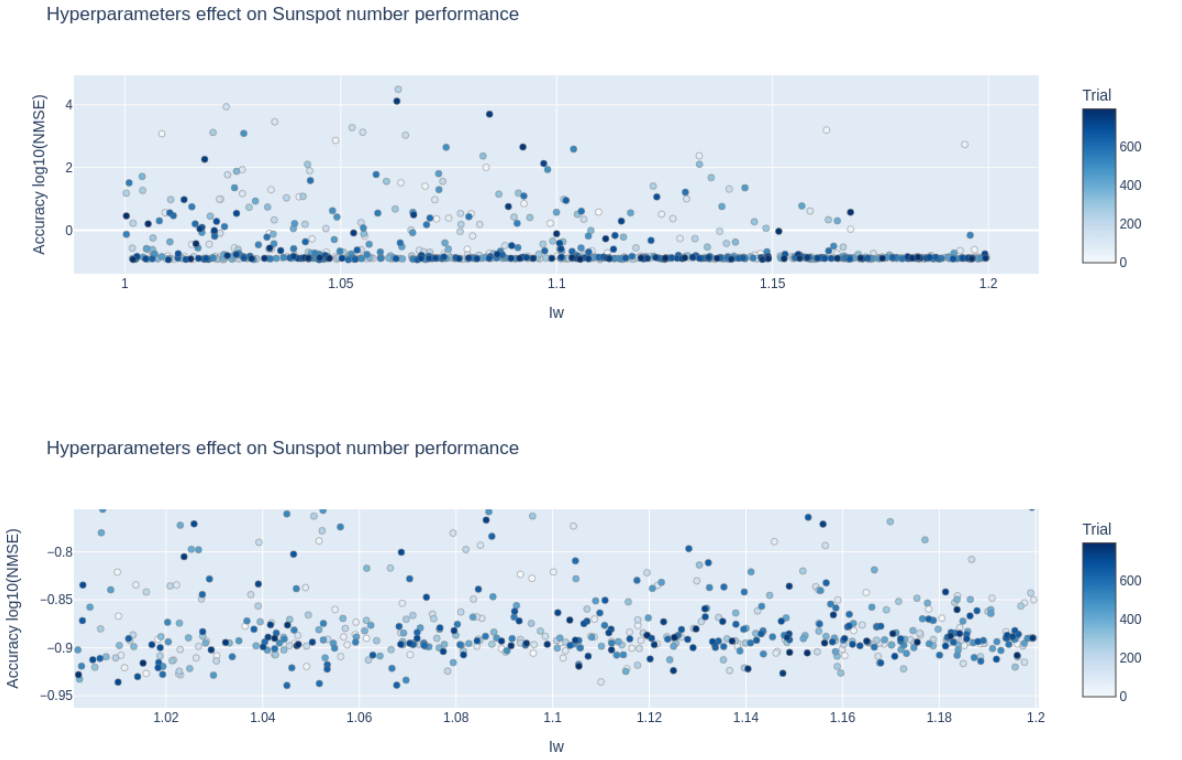


Fig. 70: Relationship between I_{w} and the target. The variability of the results seems to increase when I_{w} decreases. In the zoomed in version the best results tend to be reached at lower I_{w}.

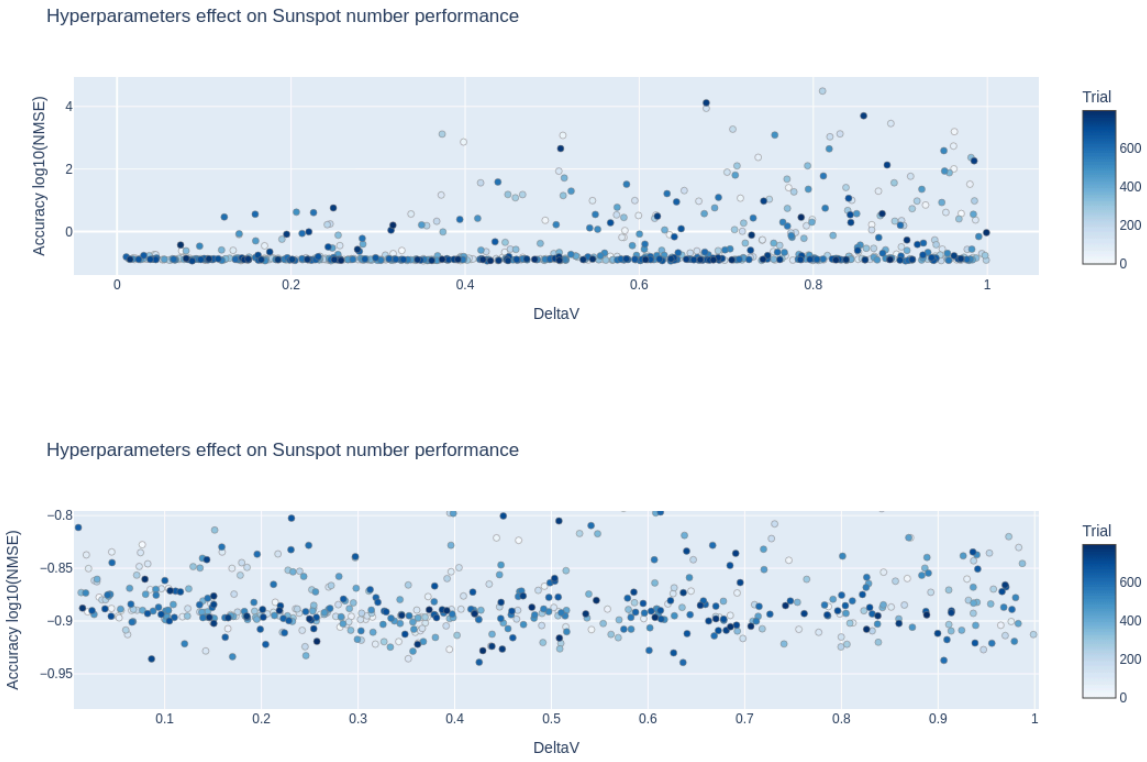


Fig. 71: Relationship between ΔV (relative to maxDV) and the target. The variability of the results seems to increase when ΔV increases. In the zoomed in version, very small ΔVs resulted in worse results. After that, the performance increase quickly saturates.

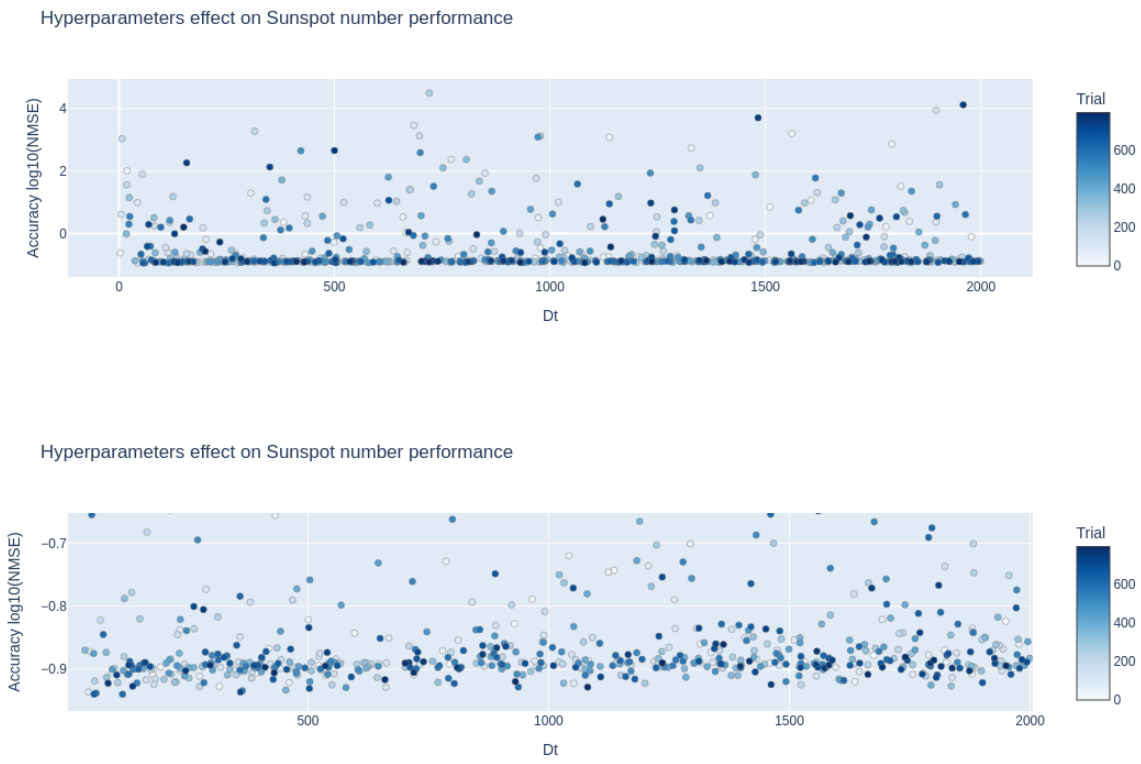


Fig. 72: Relationship between Δt and the target. No strong relationship except at very low Δt. In the zoomed in version we can see that the best performing models have relatively low Δt(lower than that rapidly yield bad performance).

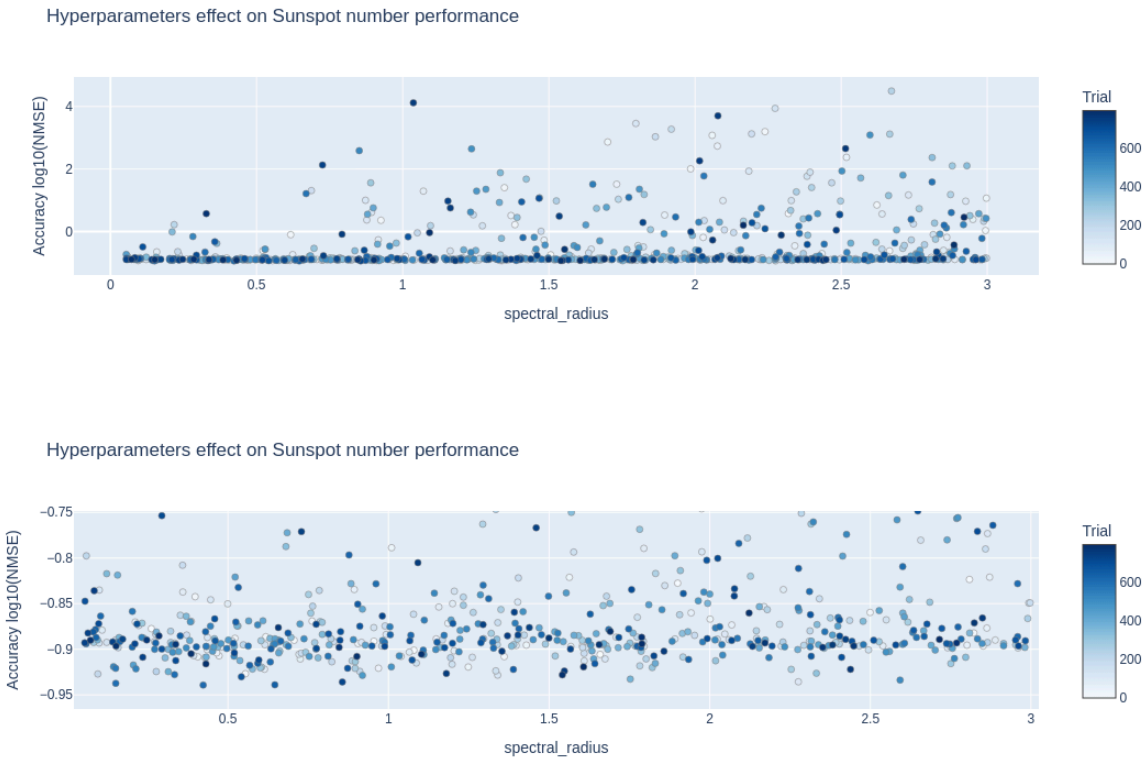


Fig. 73: Relationship between the spectral radius and the target. The variability of the results seems to increase when the spectral radius increases. In the zoomed in version the best results tend to be reached at lower spectral radii.

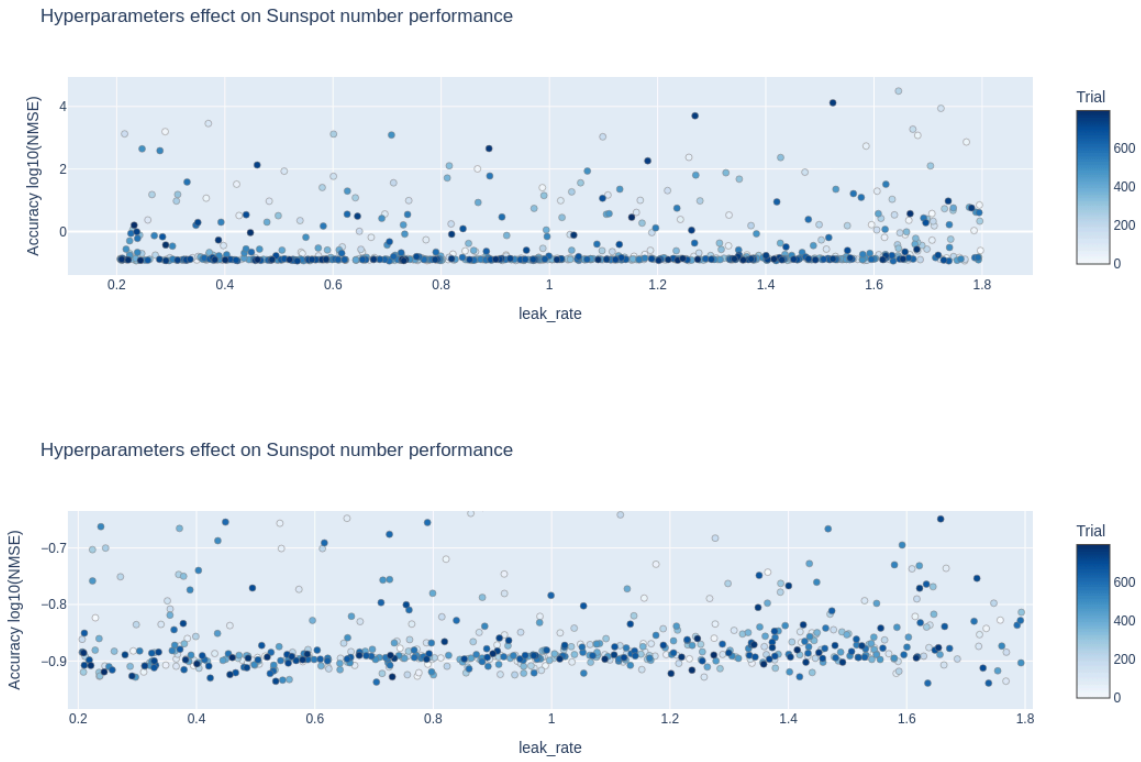


Fig. 74: Relationship between the leakage rate and the target (and a zoomed in version). Almost no effect if not for a slight decrease in performance when the leaking rate increases.

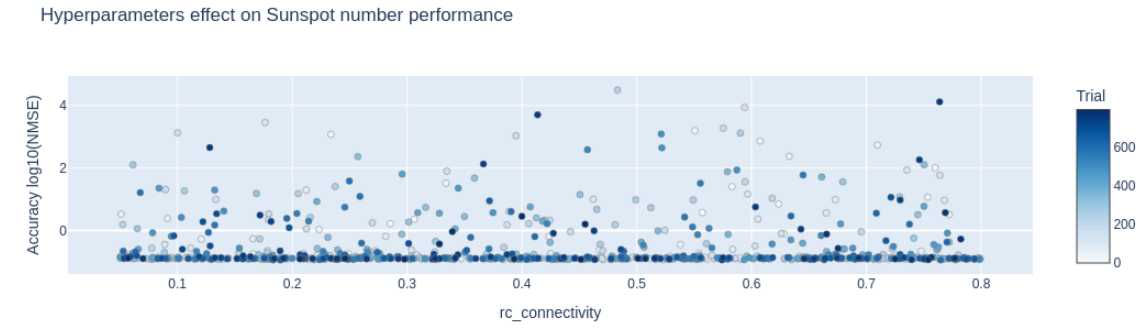


Fig. 75: Relationship between RC connectivity and the target. No strong relationship

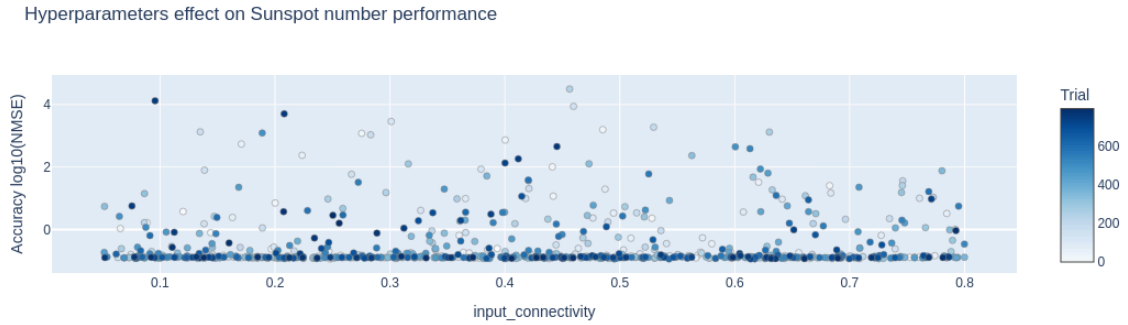


Fig. 76: Relationship between input connectivity and the target. No strong relationship

10.5.2 Results using regular (non physical) RC networks.

Number of nodes	Spectral radius	RC connectivity	Input connectivity	Leaking rate
300	0.672	0.583	0.301	0.238

Table 19: Optimal hyperparameters used for tanh reservoir

10.5.3 Hyperparameter space mapping for tanh reservoir

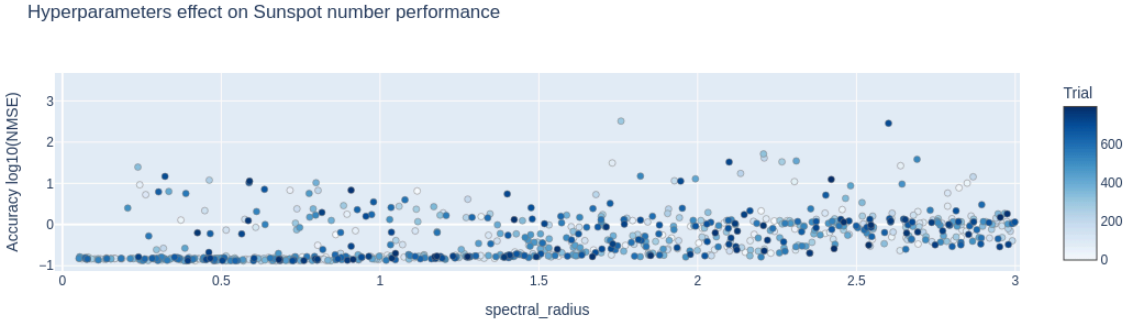


Fig. 77: Relationship between the spectral radius and the target. Very strong relationship where smaller spectral radii yield better results.

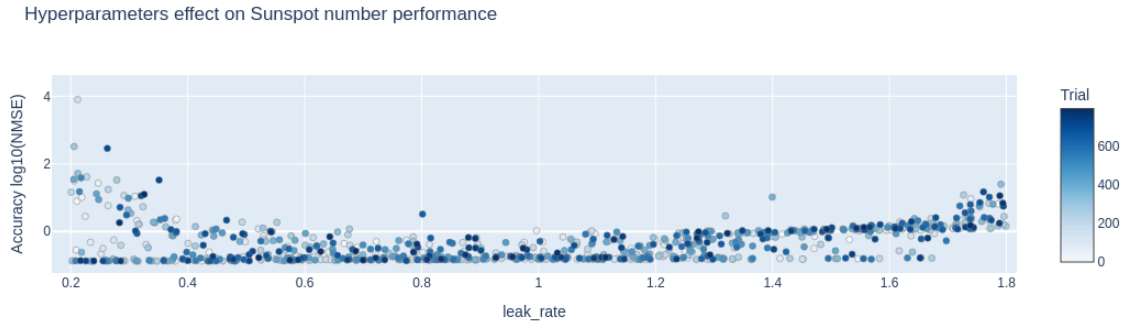


Fig. 78: Relationship between the leakage rate and the target. An optimum seems to be reached at around 1. After 1.6 even the best results decrease in accuracy.

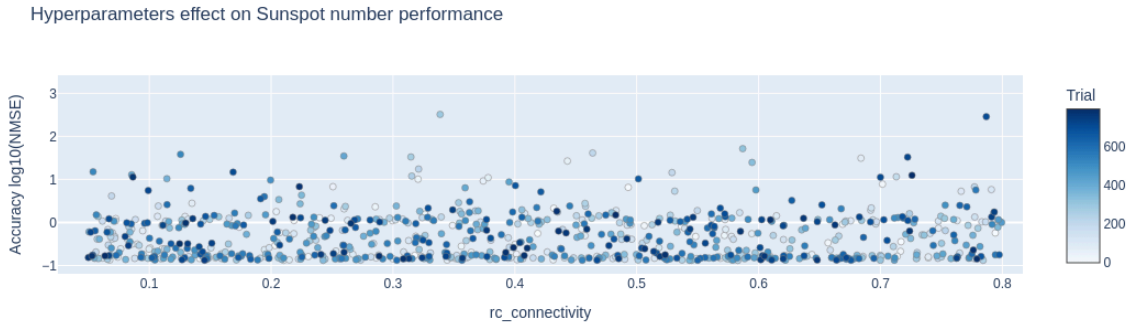


Fig. 79: Relationship between RC connectivity and the target. No strong relationship

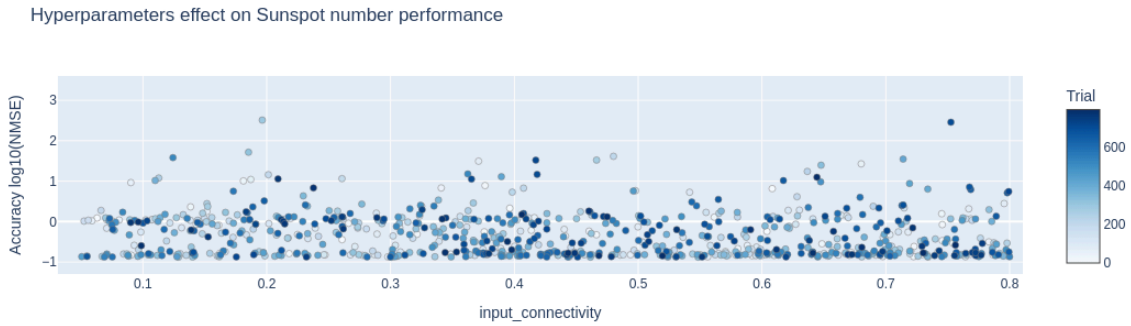


Fig. 80: Relationship between input connectivity and the target. No strong relationship

Bibliography

[1] H. Jaeger, "The "echo state" approach to analysing and training recurrent neural networks-with an erratum note," *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, vol. 148, no. 34, p. 13–14, 2001.

[2] W. Maass, T. Natschläger, and H. Markram, "Real-time computing without stable states: A new framework for neural computation based on perturbations," *Neural computation*, vol. 14, no. 11, pp. 2531–2560, 2002.

[3] M. Lukoševičius and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," *Computer science review*, vol. 3, no. 3, pp. 127–149, 2009.

[4] B. Schrauwen, D. Verstraeten, and J. Van Campenhout, "An overview of reservoir computing: theory, applications and implementations," in *Proceedings of the 15th european symposium on artificial neural networks. p. 471-482 2007*, 2007, pp. 471–482.

[5] G. Indiveri and S.-C. Liu, "Memory and information processing in neuromorphic systems," *Proceedings of the IEEE*, vol. 103, no. 8, pp. 1379–1397, 2015.

[6] C. Gallicchio, A. Micheli, and L. Pedrelli, "Deep reservoir computing: A critical experimental analysis," *Neurocomputing*, vol. 268, pp. 87–99, 2017.

[7] A. Rodan and P. Tino, "Minimum complexity echo state network," *IEEE transactions on neural networks*, vol. 22, no. 1, pp. 131–144, 2010.

[8] J. Pathak, B. Hunt, M. Girvan, Z. Lu, and E. Ott, "Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach," *Physical review letters*, vol. 120, no. 2, p. 24102–24103, 2018.

[9] E. Tsymbal and D. Pettifor, "Perspectives of giant magnetoresistance," *Solid State Physics*, vol. 56. Elsevier, pp. 113–237, 2001. doi: 10.1016/S0081-1947(01)80019-9.

[10] N. Locatelli, V. Cros, and J. Grollier, "Spin torque building blocks," *Nature Materials*, vol. 13, no. 1, pp. 11–20, Jan. 2014, doi: 10.1038/NMAT3823.

[11] A. Moureaux, C. Chopin, S. de Wergifosse, L. Jacques, and F. A. Araujo, "Spintronics for image recognition: performance benchmarking via ultrafast data-driven simulations," 2023, doi: 10.48550/ARXIV.2308.05810.

[12] A. Moureaux, S. De Wergifosse, C. Chopin, J. Weber, and F. A. Araujo, "Neuromorphic spintronics accelerated by an unconventional data-driven Thiele equation approach." Accessed: Aug. 05, 2024. [Online]. Available: http://arxiv.org/abs/2301.11025

[13] F. Duport, A. Smerieri, A. Akrout, M. Haelterman, and S. Massar, "Fully analogue photonic reservoir computer," *Scientific reports*, vol. 6, no. 1, p. 22381–22382, 2016.

[14] N. Trouvain, L. Pedrelli, T. T. Dinh, and X. Hinaut, "ReservoirPy: An Efficient and User-Friendly Library to Design Echo State Networks," *Artificial Neural Networks and Machine Learning – ICANN 2020*. Springer International Publishing, pp. 494–505, 2020. doi: 10.1007/978-3-030-61616-8_40.

[15] H. Jaeger, "Short term memory in echo state networks," doi: 10.24406/PUBLICA-FHG-291107.

[16] S. Kan, K. Nakajima, Y. Takeshima, T. Asai, Y. Kuwahara, and M. Akai-Kasaya, "Simple Reservoir Computing Capitalizing on the Nonlinear Response of Materials: Theory and Physical Implementations," *Physical Review Applied*, vol. 15, no. 2, p. 24030–24031, Feb. 2021, doi: 10.1103/PhysRevApplied.15.024030.

[17] D. Nishioka, T. Tsuchiya, M. Imura, Y. Koide, T. Higuchi, and K. Terabe, "A high-performance deep reservoir computer experimentally demonstrated with ion-gating reservoirs," *Communications Engineering*, vol. 3, no. 1, p. 81–82, 2024.

[18] D. Nishioka *et al.*, "Edge-of-chaos learning achieved by ion-electron–coupled dynamics in an ion-gating reservoir," *Science Advances*, vol. 8, no. 50, p. eade1156, 2022.

[19] T. Wada, D. Nishioka, W. Namiki, T. Tsuchiya, T. Higuchi, and K. Terabe, "A Redox-Based Ion-Gating Reservoir, Utilizing Double Reservoir States in Drain and Gate Nonlinear Responses," *Advanced Intelligent Systems*, vol. 5, no. 9, p. 2300123–2300124, 2023.

[20] W. Namiki, D. Nishioka, Y. Yamaguchi, T. Tsuchiya, T. Higuchi, and K. Terabe, "Experimental Demonstration of High-Performance Physical Reservoir Computing with Nonlinear Interfered Spin Wave Multidetection," *Advanced Intelligent Systems*, vol. 5, no. 12, p. 2300228–2300229, 2023.

[21] S. Kan, K. Nakajima, T. Asai, and M. Akai-Kasaya, "Physical implementation of reservoir computing through electrochemical reaction," *Advanced Science*, vol. 9, no. 6, p. 2104076–2104077, 2022.

[22] M. Akai-Kasaya, Y. Takeshima, S. Kan, K. Nakajima, T. Oya, and T. Asai, "Performance of reservoir computing in a random network of single-walled carbon nanotubes complexed with polyoxometalate," *Neuromorphic Computing and Engineering*, vol. 2, no. 1, p. 14003–14004, 2022.

[23] "Sunspots." Accessed: May 18, 2024. [Online]. Available: https://www.kaggle.com/datasets/robervalt/sunspots

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain
Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/epl**