



UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO  
UNIDADE ACADÊMICA DO CABO DE SANTO AGOSTINHO  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA FÍSICA

THIAGO STREITENBERGER IVO MOREIRA

Previsão de eventos extremos em osciladores caóticos acoplados utilizando  
computação de reservatório

Cabo de Santo Agostinho - PE  
2024

THIAGO STREITENBERGER IVO MOREIRA

Previsão de eventos extremos em osciladores caóticos acoplados utilizando  
computação de reservatório

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Física da Unidade Acadêmica do Cabo de Santo Agostinho da Universidade Federal Rural de Pernambuco para a obtenção do título de Mestre em Engenharia Física.

Área de concentração: Optoeletrônica  
Orientador: Prof. Dr. Weliton Soares Martins

Cabo de Santo Agostinho - PE  
2024

Dados Internacionais de Catalogação na Publicação  
Universidade Federal Rural de Pernambuco  
Sistema Integrado de Bibliotecas  
Gerada automaticamente, mediante os dados fornecidos pelo(a) autor(a)

---

M838p

Moreira, Thiago Streitenberger Ivo  
Previsão de eventos extremos em osciladores caóticos acoplados utilizando computação de reservatório / Thiago Streitenberger Ivo Moreira. - 2024.  
82 f. : il.

Orientador: Weliton Soares Martins.  
Inclui referências.

Dissertação (Mestrado) - Universidade Federal Rural de Pernambuco, Programa de Pós-Graduação em Engenharia Física, Cabo de Santo Agostinho, 2024.

1. Eventos extremos. 2. Computação de reservatório. 3. Aprendizagem de máquina. 4. Circuitos caóticos sincronizados. 5. Otimização de parâmetros. I. Martins, Weliton Soares, orient. II. Título

CDD 621

---

THIAGO STREITENBERGER IVO MOREIRA

Previsão de eventos extremos em osciladores caóticos acoplados utilizando  
computação de reservatório

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Física da  
Unidade Acadêmica do Cabo de Santo Agostinho da Universidade Federal Rural de  
Pernambuco para a obtenção do título de Mestre em Engenharia Física.

Aprovada em: 26/01/2024

Banca Examinadora

---

Prof. Dr. Weliton Soares Martins  
Orientador

---

Prof. Dr. Hugo Leonardo Davi de Souza Cavalcante  
Examinador externo

---

Prof. Dr. Rafael Alves de Oliveira  
Examinador interno

## DEDICATÓRIA

*A minha filha, Alice,  
dedico esta conquista.*

## **AGRADECIMENTOS**

O mestrado é um caminho com muitos desafios e certamente as pessoas que nos suportam nesta caminhada são a energia para continuar caminhando.

A minha esposa, Priscilla e minha filha, Alice, pela paciência e suporte nas turbulências e pelos sacrifícios para que eu alcançasse a linha de chegada.

A meus pais, avós e irmãos, em especial, ao meu irmão Felipe pela motivação e pelo apoio de sempre e a meu avô André, por ter me mostrado a beleza do pensamento científico quando eu ainda estava nos primeiros passos da infância.

Ao meu orientador e amigo, Prof. Weliton Soares Martins, pelos sábios conselhos e por sempre me ajudar a encontrar o rumo certo.

Agradeço a toda equipe de professores, técnicos e gestores da UACSA-UFRPE, sempre disponíveis para o auxílio.

*“...And yet, if there was no chaos, things would be pretty boring...” do livro Extreme events in nature and society*

## RESUMO

Vários fenômenos naturais são não-lineares, a implicação disto é que dobrar alguma de suas variáveis, em geral, não aumentará ou reduzirá o efeito duas vezes, ao contrário, poderá provocar um escalonamento irregular do efeito. Em alguns casos, observa-se o surgimento de eventos extremos, que alcançam grandes amplitudes, ocorrem em intervalos aperiódicos e, em geral, não possuem uma causa bem definida. Exemplos são os terremotos, inundações, quebras (*crash*) de bolsas de valores, apagões (*blackouts*) em redes de fornecimento de energia, instabilidades em cavidades de lasers, crises de epilepsia, etc. Naturalmente, devido ao seu impacto econômico e social adverso, há grande interesse em prever, mensurar e, quando possível, mitigar sua ocorrência. Um tipo especial de evento extremo são os dragões-rei (*dragon-kings*), pois podem ser observados quando osciladores caóticos do tipo Gauthier-Bienfang estão em regime de forte acoplamento, permitindo seu estudo em ambiente de laboratório. Nesta pesquisa, o objetivo foi obter previsões das séries caóticas, das dessincronizações e dos eventos extremos a partir dos dados produzidos por osciladores caóticos sincronizados, em especial, utilizando um método de aprendizagem de máquina conhecido como computação de reservatório (*reservoir computing*), que é um tipo de rede neural recorrente, com a peculiaridade de ser treinada somente em sua camada de saída. Uma simulação numérica de dois circuitos eletrônicos osciladores caóticos do tipo Gauthier-Bienfang gerou os dados utilizados para o treino e posterior teste da capacidade de previsão da rede de computação de reservatório, que teve seus parâmetros globais ou hiperparâmetros otimizados. A rede de computação de reservatório treinada, conseguiu prever 95% do comportamento das variáveis caóticas com até 3 tempos de Lyapunov. Nas previsões das dessincronizações do sistema de osciladores caóticos acoplados, conseguiu prever acima de 62,9% do comportamento do sinal para o intervalo de previsão menor ou igual a 6 tempos de Lyapunov. Em relação aos eventos extremos, a técnica utilizada alcançou uma taxa de acerto de 71% e 85% para a previsão de 6 e 3 tempos de Lyapunov, respectivamente.

Palavras-chave: eventos extremos; computação de reservatório; aprendizagem de máquina; circuitos caóticos sincronizados; otimização de parâmetros.

## ABSTRACT

Several natural phenomena are nonlinear, implying that doubling some of their variables generally will not increase or decrease the effect twice; instead, it may cause an irregular scaling of the effect. In some cases, the emergence of extreme events is observed, reaching large amplitudes, occurring at aperiodic intervals, and generally lacking a well-defined cause. Examples include earthquakes, floods, stock market crashes, power supply blackouts, instabilities in laser cavities, epilepsy crises, etc. Naturally, due to their adverse economic and social impact, there is great interest in predicting, measuring, and, when possible, mitigating their occurrence. A special type of extreme event is the 'Dragon-kings,' as they can be observed when chaotic oscillators of the Gauthier-Bienfang type are strongly coupled, allowing their study in a laboratory environment. In this research, the goal was to obtain predictions of chaotic series, desynchronizations, and extreme events from data produced by synchronized chaotic oscillators, especially using a machine learning method known as reservoir computing, which is a type of recurrent neural network trained only in its output layer. A numerical simulation of two Gauthier-Bienfang-type chaotic electronic oscillator circuits generated the data used for training and subsequent testing of the reservoir computing network's prediction capability, with its global parameters, the 'hyperparameters,' optimized. The trained reservoir computing network was able to predict up to 95% of the behavior of chaotic variables for the prediction interval of up to 3 Lyapunov times. In predicting desynchronizations in the coupled chaotic oscillator system, it achieved a prediction accuracy of over 62.9% for the prediction interval less than or equal to 6 Lyapunov times. Regarding extreme events, the technique used achieved an accuracy rate of 71% and 85% for predicting 6 and 3 Lyapunov time, respectively.

Keywords: extreme events; reservoir computing; machine learning; synchronized chaotic circuits; parameter optimization.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Função de densidade de probabilidades produzido a partir do acoplamento unidirecional tipo diretor-resposta (DR) de circuitos eletrônicos de Gauthier-Bienfang, onde se observam o 'fat-tail' . . . . .	21
Figura 2 – Atrator estranho do sistema de Lorenz. Figura característica obtida quando os sinais do sistema dinâmico caótico são plotados no plano tridimensional. . . . .	22
Figura 3 – Evolução de duas trajetórias partindo de condições iniciais muito próximas. A função $\delta(t)$ representa a distância ponto a ponto entre as trajetórias. . . . .	24
Figura 4 – Oscilador de Gauthier-Bienfang . . . . .	25
Figura 5 – Diagrama de fase para um oscilador de Gauthier-Bienfang, (a) regime periódico, (b) regime caótico . . . . .	26
Figura 6 – Sistema DR onde dois osciladores caóticos de Gauthier-Bienfang são acoplados de forma unidirecional. Quando os dois circuitos sincronizam, o circuito resposta (b) deve apresentar a mesma dinâmica do circuito diretor (a). . . . .	27
Figura 7 – Dessincronizações, $\ \mathbf{x}_\perp\  \neq 0$ , observadas quando o sistema DR é acoplado unidirecionalmente em $\mathbf{V}_2$ . . . . .	27
Figura 8 – Na aprendizagem supervisionada é necessário utilizar um sinal "professor", que será a referência para corrigir a resposta real. . . . .	29
Figura 9 – Exemplo de grafo com representação de nós (círculos de diversos tamanhos), arestas (retas interligando os nós), autoloops (seta que começa e termina no mesmo nó, ao centro e à esquerda da figura), forças de conexão (uma forma de mostrar a força de conexão entre os nós é variar a espessura da aresta que os interliga). . . . .	31
Figura 10 – Importância de escolher a topologia correta para o objetivo. A rede completamente conectada (a) sincroniza melhor que o Grafo Regular $2k$ (b) e a topologia Small-World (c), em contrapartida, a topologia Erdős-Renyi (d) é mais robusta a perturbações. . . . .	31
Figura 11 – O neurônio artificial e o biológico. Note que para o neurônio artificial replicar o comportamento do biológico é necessário aplicar uma soma ponderada em sua entrada, para então "ativar" o neurônio artificial executando a função sigmóide nesta soma ponderada. . . . .	32

Figura 12 – Tipos de RNA: As conexões em azul são alteradas durante o processo de aprendizagem enquanto as conexões em preto são fixas. (a) Redes neurais feedforward ou redes neurais de alimentação direta em português. (b) Redes neurais recorrentes (RNR) onde as setas indicam estruturas de autoloop e feedback. (c) Redes de computação de reservatório, cujo reservatório é fixo e apenas a camada de saída (leitura) é ajustada durante o processo de aprendizagem. . . . .	33
Figura 13 – Arquitetura de uma RCR composta de três camadas: entrada, reservatório e saída. $\mathbf{u}(t)$ corresponde ao sinal de entrada, $\mathbf{W}_{in}$ a matriz que conecta os neurônios da camada de entrada ao reservatório, $\mathbf{x}(t)$ aos vetores de estado interno do reservatório e $\mathbf{W}_{out}$ a matriz que conecta o reservatório a camada de saída. $\hat{\mathbf{Y}}(t)$ corresponde ao sinal que no processo de aprendizagem é comparado ao sinal esperado $\mathbf{Y}_{alvo}$ , produzindo o sinal de erro $\mathbf{E}(\hat{y}, y^{alvo})$ . . . . .	37
Figura 14 – Atrator estranho do circuito GB-a obtido na simulação numérica . . .	44
Figura 15 – Fluxograma dos passos que serão executados pelo algoritmo de otimização dos HP. Partindo dos testes realizados com a RCR-HP01, o caso médio e o pior caso são selecionados para o teste 01, otimizações e busca de HP02, que consiste em obter e testar 500 RCR construídas com 500 combinações de HP diferentes, variando todos os HP ao mesmo tempo, em cada combinação. Partindo do HP02 obtido, o teste 02 irá produzir 3500 RCR diferentes, gerando 500 combinações por HP, variando um HP por vez e mantendo o melhor resultado para o próximo HP. . . . .	48
Figura 16 – Diagramas de caixa ( <i>boxplot</i> ) dos resultados das previsões iniciais obtidas com a RCR produzida com HP01. Na cor vermelha, <i>online com loopback</i> (ON_CL), na cor verde, <i>online sem loopback</i> (ON_SL), na cor laranja, <i>offline com loopback</i> (OF_CL) e na cor azul, <i>offline sem loopback</i> (OF_SL). Em (a) estão destacados os resultados para os testes com <i>loopback</i> , cores vermelha e laranja, observe que estes resultados são todos negativos. Em (b) estão ampliados os resultados para os testes sem <i>loopback</i> , cores verde e azul, note a similaridade <i>online</i> e <i>offline</i> . . . . .	52
Figura 17 – Diagramas de caixa dos resultados das previsões iniciais obtidas pela RCR produzida com HP01. Nesta figura são apresentados somente os modos <i>offline sem loopback</i> OF_SL (cor laranja) e <i>online sem loopback</i> ON_SL (cor azul). Observe a grande similaridade nos resultados para os dois modos. . . . .	53

Figura 18 – Valores de $R^2$ para as previsões da RCR-HP01 ao variar TL no modo <i>offline</i> sem <i>loopback</i> para as diversas combinações de variáveis. . .	54
Figura 19 – Valores de NRMSE para as previsões da RCR-HP01 ao variar TL no modo <i>offline</i> . . . . .	55
Figura 20 – Comparação entre $V_{1d}$ , traço na cor azul, e $V_{2d}$ , traço na cor laranja, do circuito diretor do sistema DR. Em (a) estão os sinais completos de toda a faixa de teste de previsão para as duas tensões. Em (b) está uma ampliação dos dois sinais para melhor visualização, note a diferença de amplitude entre as tensões. A hipótese é que $V_{2d}$ possui amplitudes maiores, as quais a RCR produzida com HP01 não conseguiu aprender. . . . .	56
Figura 21 – Comparação entre as previsões obtidas para $V_{1d}$ com (a) RCR-HP01 e (b) RCR-HP03 otimizada em 10TL. Note que em (b), a RCR-HP03 produz uma previsão, traço na cor vermelha, que está mais próxima da saída correta, traço na cor azul, porém ainda apresenta divergências.	57
Figura 22 – Comparação dos valores de $R^2$ obtidos nas previsões da série caótica de $V_{1d}$ utilizando a RCR-HP01, traço vermelho, a RCR-HP03 otimizada no intervalo de 10TL, traço na cor azul e a RCR-HP03 otimizada no intervalo de 20TL, traço na cor verde. Comparando o traço vermelho com o traço azul, observe que a RCR-HP03, que foi obtida em uma otimização no IP de 10TL, melhorou os valores de $R^2$ em todos os outros IP. Já a RCR-HP03 de 20TL, que não convergiu, apresentou resultados piores que a RCR-HP01 na maioria dos IP. .	58
Figura 23 – Amostras das previsões de $V_{2d}$ no IP de 3TL utilizando (a) RCR-HP01 sem ajuste de HP, neste caso, a comparação entre saída obtida, traço na cor vermelha e saída correta, traço na cor azul, gerou um $R^2 = 56\%$ , (b) RCR-HP02, com HP N, Lr, Sr e Iss reduzidos após ajuste, a mesma comparação produziu $R^2 = 92\%$ e (c) RCR-HP03, que reduziu rc_con aumentando a dispersão do reservatório, a mesma comparação produziu $R^2 = 95\%$ . . . . .	59
Figura 24 – Amostras das previsões de $V_{2d}$ no IP de 20TL utilizando (a) RCR-HP01, com $R^2 = 5\%$ e previsão apresentando pouca relação com a saída correta, (b) RCR-HP02, com $R^2 = 23\%$ e um $NRMSE = 79\%$ e (c) RCR-HP03, com $R^2 = 18\%$ e um $NRMSE = 82\%$ . Tanto com HP02 como com HP03 a RCR ainda não produziu a amplitude e frequência corretas. A hipótese de que um aumento no processamento, com mais neurônios e no número de testes de otimização devem permitir a convergência da RCR. . . . .	61

Figura 25 – Variação de $R^2$ em função da proporção utilizada para treino e teste no conjunto de $10^5$ pontos da variável $V_{2d}$ . Observe que a redução no conjunto de treino implica em uma pequena piora no $R^2$ , que na menor proporção, 10/90, permanece acima de 91%. . . . .	62
Figura 26 – Variação de $RMSE$ em função da proporção utilizada para treino e teste em $10^5$ pontos da variável $V_{2d}$ . Note que ocorre um pequeno aumento do erro, porém permanece abaixo de 0,3 na menor proporção, 10/90. . . . .	62
Figura 27 – Comparação da flutuação do erro $R^2$ na previsão de $V_{2d}$ , obtido com (a) RCR-HP01, traço na cor vermelha, (b) RCR-HP03 otimizada em 3TL, traço na cor azul, e (c) RCR-HP03 otimizada em 20TL, traço na cor verde. Observe que para $IP < 15TL$ a RCR-HP03 de 3TL apresenta os melhores resultados. . . . .	63
Figura 28 – Diagramas de caixa comparando a flutuação de $R^2$ na previsão das dessincronizações, que ocorrem quando $\ \mathbf{X}_\perp\  > 0$ no sistema DR acoplado em $V_2$ . Como entrada da RCR foram utilizadas as variáveis, ou combinações de variáveis do sistema DR. Foram avaliadas as condições <i>online</i> com <i>loopback</i> - ON_CL na cor verde, <i>online</i> sem <i>loopback</i> - ON_SL na cor azul, <i>offline</i> com <i>loopback</i> - OF_CL na cor vermelha e <i>offline</i> sem <i>loopback</i> - OF_SL na cor laranja. . . . .	64
Figura 29 – Resultados das previsões de $\ \mathbf{X}_\perp\ $ em termos de variação de $R^2$ , tendo na entrada da RCR-HP01 as variáveis do sistema DR, excluindo $\ \mathbf{X}_\perp\ $ e "todas". Na cor azul estão os resultados dos testes <i>online</i> sem <i>loopback</i> - ON_SL e na cor laranja estão os resultados dos testes <i>offline</i> sem <i>loopback</i> - OF_SL. Note que o melhor resultado é observado com a combinação $V_{1d}V_{2d}$ no centro da figura. . . . .	65
Figura 30 – Resultado das previsões de $\ \mathbf{X}_\perp\ $ obtidas com a RCR-HP01 em função do intervalo de previsão em TL. Na entrada da RCR foram aplicadas as variáveis do sistema DR e $\ \mathbf{X}_\perp\ $ . Observe que $\ \mathbf{X}_\perp\ $ , traço na cor amarela, na entrada da RCR produz o melhor resultado, enquanto a combinação "todas", traço na cor cinza, não alcança valores mínimos de convergência. . . . .	66
Figura 31 – Resultados das previsões de $\ \mathbf{X}_\perp\ $ em termos de variação de $R^2$ , tendo na entrada da RCR-HP01 as variáveis do sistema DR, excluindo $\ \mathbf{X}_\perp\ $ e "todas". Esta figura detalha os dados contidos nos diagramas de caixa da Figura 29. . . . .	67
Figura 32 – Comparação entre as previsões de $\ \mathbf{X}_\perp\ $ obtidas com a RCR-HP01, RCR-HP03 e RCR-HP04 em vários IP. Observe que os três HP foram otimizados no IP de 6TL. . . . .	68

- Figura 33 – Comparação entre as previsões de  $\|X_{\perp}\|$  obtidas com a RCR-HP01 e RCR-HP03, tendo como variável na entrada da RCR a combinação de variáveis  $V_{1d\_}V_{2d}$ . Observe que as otimizações de HP01 e HP03 foram realizadas no IP 5TL. Note também a melhoria significativa obtida com a RCR-HP03, traço na cor azul. . . . . 69
- Figura 34 – Gráfico da função de densidade de probabilidade produzida a partir dos dados simulados do sistema DR acoplado em  $V_2$ . O traço vertical na cor vermelha indica a amplitude de 1,5 unidades arbitrárias, a partir da qual o evento é considerado extremo. Observe que após o tracejado vertical, a função apresenta o comportamento característico dos *Dragon-King*, uma cauda "gorda"ou *fat tail*. . . . . 69
- Figura 35 – Previsão do  $\|X_{\perp}\|$  (na cor vermelha) do sistema DR acoplado em  $V_2$ , obtido pela RCR construída com o HP03 otimizado no IP de 6TL, treinada com 17000 pontos e testada em 11000 pontos. O traço horizontal em (a) e (b) na cor vermelha tracejado, representa o limite de 1,5 u.a. onde uma dessincronização será considerada evento extremo. O evento detectado pela RCR é o ponto na cor preta sobre o traço vermelho em (b) e o evento real é o ponto na cor verde sobre o traço verde em (b), a linha azul representa a saída correta esperada na previsão, neste caso, 6TL antes de ocorrer o evento real. Na primeira detecção, evento com um traço vertical na cor preta, a RCR antecipou o evento real em 1,1TL. . . . . 70

## LISTA DE TABELAS

Tabela 1 – Valores aplicáveis aos hiperparâmetros da RCR . . . . .	38
Tabela 2 – Fatores que influenciam na qualidade das previsões da RCR . . . . .	43
Tabela 3 – Valores utilizados para simular numericamente o acoplamento do sistema DR em $V_2$ . . . . .	44
Tabela 4 – Hiperparâmetros HP01 para gerar a RCR inicial . . . . .	45
Tabela 5 – Comparação entre HP01, HP02 e HP03 obtidos na otimização onde a entrada da RCR é a variável $V_{1d}$ e a saída é treinada para antecipar a variável $V_{1d}$ em um IP de 10TL . . . . .	57
Tabela 6 – Comparação entre HP01, HP02 e HP03 obtidos na otimização onde a entrada da RCR é a variável $V_{1d}$ e a saída é treinada para antecipar a variável $V_{1d}$ em um IP de 20TL . . . . .	58
Tabela 7 – Comparação entre HP01, HP02 e HP03 obtidos na otimização onde a entrada da RCR é a variável $V_{2d}$ e a saída é treinada para antecipar a variável $V_{2d}$ em um IP de 3TL . . . . .	59
Tabela 8 – Comparação entre HP01, HP02 e HP03 obtidos na otimização onde a entrada da RCR é a variável $V_{2d}$ e a saída é treinada para antecipar a variável $V_{2d}$ em um IP de 20TL . . . . .	60
Tabela 9 – Comparação entre HP01, HP02 e HP03 obtidos na otimização onde a entrada da RCR é a combinação $V_{1d}V_{2d}$ e a saída é treinada para antecipar a variável $\ X \perp\ $ em um IP de 5TL . . . . .	65
Tabela 10 – Comparação entre HP01, HP02, HP03 e HP04 obtidos na otimização onde a entrada da RCR é a variável $\ X \perp\ $ e a saída é treinada para antecipar a variável $\ X \perp\ $ em um IP de 6TL . . . . .	67
Tabela 11 – Comparação entre as previsões de eventos extremos obtidas nos IP 1, 3, e 6TL com as RCR-HP03 e RCR-HP04 . . . . .	71

## LISTA DE EQUAÇÕES

Equação 2.1 .....	23
Equação 2.2 .....	23
Equação 2.3 .....	23
Equação 2.4 .....	24
Equação 2.5 .....	25
Equação 2.6 .....	25
Equação 2.7 .....	25
Equação 2.8 .....	25
Equação 2.9 .....	25
Equação 2.10 .....	25
Equação 2.11 .....	25
Equação 2.12 .....	26
Equação 2.13 .....	26
Equação 2.14 .....	26
Equação 2.15 .....	26
Equação 2.16 .....	26
Equação 2.17 .....	27
Equação 2.18 .....	27
Equação 2.19 .....	28
Equação 2.20 .....	28
Equação 2.21 .....	29
Equação 2.22 .....	29
Equação 3.1 .....	31
Equação 3.2 .....	31
Equação 3.3 .....	31
Equação 3.4 .....	31
Equação 3.5 .....	32
Equação 3.6 .....	33
Equação 3.7 .....	33

Equação 3.8 .....	34
Equação 3.9 .....	34
Equação 3.10 .....	35
Equação 3.11 .....	35
Equação 3.12 .....	36
Equação 3.13 .....	36
Equação 3.14 .....	37
Equação 3.15 .....	37
Equação 3.16 .....	37
Equação 3.17 .....	37
Equação 3.18 .....	39
Equação 3.19 .....	39
Equação 3.20 .....	40
Equação 3.21 .....	40
Equação 3.22 .....	40
Equação 3.23 .....	40
Equação 3.24 .....	40
Equação 3.25 .....	40
Equação 3.26 .....	40
Equação 3.27 .....	40
Equação 3.28 .....	41
Equação 3.29 .....	42
Equação 3.30 .....	42
Equação 3.31 .....	43
Equação 3.32 .....	43
Equação 3.33 .....	43
Equação 3.34 .....	43
Equação 4.1 .....	50
Equação 4.2 .....	50
Equação 4.3 .....	50
Equação 4.4 .....	50

Equação 4.5 .....	53
Equação 4.6 .....	57
Equação 4.7 .....	58
Equação 4.8 .....	66

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>17</b>
<b>2</b>	<b>EVENTOS EXTREMOS E SINCRONIZAÇÃO DE SISTEMAS CAÓTICOS</b>	<b>19</b>
2.1	EVENTOS EXTREMOS	19
2.2	SISTEMAS CAÓTICOS	21
2.3	SINCRONIZAÇÃO DE SISTEMAS CAÓTICOS	22
2.4	SISTEMA DE GAUTHIER-BIENFANG COMO FONTE DE EVENTOS EXTREMOS	25
<b>3</b>	<b>APRENDIZAGEM DE MÁQUINA E COMPUTAÇÃO DE RESERVATÓRIO</b>	<b>28</b>
3.1	APRENDIZAGEM DE MÁQUINA	28
3.2	FORMAS DE APRENDIZAGEM DE MÁQUINA	29
3.3	REDES NEURAIS ARTIFICIAIS	30
3.3.1	Redes	30
3.3.2	Redes Neurais Artificiais	31
3.4	REDES DE COMPUTAÇÃO DE RESERVATÓRIO	34
3.4.1	Hiperparâmetros de uma RCR	38
3.4.2	Otimização dos Hiperparâmetros da RCR	39
3.4.3	Treinamento da RCR	39
3.4.4	Métricas de qualidade das previsões	40
<b>4</b>	<b>PROPOSTA DE USO DA REDE DE COMPUTAÇÃO DE RESERVATÓRIO NA PREVISÃO DE SÉRIES CAÓTICAS E DESSINCRONIZAÇÕES</b>	<b>42</b>
4.1	SIMULAÇÃO NUMÉRICA DAS VARIÁVEIS DO CIRCUITO GB	43
4.2	OTIMIZAÇÃO DOS HIPERPARÂMETROS DA RCR	44
4.2.1	Hiperparâmetros para Previsão de Séries Caóticas	45
4.2.2	Hiperparâmetros para Previsão das Dessincronizações	48
4.2.3	Previsões dos Eventos Extremos do Sistema DR Acoplado em $V_2$	49
<b>5</b>	<b>RESULTADOS E DISCUSSÃO</b>	<b>51</b>
5.1	PREVISÕES DAS SÉRIES CAÓTICAS	51
5.1.1	Resultados para $V_{1d}$	53
5.1.2	Resultados para $V_{2d}$	56
5.2	PREVISÕES DA VARIÁVEL $\ X \perp\ $ DO SISTEMA DR QUANDO ACOPLADO EM $V_2$	62
5.3	PREVISÕES DOS EVENTOS EXTREMOS DO SISTEMA DR	68
<b>6</b>	<b>CONCLUSÕES E PERSPECTIVAS</b>	<b>73</b>
	<b>REFERÊNCIAS</b>	<b>75</b>

## 1 INTRODUÇÃO

Os fenômenos da natureza são intrinsecamente não lineares (Strogatz, 2018), alguns desses fenômenos apresentam uma dinâmica caótica e, mesmo conhecendo o modelo matemático, não é possível prever tais sistemas a longo prazo. Eventos extremos podem exibir comportamento caótico ou aleatoriedade. Importante notar que alguns eventos apresentam alguma correlação interna (Sornette, 2009), assim antes de eventos de grande impacto, espera-se observar sinais ou eventos menores que apontam a ocorrência de uma catástrofe.

Os eventos extremos são ocorrências de grande amplitude, de difícil previsão, que podem causar impacto social e econômico (Albeverio, Jentsch e Kantz, 2006). Além disso, são observados em diversos sistemas de áreas distintas, alguns exemplos são as perturbações em sistemas laser (Kovalsky Marcelo G.; Hnilo, 2011), as instabilidades em redes de fornecimento de energia (Trakas Dimitris N.; Panteli, 2018), a epilepsia (Arcangelis, 2012), os terremotos, as chuvas torrenciais, os ataques terroristas, as flutuações agressivas em mercados financeiros (Chowdhury *et al.*, 2022; Pyragas e Pyragas, 2020) entre outros.

A busca por modelos matemáticos e experimentais que permitam prever, gerenciar e, quando possível, mitigar a ocorrência de tais eventos, atrai pesquisas científicas, que avançam em duas vertentes, a estocástica, cujo enfoque é identificar as probabilidades de ocorrência dos eventos (Farazmand e Sapsis, 2019; Pyragas e Pyragas, 2020) ou através da abordagem por aprendizagem de máquina, cujo enfoque é obter modelos, geralmente por regressão linear, que permitam previsões diretamente da análise dos dados, obtidos a partir do fenômeno físico (Chowdhury *et al.*, 2022; Cucchi *et al.*, 2022; Racca e Magri, 2022).

A imprevisibilidade dos sistemas caóticos compartilha similaridades com a irregularidade dos eventos extremos. De fato, Gauthier e Bienfang, observaram em um estudo de 1996 (Gauthier e Bienfang, 1996), que circuitos eletrônicos caóticos sincronizados, dessincronizavam de forma abrupta e breve, independente da força de acoplamento escolhida. Em 2013, foi demonstrado que tal comportamento tem características de eventos extremos do tipo *Dragon-King* (Sornette, 2009; Cavalcante *et al.*, 2013).

Utilizando a abordagem por aprendizagem de máquina, estudos apontaram que as redes de computação de reservatório (RCR), uma ramificação de redes neurais recorrentes (Haykin, 2001), têm resultados satisfatórios para previsão de séries caóticas (Jaeger, 2001; Farazmand e Sapsis, 2019; Nakajima, 2021) e antecipação de eventos extremos (Pyragas e Pyragas, 2020; Racca e Magri, 2022).

Nesta dissertação, o objetivo principal é prever as séries caóticas, as dessincronizações e os eventos extremos observados nos circuitos eletrônicos de Gauthier

e Bienfang (Gauthier e Bienfang, 1996) acoplados, aqui designados sistema Diretor-Resposta (DR), por meio da RCR. Como a RCR tem forte dependência de parâmetros globais ou hiperparâmetros (HP), que são as regras para construí-la e conectá-la. Para alcançar o objetivo principal, será necessário aplicar métodos de otimização, visando encontrar os HP que maximizam a eficiência da RCR.

A presente dissertação está estruturada da seguinte forma: no capítulo 2, são apresentados os fundamentos de eventos extremos e sincronização de sistema caóticos; no capítulo 3, são explorados os conceitos de aprendizagem de máquina e redes de computação de reservatório; no capítulo 4, são apresentados os métodos de otimização da RCR e as técnicas de previsão aplicadas; no capítulo 5 são discutidos os resultados obtidos; e no capítulo 6 são apresentadas as conclusões e perspectivas.

## 2 EVENTOS EXTREMOS E SINCRONIZAÇÃO DE SISTEMAS CAÓTICOS

Neste capítulo, serão abordados os fundamentos sobre eventos extremos e sincronização de sistemas caóticos, com enfoque para o circuito eletrônico proposto por Gauthier-Bienfang (Gauthier e Bienfang, 1996), cujo sinal caótico será utilizado para treinar e testar a rede de computação de reservatório (RCR) desenvolvida ao longo desta dissertação.

### 2.1 EVENTOS EXTREMOS

Os eventos extremos, em geral, tem consequências adversas negativas e ocorrem em sistemas dinâmicos complexos que envolvem múltiplas áreas de pesquisa. Para se ter uma percepção do impacto dos eventos extremos, o Tsunami que atingiu a costa da Ásia em dezembro de 2004, matou 200.000 pessoas (Albeverio, Jentsch e Kantz, 2006); a crise norte-americana do *subprime*, de 2008, causou uma perda de 3 trilhões de dólares no mundo em apenas 2 semanas (Crise [...], 2008); mais recente, a epidemia da COVID-19, entre 2020 e 2023, causou aproximadamente 6,9 milhões de mortes (WHO, 2020) no mundo. Um estudo da Organização Meteorológica Mundial aponta que 56% das mortes e 75% das perdas financeiras ocorridas entre 1970 e 2019 (WMO, 2020) estão relacionadas com eventos extremos climáticos.

Os extremos também trazem desafios para o desenvolvimento de tecnologias como o laser (Kovalsky Marcelo G.; Hnilo, 2011); redes de fornecimento de energia (Trakas Dimitris N.; Panteli, 2018) e para o entendimento de doenças do cérebro, como a epilepsia (Arcangelis, 2012; Chowdhury *et al.*, 2022).

A literatura aponta que ainda não há consenso sobre uma definição geral para eventos extremos (Nakajima, 2021; Albeverio, Jentsch e Kantz, 2006). Entretanto, algumas propriedades comuns permitem identificá-los:

1. Ocorrem em sistemas dinâmicos como manifestação de suas complexas relações, podendo ter origens internas ou externas (Sornette, 2007);
2. Os eventos extremos são aqueles de maior magnitude na evolução temporal de um observável (Chowdhury *et al.*, 2022); e
3. Uma característica importante é sua **ocorrência irregular**, esta assinatura também pode ser observada na evolução caótica das trajetórias dos sistemas dinâmicos não lineares acoplados (Chowdhury *et al.*, 2022).

Sobre a propriedade 1, além de aspectos internos, estímulos externos, tais como ruídos ou interferências, podem desencadear comportamentos fora do equilíbrio, que causam grandes desvios na dinâmica do sistema.

Uma vez que o extremo de um terremoto (Boore, 1989) é diferente do extremo de um índice no mercado financeiro (Albeverio, Jentsch e Kantz, 2006), não há uma regra geral sobre a quantificação associada à expressão 'maior magnitude' da propriedade 2. Portanto, o extremo de um observável deve ser particularizado na amostra temporal do fenômeno em análise. Especificamente, deve-se estabelecer uma referência para o conjunto de eventos extremos.

A propriedade 3 é fundamental e permitirá investigar propriedades dos eventos extremos. A combinação da propriedade 3 com a observação do fenômeno de sincronização caótica tornou-se fundamental para a compreensão dos osciladores caóticos acoplados, pois eventos extremos são observados nestes sistemas, quando grandes desvios de sincronização ocorrem.

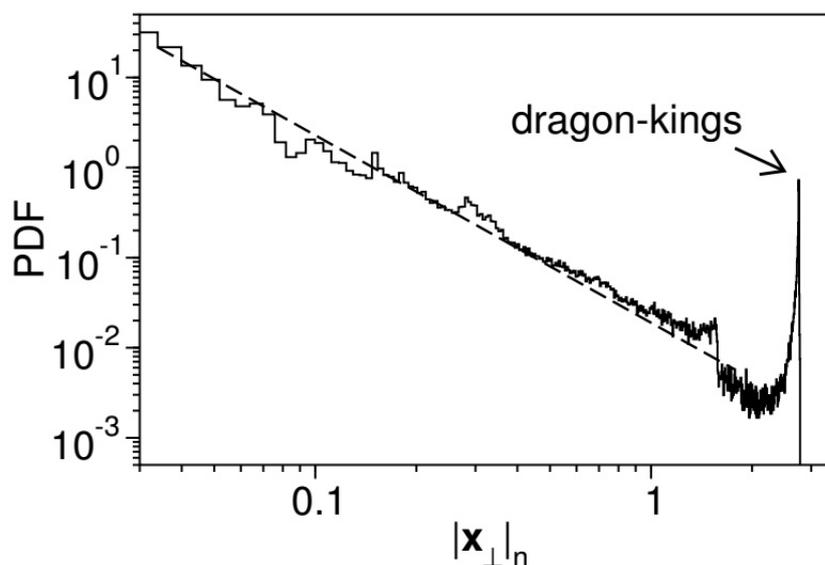
Outro aspecto importante para caracterizar eventos extremos são as funções de distribuições de probabilidade. Alturas e pesos de pessoas são variáveis com distribuições normais (Gaussianas) suaves, onde é possível se identificar valores típicos e pequenas variações ao redor deles. Já a distribuição de riquezas segue a lei de Pareto, uma distribuição por lei de potência, uma vez que é possível existir uma parcela da população duas, dez, até cem vezes mais ricas que outra parcela (Sornette, 2007).

Eventos extremos do tipo *dragon-king* são observados nos valores de menores probabilidades (cauda) das distribuições de probabilidades que seguem uma lei de potência (Sornette, 2009; Albeverio, Jentsch e Kantz, 2006), em especial, apresentam a peculiaridade de uma cauda "gorda", do inglês *fat-tail*. Além disso, as leis de potência caem muito mais lentamente que distribuições exponenciais (Gaussianas), indicando uma maior probabilidade de ocorrência (Albeverio, Jentsch e Kantz, 2006). Na figura 1 está a distribuição de densidade de probabilidades observada no sistema DR, quando acoplado em  $V_2$  (Cavalcante *et al.*, 2013). Em especial, o estudo de Sornette (Sornette, 2009) aponta que os eventos extremos do tipo *dragon-king* ocorrem em vários setores onde haja forte correlação ou sincronização entre as variáveis. Sugere também que tais eventos possuem mecanismos específicos de formação, como transições de fase, bifurcação, catástrofe.

Tendo o conceito de eventos extremos, torna-se importante compreender quais tipos de mecanismos produzem estes eventos, os principais são a crise induzida interna, intermitência, intermitência de Pomeau-Manneville, salto de atratores em sistemas multiestáveis devido a ruído interno ou externo. Além destes, deve-se incluir nesta lista os osciladores acoplados, pois podem apresentar intermitência do tipo on-off, intermitência do tipo in-out, sincronização de fase imperfeita entre outros (Chowdhury *et al.*, 2022).

O estudo de eventos extremos pode ser dividido em 4 partes: i) Mecanismos; ii)

Figura 1 – Função de densidade de probabilidades produzido a partir do acoplamento unidirecional tipo diretor-resposta (DR) de circuitos eletrônicos de Gauthier-Bienfang, onde se observam o 'fat-tail'



Fonte: Adaptado de (Cavalcante *et al.*, 2013)

Previsão; iii) Mitigação; iv) Estatística (Farazmand e Sapsis, 2019). O enfoque neste trabalho é a previsão e, quando possível, a mitigação.

As previsões podem ser estudadas através das estatísticas dos extremos, que permitem avaliar as probabilidades de ocorrência de tais eventos. Alternativamente, o uso de técnicas baseadas em dados, em especial, o uso de modelos de redes neurais artificiais apresentaram bons resultados (Racca e Magri, 2022). Neste sentido, destaca-se o paradigma da computação de reservatório, com aplicações na previsão da dinâmica não-linear de sistemas caóticos e potencial aplicação na previsão de eventos extremos (Jaeger, 2001; Cavalcante *et al.*, 2013; Chowdhury *et al.*, 2022; Ferreira, 2011; Cucchi *et al.*, 2022).

Nos próximos tópicos serão explorados os conceitos de caos e sistemas caóticos, tendo como enfoque o circuito eletrônico de Gauthier-Bienfang (Gauthier e Bienfang, 1996).

## 2.2 SISTEMAS CAÓTICOS

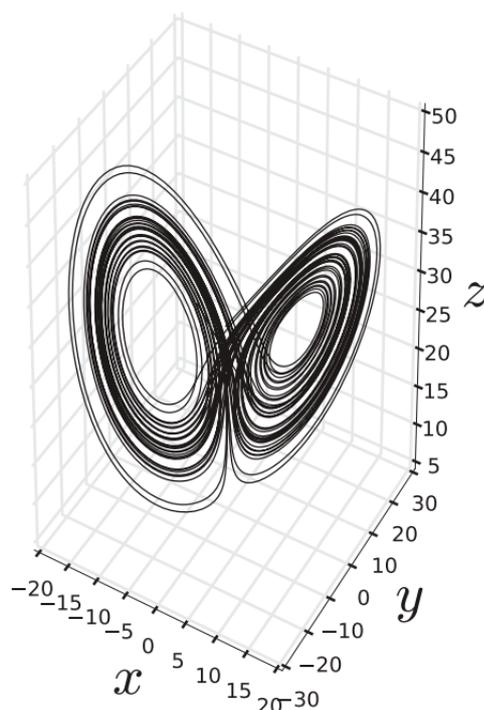
Poincaré, no séc. XIX, foi o pioneiro ao desenvolver um poderoso método geométrico para analisar problemas não-lineares. Além disso, foi o primeiro a notar a possibilidade de sistemas dinâmicos exibirem caos, cuja definição ainda não é um conceito fechado, entretanto em termos práticos a seguinte definição operacional servirá aos propósitos deste trabalho:

**Definição 1 (Caos)** *Caos é um comportamento aperiódico de longo prazo em um sistema determinístico que exhibe dependência sensível às condições iniciais (Strogatz,*

2018).

Assim, para que um sistema dinâmico tenha comportamento caótico é necessário possuir dimensão maior que 2, ser não linear e exibir forte dependência das condições iniciais. Além disso, se um sistema dinâmico apresentar caos, seu comportamento no espaço de fases, deve produzir um gráfico similar ao atrator estranho da Figura 2 que é a assinatura dos sistemas caóticos. Complementar a esta análise, também pode-se avaliar se existe uma divergência entre trajetórias no espaço de fases, determinando seu expoente de Lyapunov, que mede o quão rápido tais trajetórias se afastam (Strogatz, 2018). Por apresentarem um expoente de Lyapunov global positivo, uma previsão de longo prazo se torna praticamente impossível onde uma pequena incerteza é amplificada rapidamente (Strogatz, 2018).

Figura 2 – Atrator estranho do sistema de Lorenz. Figura característica obtida quando os sinais do sistema dinâmico caótico são plotados no plano tridimensional.



Fonte: Adaptado de (Eroglu, Lamb e Pereira, 2017)

### 2.3 SINCRONIZAÇÃO DE SISTEMAS CAÓTICOS

Sincronização se refere a tendência de dois ou mais sistemas terem o mesmo comportamento dinâmico e implica em forte correlação entre estes sistemas, quando acoplados. A sincronização é fundamental para a compreensão de diversos fenômenos e tecnologias, tais como o funcionamento dos lasers, o projeto de residências e prédios, os supercondutores, a epilepsia entre outros (Eroglu, Lamb e Pereira, 2017).

Cabe destacar que há tipos de sincronização, com destaque para a sincronização completa (SC), a sincronização em fase (SF) e a sincronização generalizada (SG) (Eroglu, Lamb e Pereira, 2017).

- SC: Ocorre quando dois sistemas seguindo um acoplamento difusivo, ou seja, onde a influência mútua entre eles ocorre de maneira suave e contínua, possuem dinâmicas idênticas e oscilam em uníssono;
- SF: Cada sistema oscila em uma frequência própria, porém para pequenas forças de acoplamento as fases podem ser acopladas, com as amplitudes permanecendo não correlacionadas; e
- SG: Surgem alguns pontos em comum entre os sistemas dinâmicos, onde se observa uma relação de sincronização mais complexa. A sincronização SC é considerada um caso especial da SG (Parlitz *et al.*, 1996).

A pesquisa em sincronização de caos foi impulsionada por estudos como o de Fujisaka e Yamada, que desenvolveram a teoria da estabilidade do movimento sincronizado em sistemas de osciladores acoplados (Fujisaka e Yamada, 1983); outro destaques são os estudos de Pecora e Carrol (Pecora e Carroll, 1990), que trataram da sincronização em sistemas caóticos, e Gauthier e Bienfang (Gauthier e Bienfang, 1996), que abordaram a sincronização intermitente em osciladores caóticos acoplados.

Sincronizar sistemas caóticos nada mais é do que ajustar a operação de dois ou mais sistemas caóticos que oscilam de forma independente, a fim de que passem a apresentar uma dinâmica relacionada (Feitoza, 2022).

O estudo de Gauthier e Bienfang (Gauthier e Bienfang, 1996) sugere que dois sistemas caóticos similares não sincronizam necessariamente. Ao contrário, longos intervalos de sincronização de qualidade são interrompidos aleatoriamente por dessincronizações bruscas de grande amplitude, em relação ao tamanho do atrator. Estas dessincronizações são denominadas de borbulhamento do atrator (*attractor bubbling*) e, em geral, estão associadas com ruídos ou pequenas diferenças de parâmetros entre os sistemas caóticos acoplados (Cavalcante *et al.*, 2013).

A sincronização de sistemas caóticos ocorre através do acoplamento unidirecional ou bidirecional dos sistemas em questão. No acoplamento unidirecional, também denominado diretor-resposta (DR), o sistema diretor controla a dinâmica do sistema resposta. À medida que uma ou, mais variáveis de estado do sistema diretor são passadas ao sistema resposta de forma direta ou indiretamente. No bidirecional, a dinâmica dos sistemas sofrem influência mútua.

Para verificar como isso ocorre, suponha duas trajetórias iniciando seus movimentos nos pontos  $x(t)$  e  $x(t) + \delta(t)$ , em que  $\delta(t)$  é um vetor de comprimento inicial

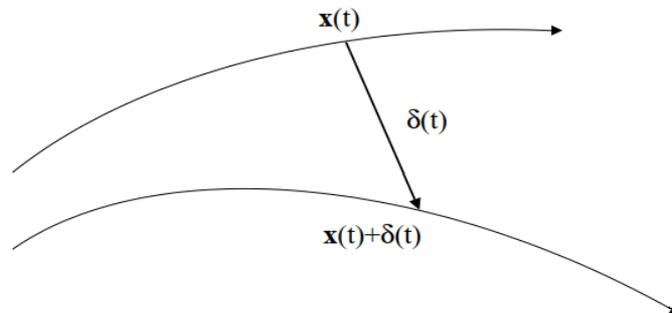
muito pequeno. Na Fig. 3, traz-se uma ideia de como seriam essas trajetórias no espaço de fase. Para verificar se as trajetórias se aproximam ou se afastam, basta analisar o comprimento do vetor  $\delta(t)$ , que é dado por:

$$|\delta(t)| \approx |\delta(0)|e^{\lambda t}, \quad (2.1)$$

em que  $\lambda$  é o expoente de Lyapunov. Da Eq. 2.1, resta claro que o comprimento do vetor  $\delta(t)$  ao longo do tempo depende do expoente de Lyapunov, haja vista que, se  $\lambda < 0$ ,  $\delta(t)$  reduz e, por conseguinte, as trajetórias se aproximam. Por outro lado, se  $\lambda > 0$ ,  $\delta(t)$  aumenta e, naturalmente, as trajetórias se afastam. Logo, em última instância, são os expoentes de Lyapunov que mostram o quão rápido ou lento uma trajetória se afasta ou se aproxima da outra.

Vale destacar que um sistema dinâmico terá tantos expoentes de Lyapunov quantas forem as suas dimensões, ou seja, um sistema com dimensão  $n$  terá  $n$  expoentes de Lyapunov que representam as taxas médias de convergência ou divergência das  $n$  trajetórias independentes no espaço de fase. Observe que, se ao menos um dos expoentes de Lyapunov for positivo, as trajetórias serão divergentes. Logo, basta analisar o maior deles para concluir acerca do comportamento dinâmico das trajetórias (Strogatz, 2018).

Figura 3 – Evolução de duas trajetórias partindo de condições iniciais muito próximas. A função  $\delta(t)$  representa a distância ponto a ponto entre as trajetórias.



Fonte: Adaptado de (Strogatz, 2018)

De um ponto de vista mais geral, a sincronização de dois sistemas caóticos é realizada com o acoplamento *difusivo unilateral*, também chamado de *controle de feedback negativo*, que é obtido adicionando-se um termo de amortecimento ao sistema resposta da seguinte forma:

$$\dot{\mathbf{x}}_d = \mathbf{F}(x_d) \quad e \quad \dot{\mathbf{x}}_r = \mathbf{F}(x_r) + \alpha \mathbf{K}(\mathbf{x}_d - \mathbf{x}_r) \quad (2.2)$$

onde  $\mathbf{x}_d$  ( $\mathbf{x}_r$ ) é o vetor de estados do circuito diretor (resposta),  $\mathbf{F}(x_i)$  com  $i = d, r$ , corresponde à matriz de fluxo dos circuitos diretor e resposta, respectivamente,  $\mathbf{K}$  é

uma matriz que seleciona a combinação linear dos componentes  $\mathbf{x}$  que serão usados na diferença (termo de amortecimento) e  $\alpha$  determina a força de acoplamento.

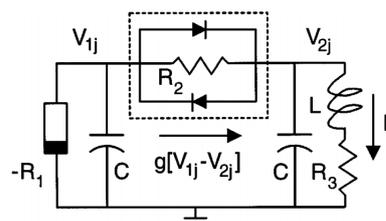
Vale destacar que a força de acoplamento, que é a intensidade da influência de um sistema em outro, matematicamente corresponde ao termo  $\alpha$  na equação 2.2, modifica diretamente a qualidade da sincronização (Pikovsky, Rosenblum e Kurths, 2002; Boccaletti *et al.*, 2002). Isto é, para forças de acoplamento abaixo de uma força de acoplamento crítica, os sistemas interagentes não sincronizam e evoluem de forma independente um do outro. Em contrapartida, para forças de acoplamento acima dessa força crítica, os sistemas sincronizam e passam a apresentar comportamento dinâmico relacionado.

## 2.4 SISTEMA DE GAUTHIER-BIENFANG COMO FONTE DE EVENTOS EXTREMOS

Um desafio no estudo de eventos extremos é a sua produção em laboratório de tal forma que se torne possível analisá-los. Em 1996, Gauthier e Bienfang (Gauthier e Bienfang, 1996) apresentaram um estudo sobre a sincronização de dois circuitos eletrônicos caóticos, cujas dessincronizações aperiódicas, observadas quando os dois circuitos estão acoplados, mais tarde, foram caracterizadas como eventos extremos (Cavalcante *et al.*, 2013). Este sistema será utilizado como fonte de eventos extremos nesta dissertação, pois é uma fonte robusta de caos, simples de simular numericamente. Nesta seção, o sistema proposto por Gauthier será apresentado, com ênfase em sua capacidade de gerar eventos extremos.

Na Fig. 4 está o circuito eletrônico de Gauthier e Bienfang (Gauthier e Bienfang, 1996) (GB), que é um oscilador composto basicamente por um resistor negativo ativo, responsável pela manutenção da energia do circuito, por um elemento não-linear, responsável pela dinâmica complexa, e por um circuito RLC passivo, responsável pelas oscilações (Feitoza, 2022). A dinâmica deste circuito é governada pelas equações adimensionais 2.3, 2.4 e 2.5. A partir de simulações numéricas destas equações, obtém-se o retrato de fase deste oscilador em regime periódico, Fig. 5(a) e caótico Fig. 5(b). Para consulta dos valores utilizados e mais detalhes sobre a dinâmica do circuito, consultar o excelente estudo de (Feitoza, 2022).

Figura 4 – Oscilador de Gauthier-Bienfang



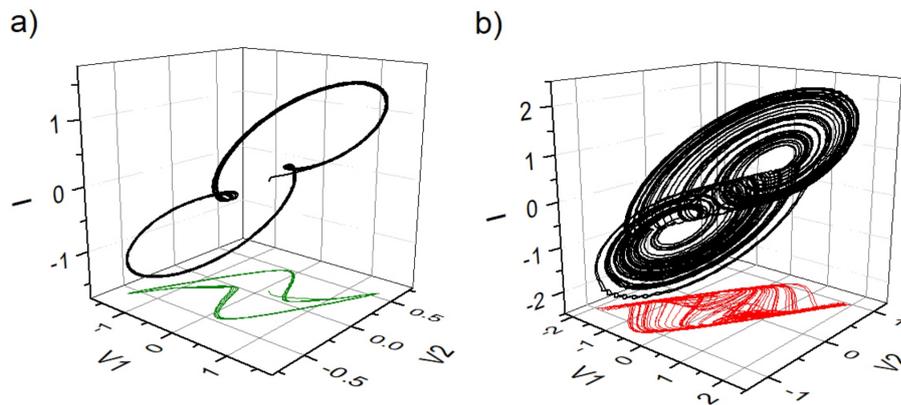
Fonte: Adaptado de (Gauthier e Bienfang, 1996).

$$\dot{V}_{1j} = \frac{V_{1j}}{R_1} - g[V_{1j} - V_{2j}]. \quad (2.3)$$

$$\dot{V}_{2j} = g[V_{1j} - V_{2j}] - I_j, j = m, s. \quad (2.4)$$

$$\dot{I}_j = V_{2j} - R_4 I_j. \quad (2.5)$$

Figura 5 – Diagrama de fase para um oscilador de Gauthier-Bienfang, (a) regime periódico, (b) regime caótico



Fonte: Adaptado de (Feitoza, 2022).

No estudo de Gauthier, dois osciladores como o da Figura 4 são preparados para funcionar na região caótica, em regime de acoplamento difusivo unilateral, aqui denominado sistema Diretor-Resposta (DR), ver Figura 6. A dinâmica do acoplamento é descrita pelas equações diferenciais:

$$\dot{\mathbf{x}}_d = \mathbf{F}[x_d]. \quad (2.6)$$

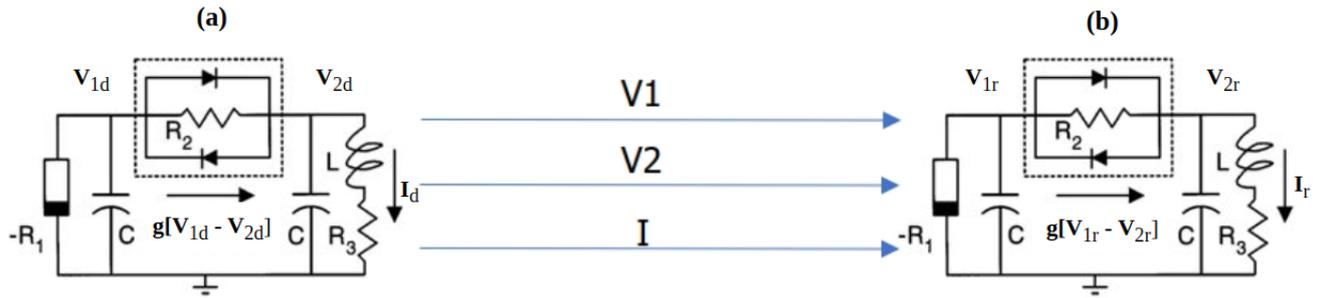
$$\dot{\mathbf{x}}_r = \mathbf{F}[x_r] - c\mathbf{K}(\mathbf{x}_d - \mathbf{x}_r). \quad (2.7)$$

$$\mathbf{F}[x_j] = \begin{pmatrix} \frac{V_{1j}}{R_1} - g[V_{1j} - V_{2j}] \\ g[V_{1j} - V_{2j}] - I_j \\ V_{2j} - R_4 I_j \end{pmatrix}, \text{ com } j = d, r. \quad (2.8)$$

onde  $\mathbf{x}_d$  ( $\mathbf{x}_r$ ) é a posição do oscilador diretor (resposta) no espaço de fase,  $\mathbf{F}$  representa o fluxo dos osciladores,  $\mathbf{K}$  é uma matriz de acoplamento quadrada com dimensões  $n \times n$ ,  $c$  é a força de acoplamento escalar e  $\mathbf{x}_j^T = (V_{1j}, V_{2j}, I_j)$ .

As setas na Figura 6 indicam a possibilidade de sincronizar o circuito resposta (Figura 6(b)) com quaisquer combinações das variáveis  $V_1$ ,  $V_2$  e  $I$  do circuito diretor (Figura 6(a)). Entretanto, quando a sincronização é realizada em  $V_2$ , para qualquer

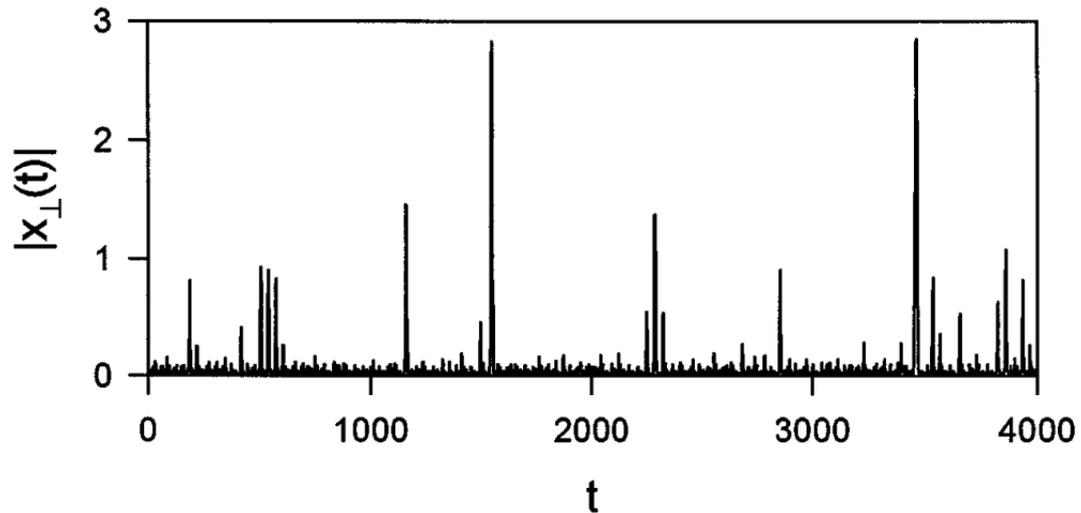
Figura 6 – Sistema DR onde dois osciladores caóticos de Gauthier-Bienfang são acoplados de forma unidirecional. Quando os dois circuitos sincronizam, o circuito resposta (b) deve apresentar a mesma dinâmica do circuito diretor (a).



Fonte: Adaptado de (Gauthier e Bienfang, 1996).

força de acoplamento, o sistema apresenta borbulhamentos ou grandes desvios em sua variedade de sincronização de forma aleatória e breve (Feitoza, 2022). Essas dessincronizações em  $V_2$  podem ser observadas ao se plotar o  $\|\mathbf{x}_\perp\| = \|\mathbf{x}_d - \mathbf{x}_r\|$  ao longo do tempo, conforme Figura 7. Se  $\|\mathbf{x}_\perp\| = 0$ , o sistema estará sincronizado, se  $\|\mathbf{x}_\perp\| \neq 0$ , então se observam as dessincronizações.

Figura 7 – Dessincronizações,  $\|\mathbf{x}_\perp\| \neq 0$ , observadas quando o sistema DR é acoplado unidirecionalmente em  $V_2$ .



Fonte: Adaptado de (Gauthier e Bienfang, 1996).

Quando o sistema dessincroniza, seu estado segue uma órbita que se afasta da variedade de sincronização, mas que eventualmente retorna ao atrator. O estudo de Cavalcante (Cavalcante *et al.*, 2013) demonstrou que estas dessincronizações em  $V_2$  possuem a assinatura de eventos extremos do tipo *Dragon-King* (Sornette, 2009).

### 3 APRENDIZAGEM DE MÁQUINA E COMPUTAÇÃO DE RESERVATÓRIO

Neste capítulo, os conceitos fundamentais da aprendizagem de máquina (AM) serão explorados, delineando suas formas, algoritmos e técnicas. Também serão apresentados os princípios básicos de redes, com enfoque para os conceitos de redes neurais recorrentes (RNR). Em seguida, o funcionamento de redes de computação de reservatório (RCR) será explicado, pois é a principal técnica de AM aplicada neste trabalho.

#### 3.1 APRENDIZAGEM DE MÁQUINA

A aprendizagem de máquina (AM) teve origem na década de 60 como um campo da inteligência artificial, visando identificar padrões a partir de dados (Izbicki e Santos, 2020). Os estudos em AM se concentram no desenvolvimento de algoritmos e técnicas que capacitam computadores a realizar tarefas específicas sem programação explícita, aprendendo e se aprimorando com base nos dados disponíveis. Os algoritmos de AM têm a capacidade de reconhecer padrões, fazer inferências a partir dos dados e tomar decisões ou realizar previsões sobre os sistemas analisados.

Os avanços nesse campo resultaram em tecnologias disruptivas, como o *Deep Blue* da IBM, desenvolvido em 2002. Este computador foi capaz de analisar milhares de partidas de xadrez em segundos, derrotando o campeão humano da época, Garry Kasparov (Campbell, Hoane e Hsu, 2002). Outro feito de AM foi o algoritmo *AlphaGo*, que também superou o 18 vezes campeão humano, Lee Sedol, no jogo Go, considerado mais complexo que o xadrez (Silver *et al.*, 2016). Recentemente, o *ChatGPT* tem se destacado como uma tecnologia capaz de interagir e responder a uma ampla variedade de perguntas de forma natural, simulando uma conversa informal (Liu *et al.*, 2023).

No contexto da AM, surgiram diversas abordagens, como a aprendizagem supervisionada, não supervisionada, por reforço e outras. Além disso, as redes de computação de reservatório (RCR) emergiram em 2001 como uma derivação das redes neurais recorrentes (Jaeger, 2001). Essas RCRs se destacam por sua estrutura única, composta por um reservatório de neurônios interconectados por pesos fixos e uma camada de saída treinável. Esta arquitetura permite à rede manter e processar informações temporais, sendo particularmente útil no processamento de sequências temporais complexas.

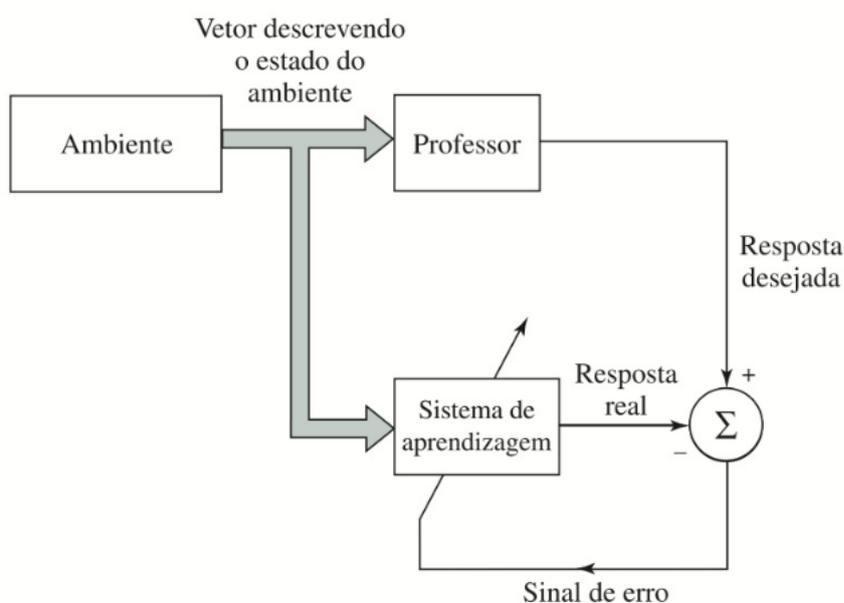
As RCRs oferecem baixo custo de treinamento e rápida convergência no tratamento de séries temporais complexas. Sua capacidade de preservar informações temporais, como padrões e correlações, as torna aplicáveis em uma variedade de áreas, incluindo processamento de linguagem natural, previsão financeira e análise de sinais temporais em engenharia. Esta abordagem flexível e adaptável das RCRs as

torna uma ferramenta valiosa aplicada na previsão e estudo de sistemas dinâmicos.

### 3.2 FORMAS DE APRENDIZAGEM DE MÁQUINA

A escolha da abordagem de Aprendizado de Máquina (AM) está intrinsecamente ligada ao tipo de problema a ser solucionado. O método supervisionado é aplicado em situações onde os dados de treinamento fornecidos ao algoritmo de AM incluem as respostas desejadas. Conforme ilustrado na Figura 8, durante a fase de treinamento, o algoritmo utiliza um sinal de referência, conhecido como “professor”, e busca minimizar o erro em relação a esse sinal (Izbicki e Santos, 2020). Algoritmos de aprendizado supervisionado incluem métodos como regressão linear, árvores de decisão, redes neurais, entre outros.

Figura 8 – Na aprendizagem supervisionada é necessário utilizar um sinal "professor", que será a referência para corrigir a resposta real.



Fonte: Adaptado de (Haykin, 2001)

Em contraste, quando o objetivo é identificar padrões nos dados sem uma referência explícita, a técnica não supervisionada é mais recomendada. Nesse cenário, o sistema tenta aprender sem um sinal 'professor'. Na forma não supervisionada, são aplicadas técnicas como *clustering* (agrupamento) ou análise de componentes principais (ACP), buscando explorar estruturas ou padrões nos dados de forma independente.

Outro aspecto importante da AM consiste em escolher entre aprendizagem *online* e *offline*. Na aprendizagem online, o algoritmo realiza o aprendizado monitorando algum observável em tempo real, enquanto na aprendizagem *offline*, o algoritmo utiliza um conjunto de dados já coletados do sistema para realizar seu treinamento. Vale ressaltar que existem outras formas de aprendizado de máquina como o semi-

supervisionado, o aprendizado por reforço, há também aqueles baseados em instâncias ou em modelos, etc.

Neste trabalho, optou-se pela técnica supervisionada, uma vez que a simulação numérica permitiu a obtenção antecipada do sinal de referência. Além disso, ao longo desta pesquisa, foram avaliadas as abordagens de aprendizagem *online* e *offline*.

### 3.3 REDES NEURAIS ARTIFICIAIS

Uma vez estabelecidas as formas de aprendizagem de máquina, o próximo passo será compreender as redes neurais artificiais (RNA). Para isso, serão apresentados alguns fundamentos sobre redes ou grafos e, em seguida, serão explorados os elementos básicos de processamento de uma RNA, os neurônios artificiais, juntamente com os tipos de arquitetura, considerando o fluxo de informação e a capacidade de sincronização.

#### 3.3.1 Redes

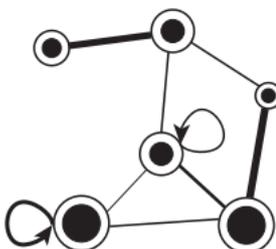
Redes ou grafos são formas abstratas de representar as relações entre diversos elementos<sup>1</sup> (Boccaletti *et al.*, 2006). Cada elemento de uma rede é denominado nó ou nodo e a conexão entre os nós da rede é representada por uma linha, chamada aresta. A força dessa conexão pode ser representada alterando a espessura da aresta ou especificando sobre ela um valor, o seu peso sináptico<sup>2</sup>. Por exemplo, na Figura 9 é representado um grafo com seis nós conectados por arestas de diferentes intensidades, as quais são representadas pela variação na espessura das arestas. Alguns desses nós possuem *autoloops*, os quais são criados ao introduzir um atraso temporal na informação reinserida no próprio nó. Note também que há uma diferenciação na importância dos nós, através da variação do diâmetro. Esta propriedade permite identificar nós com alguma propriedade especial, por exemplo, em algumas redes de telecomunicações existem estações *HUB* centralizadoras, que recebem muitas conexões, assim o diâmetro aumentado indica, neste caso, uma estação com alta capacidade de tráfego de dados.

A representação por rede auxilia no entendimento da topologia, aspecto fundamental para avaliar sua dinâmica nos quesitos de fluxo de informações entre nós; condições de sincronização da rede; pontos de convergência, divergência e clusterização da rede (Boccaletti *et al.*, 2006).

<sup>1</sup> Os elementos podem ser quaisquer unidades que possuam alguma relação e dependem da natureza do fenômeno. Alguns exemplos são os eventos que ocorrem em uma linha temporal de vida, como nascimento, graduação, casamento, morte; a hierarquia de uma empresa como chefe, diretores, gestores, técnicos; a comunicação entre dispositivos de telecomunicações, como roteadores, *switches*, servidores, discos de backup; a relação entre os capítulos, seções e subseções desta dissertação, as redes sociais ou a própria Internet, etc.

<sup>2</sup> Esta expressão foi utilizada em analogia com as conexões das redes neurais.

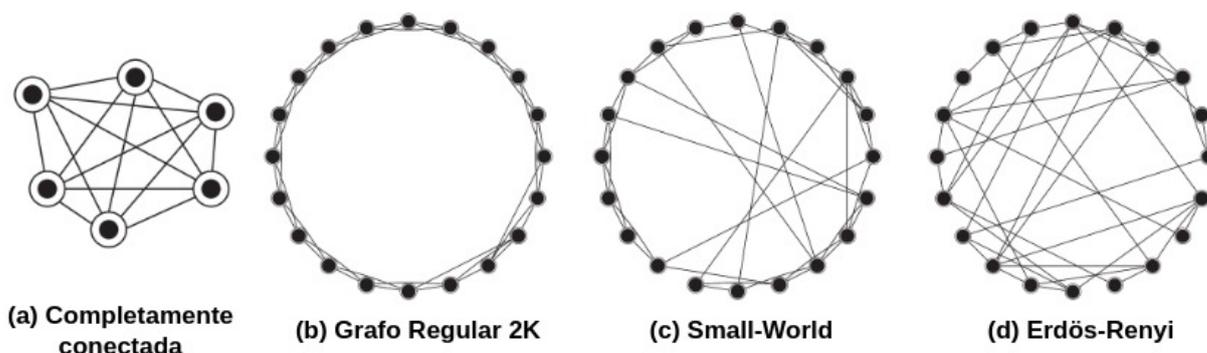
Figura 9 – Exemplo de grafo com representação de nós (círculos de diversos tamanhos), arestas (retas interligando os nós), autoloops (seta que começa e termina no mesmo nó, ao centro e à esquerda da figura), forças de conexão (uma forma de mostrar a força de conexão entre os nós é variar a espessura da aresta que os interliga).



Fonte: Adaptado de (Eroglu, Lamb e Pereira, 2017)

Na Figura 10 estão representadas algumas topologias de redes, cuja análise permite identificar configurações de rede mais adequadas aos objetivos de projeto. A título de exemplo, o estudo de Eroglu (Eroglu, Lamb e Pereira, 2017) indica que uma rede completamente conectada, representada na Figura 10(a), sincroniza melhor que uma rede com topologia *small-world*, Figura 10(c). Por outro lado, redes na configuração *Erdős-Renyi*, Figura 10(d), apresentam maior robustez a perturbações.

Figura 10 – Importância de escolher a topologia correta para o objetivo. A rede completamente conectada (a) sincroniza melhor que o Grafo Regular 2k (b) e a topologia Small-World (c), em contrapartida, a topologia Erdős-Renyi (d) é mais robusta a perturbações.



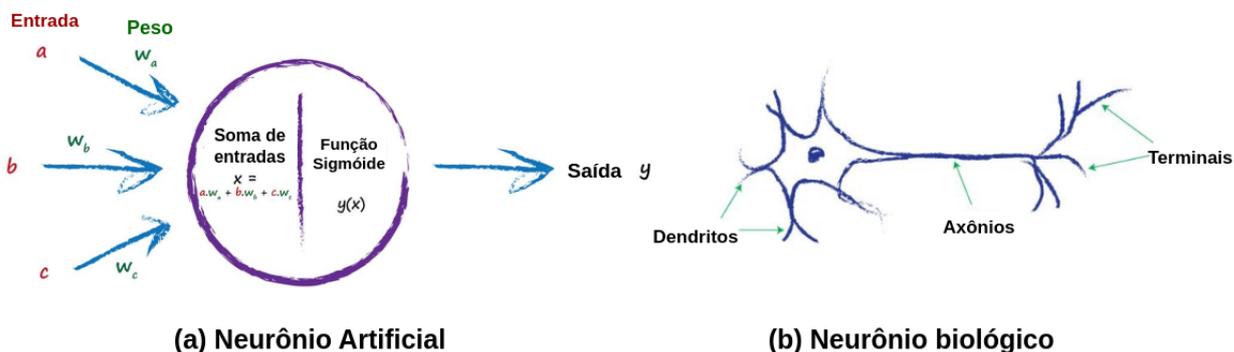
Fonte: Adaptado de (Eroglu, Lamb e Pereira, 2017)

As redes cujos nós são neurônios artificiais, Figura 11 (a), são a base original onde foram desenvolvidas as RCR e serão analisadas na próxima seção.

### 3.3.2 Redes Neurais Artificiais

O cérebro humano, é um sistema de processamento altamente *complexo*, *não-linear* e *paralelo*, que trata as informações de forma totalmente diferente dos computadores convencionais (Haykin, 2001). Além disso, é capaz de aprender com a experiência, apresentando alta *plasticidade*, sendo sua menor unidade de processamento o *neurônio*, representado na Figura 11 (b). Na busca de modelos para reproduzir

Figura 11 – O neurônio artificial e o biológico. Note que para o neurônio artificial replicar o comportamento do biológico é necessário aplicar uma soma ponderada em sua entrada, para então "ativar" o neurônio artificial executando a função sigmóide nesta soma ponderada.



Fonte: Adaptado de (RASHID, 2016)

a computação cerebral, foram desenvolvidas as redes neurais artificiais (RNA), cuja unidade básica de processamento é o neurônio artificial.

Na Figura 11(a) e (b) são apresentados o diagrama de funcionamento de um neurônio artificial e um biológico, respectivamente. O neurônio biológico, de forma simplificada, recebe estímulos de outras unidades conectadas a ele através dos dendritos. O sinal é então processado no corpo celular e transmitido através do axônio. No modelo artificial os dendritos são representados pelas entradas  $a$ ,  $b$ ,  $c$  que recebem sinais ou dados de conexões externas. Os sinais são então somados de forma ponderada pelos seus respectivos pesos ( $W_a$ ,  $W_b$  e  $W_c$ ), para então serem processados pela função de ativação do neurônio, que simula o limiar de excitação do neurônio biológico. O sinal processado é então enviado para a saída  $y$ .

Inicialmente o neurônio foi modelado para apresentar uma saída binária, onde o neurônio artificial está ativo ou inativo. Com o avanço de estudos na área, surgiram propostas de incorporar ao neurônio funções de ativação não lineares, como a função sigmóide e a tangente hiperbólica, que produzem uma saída contínua, tipicamente entre -1 e 1. Essas funções oferecem uma gama mais ampla de respostas, possibilitando uma representação mais complexa de padrões e informações nos modelos neuronais. Para os propósitos deste trabalho, será utilizada a função tangente hiperbólica para ativar os neurônios artificiais.

No ambiente artificial, os neurônios são interconectados em diversas topologias, resultando na formação da RNA, assim como ocorre no cérebro humano. As RNA são normalmente desenvolvidas por meio de programação em computadores digitais (Trouvain *et al.*, 2020; Steiner *et al.*, 2022), mas também podem ser implementadas em *hardware* dedicado (Cucchi *et al.*, 2022). Assim, uma RNA é caracterizada como uma rede *maciçamente* interligada e *paralelamente* distribuída, constituída de unidades de processamento simples. Sua principal vantagem é a capacidade de aprender a partir de dados e generalizar as informações adquiridas, diferentemente de um simples

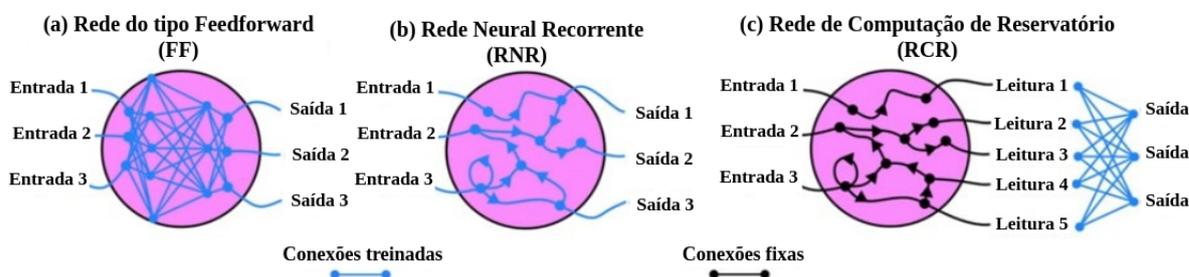
mapeamento entrada-saída (Haykin, 2001). As RNA se aproximam do comportamento cerebral quanto a duas características:

- O conhecimento é adquirido pela rede a partir de seu ambiente, por meio de um processo de aprendizagem (Haykin, 2001); e
- Forças de conexão entre neurônios, conhecidas como pesos sinápticos, são ajustados durante o procedimento de aprendizagem armazenando o conhecimento adquirido (Haykin, 2001).

O procedimento utilizado para realizar o processo de aprendizagem é chamado de *algoritmo de aprendizagem*. Como mencionado acima, sua função é modificar os pesos sinápticos da rede de uma forma ordenada para alcançar um objetivo de projeto desejado, como por exemplo, reconhecimento de padrões, previsão de séries temporais, classificação, etc, sendo capaz de generalizar o conhecimento adquirido.

Na Figura 12, estão elencados três tipos de redes neurais que diferem quanto a sua arquitetura e a maneira como ocorre sua aprendizagem. As redes *feedforward* (FF), Figura 12(a), foram as primeiras a serem desenvolvidas. Em sua arquitetura as informações fluem diretamente da camada de entrada através das camadas intermediárias (conhecidas como camadas ocultas) até a camada de saída, ou seja, a informação avança em uma direção, da entrada para a saída, sem *loops* ou realimentações. Entretanto, este tipo de rede demanda uma grande capacidade computacional para convergir, pois durante a aprendizagem os pesos sinápticos de todas as camadas da rede precisam ser ajustados.

Figura 12 – Tipos de RNA: As conexões em azul são alteradas durante o processo de aprendizagem enquanto as conexões em preto são fixas. (a) Redes neurais feedforward ou redes neurais de alimentação direta em português. (b) Redes neurais recorrentes (RNR) onde as setas indicam estruturas de autoloop e feedback. (c) Redes de computação de reservatório, cujo reservatório é fixo e apenas a camada de saída (leitura) é ajustada durante o processo de aprendizagem.



Fonte: Adaptado de (Albeverio, Jentsch e Kantz, 2006)

As redes neurais recorrentes (RNRs), Figura 12(b), são redes que possuem estruturas de *autoloop* e *feedback*. Esses recursos habilitam essas redes para lidar com sistemas dinâmicos como reconhecimento de voz, detecção de movimento, séries temporais, etc. São redes intrinsecamente não-lineares, permitindo resolução de uma gama maior e mais complexa de problemas, porém ainda demandam grande esforço

computacional para seu treinamento, pois ainda precisam modificar os pesos sinápticos em todas as camadas para efetivarem o treinamento. Por fim, as redes de computação de reservatório (RCR), representadas na Figura 12(c) são redes intrinsecamente não-lineares baseadas em RNRs, tendo como principal característica que apenas a camada de saída é ajustada durante o procedimento de aprendizagem. Isso permite uma convergência mais rápida, já que requer treinamento somente nas conexões da camada de saída (Lukoševičius, 2012; Pyragas e Pyragas, 2020).

### 3.4 REDES DE COMPUTAÇÃO DE RESERVATÓRIO

As redes de computação de reservatório (RCRs) têm uma arquitetura baseada nas RNRs (Steiner *et al.*, 2022), constituídas em camadas, onde a camada intermediária consiste em uma rede com topologia gerada de forma aleatória, chamada de reservatório e o treinamento ocorre exclusivamente na camada de saída ou camada de leitura. Portanto, as RCR combinam a capacidade das RNRs com uma estratégia de treinamento simples e eficiente, sendo capazes de lidar com sistemas dinâmicos, assim como as RNRs. E, conseqüentemente, têm a capacidade de aprender os atratores de sistemas caóticos (Lu, Hunt e Ott, 2018; Antonik *et al.*, 2018).

Aplicações das RCRs abrangem uma ampla gama de áreas como previsão de eventos extremos em fluidos caóticos (Racca e Magri, 2022), reconhecimento de voz, controle de radares, processamento de imagens, controle de plasma, análise de dados musicais, classificação de batimentos cardíacos, etc. (Steiner *et al.*, 2022; Pyragas e Pyragas, 2020; Pathak *et al.*, 2018; Lu, Hunt e Ott, 2018).

Na área de AM, as primeiras ideias relacionadas com o uso de técnicas em que apenas a camada de saída é treinada remontam a Kirby e Day, que publicaram um artigo pioneiro em 1990, propondo uma arquitetura de rede para lidar com aprendizado sequencial, denominada de redes de reverberação de contexto (Kirby e Day, 1990); e a Dominey, que em 1995, propôs um modelo simplificado de uma rede neural de duas camadas, baseado na anatomia cortical e subcortical do sistema oculomotor de primatas. A primeira camada corresponde ao córtex pré-frontal que desempenha o papel do reservatório e a segunda camada corresponde ao estriado, que é a camada de leitura, na qual utilizou técnicas de regressão linear para, a partir desta camada, coletar informações e enviá-las ao núcleo caudal (Dominey, 1995; Nakajima, 2021).

No início dos anos 2000, foram propostos, de forma independente, dois modelos fundamentais de RCRs. O modelo de rede de eco de estado (*eco state network*-ESN) (Jaeger, 2001), abordando a relação entre o eco de estado e a capacidade da memória do reservatório. O eco de estado implica que o efeito de um estado anterior e uma entrada anterior devem desaparecer gradualmente em um estado futuro, à medida que o tempo passa, e não deve persistir ou mesmo ser amplificado (Ferreira, 2011). E o

modelo de máquina de estado líquido, em inglês *liquid state machines* (LSM) (Maass, Natschläger e Markram, 2002; Maass, 2011). As LSM foram desenvolvidas seguindo restrições das redes neurais biológicas, de modo que a probabilidade de conexão entre dois neurônios depende da distância entre suas posições. Um reservatório desse tipo é frequentemente chamado de "líquido", e a operação do LSM é denominada "computação líquida".

O conceito de redes de computação de reservatório (RCR) engloba uma variedade de técnicas de aprendizagem de máquina que utiliza projeções aleatórias e não lineares dos sinais de entrada em um espaço de altas dimensões (Steiner *et al.*, 2022). Tais redes também são entendidas como um paradigma de rede neural recorrente, construída de forma aleatória, onde a camada intermediária não necessita de treinamento (Oliveira e Juliani, 2018). Por ser um campo prolífico na área de aprendizagem de máquina, atualmente existem vários modelos de RCR, tais como as NG-RC (Gauthier *et al.*, 2021), ELM, LSM (Steiner *et al.*, 2022). Além disso, desde que atendam os requisitos de alta-dimensão, não-linearidade, decaimento de memória e a propriedade da separação, as RCR podem ser construídas utilizando-se quaisquer tipos de redes complexas (Tanaka *et al.*, 2019), o que inclui sistemas físicos com características não-lineares (Cucchi *et al.*, 2022).

O modelo básico da RCR, utilizado nesta dissertação, pode ser visualizado na Figura 13, que apresenta o diagrama esquemático do seu funcionamento. Neste caso, trata-se de uma rede neural de tempo discreto (Jaeger, 2001), composta por três camadas. A camada de entrada, responsável por preparar os dados para processamento. A camada intermediária ou reservatório (conhecida em inglês como *reservoir*) caracterizada como uma RNR que realiza transformações não-lineares nos sinais de entrada, mapeando-os em um espaço de alta dimensão. E a camada de saída ou leitura, onde ocorre o treinamento da rede, geralmente realizado utilizando a técnica de regressão linear.

De forma geral, a arquitetura e o funcionamento de uma RCR foi discutida, descrevendo cada camada e sua função. A próxima etapa será abordar o processo de treinamento supervisionado de uma RCR, que estará treinada e efetivamente pronta para uso após a conclusão dos seguintes passos:

1. Selecionar ou gerar o conjunto de dados composto pelo sinal de entrada  $\mathbf{u}(\mathbf{t})$  e sua saída esperada, definida por  $\mathbf{Y}^{alvo}(t)$ , que desempenha o papel do sinal "professor" no contexto da aprendizagem supervisionada. Esses dados podem ser obtidos a partir de observações experimentais ou de simulações numéricas do fenômeno físico estudado. Este conjunto de dados utilizado para treinar a RCR, pode ser organizado na forma de matrizes ou tensores, onde cada linha representa um passo temporal e cada coluna um observável diferente, ou alterna-

tivamente, pode ser organizado ponto a ponto, no caso de treinamento *on-line*, mantendo a mesma estrutura anterior. Muitas vezes é razoável normalizar os dados na entrada, embora a normalização não seja um requisito obrigatório para o funcionamento da RCR (Jaeger, 2001). Além disso, um requisito essencial para o processo de aprendizagem é a qualidade e a quantidade de dados disponível no treinamento. A quantidade de dados deve ser representativa dos eventos ocorridos no sistema, permitindo a RCR generalizar suas previsões para todos os casos possíveis. No caso de dados ruidosos, a RCR enfrentará dificuldades em identificar padrões nos dados, o que pode afetar a precisão das previsões, tornando-a mais susceptível a erros. Portanto, é necessário estabelecer uma proporção entre dados de treinamento e teste na amostra. Essa divisão visa garantir uma avaliação do desempenho do modelo em dados não vistos, avaliando a capacidade da RCR em generalizar suas previsões.

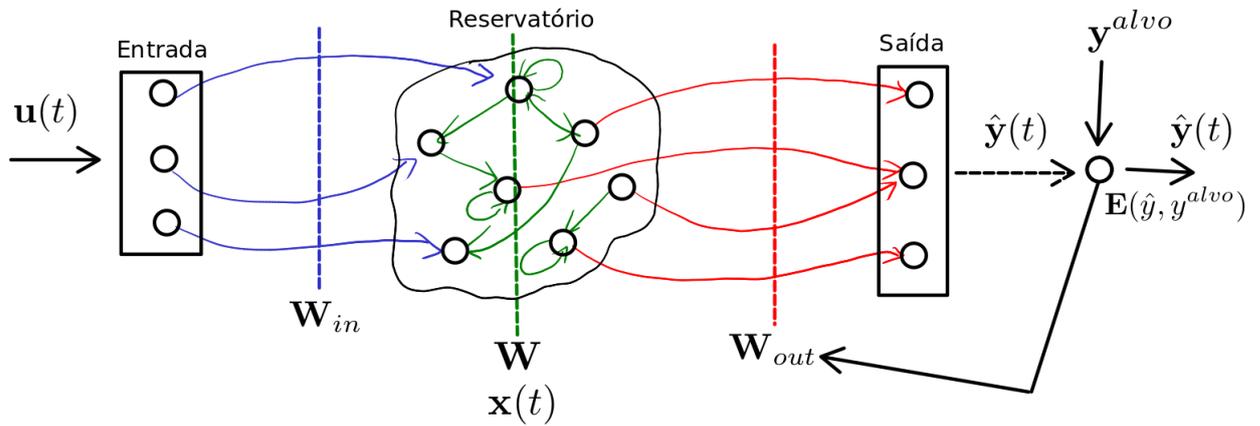
2. Em seguida, deve ser gerada a matriz de pesos sinápticos que conecta os neurônios de entrada aos do reservatório (representada pela linha tracejada em azul na Figura 13), chamada  $\mathbf{W}_{in}$ , com dimensões  $\dim(u(t)) \times \dim(x(t))$ . A matriz  $\mathbf{W}$ , responsável pelas conexões internas do reservatório (representada pela linha tracejada em verde na Figura 13), com dimensões  $\dim(x(t)) \times \dim(x(t))$ . E  $\mathbf{W}_{back}$ , a matriz de *feedback*, que possui as mesmas dimensões da matriz  $\mathbf{W}_{out}$ , porém trata-se de um recurso não obrigatório para o funcionamento da RCR, responsável por enviar uma amostra da saída ao reservatório. As três matrizes ( $\mathbf{W}_{in}$ ,  $\mathbf{W}$  e  $\mathbf{W}_{back}$ ) são produzidas a partir de um conjunto de parâmetros chamados de hiperparâmetros, os quais definem as propriedades da rede neural gerada. Os hiperparâmetros serão discutidos na próxima seção. Uma vez que estas matrizes são construídas, os pesos de suas conexões permanecem inalterados.
3. O próximo passo consiste diretamente no processo de treinamento, sendo responsável por gerar a matriz  $\mathbf{W}_{out}$ , que mapeia os vetores de estado do reservatório para uma saída que se aproxima do  $\hat{\mathbf{Y}}_{alvo}$ , definindo os pesos sinápticos das conexões entre a camada de reservatório e a camada de saída. Esta é a única matriz modificada durante o treinamento da RCR e, no contexto do treinamento supervisionado, é obtida minimizando a raiz quadrada do erro médio, em inglês, root mean square error (RMSE), equação 3.1, entre  $\mathbf{Y}_{alvo}$  e o  $\hat{\mathbf{Y}}(t) = \mathbf{W}_{out}\mathbf{X}$ . A matriz  $\mathbf{W}_{out}$  é a matriz incógnita com os pesos ainda desconhecidos e a matriz  $\mathbf{X}$  é o resultado obtido na saída do reservatório com todos os dados de treinamento. Uma solução para encontrar  $\mathbf{W}_{out}$  minimizando a equação 3.1 é o método dos mínimos quadrados, equação matricial 3.2, cuja solução é obtida aplicando técnicas de cálculo numérico, de modo que haja a convergência de  $\hat{\mathbf{Y}}(t)$  para o  $\mathbf{Y}_{alvo}$ . A equação 3.1 também é conhecida como função custo;

4. Neste ponto, a RCR já estará treinada, bastando apenas alimentar a entrada com novos sinais  $\mathbf{u}(t)$  e, a partir dos vetores de estado interno do reservatório representados por  $\mathbf{x}(t)$ , computar o sinal de saída  $\hat{\mathbf{Y}}(t)$ .

$$E(\hat{y}, y^{alvo}) = \frac{1}{N_y} \sum_{i=1}^{N_y} \sqrt{\frac{1}{T} \sum_{n=1}^T (y_i(t) - y_i^{alvo}(t))^2}. \quad (3.1)$$

$$\mathbf{W}_{out} = \mathbf{Y}_{alvo} \mathbf{X}^T (\mathbf{X} \mathbf{X}^T + \beta \mathbf{I})^{-1}. \quad (3.2)$$

Figura 13 – Arquitetura de uma RCR composta de três camadas: entrada, reservatório e saída.  $\mathbf{u}(t)$  corresponde ao sinal de entrada,  $\mathbf{W}_{in}$  a matriz que conecta os neurônios da camada de entrada ao reservatório,  $\mathbf{x}(t)$  aos vetores de estado interno do reservatório e  $\mathbf{W}_{out}$  a matriz que conecta o reservatório a camada de saída.  $\hat{\mathbf{Y}}(t)$  corresponde ao sinal que no processo de aprendizagem é comparado ao sinal esperado  $\mathbf{Y}_{alvo}$ , produzindo o sinal de erro  $\mathbf{E}(\hat{y}, y^{alvo})$ .



Fonte: Autor, 2022

Dessa forma, uma vez obtida as matrizes  $\mathbf{W}^{in}$ ,  $\mathbf{W}$ ,  $\mathbf{W}^{out}$  e  $\mathbf{W}^{back}$ , os vetores dos dados de entrada são inseridos na RCR, onde são processados conforme duas equações dinâmicas discretas, em que a variável  $n$  corresponde a discretização da variável  $t$ . A equação 3.3, determina o estado do reservatório em função do tempo, enquanto a equação 3.4, determina a saída. Embora a matriz  $\mathbf{W}^{back}$  apareça nas equações, não é requisito necessário para o funcionamento da RCR, sendo atribuída a ela valores iguais a zero. Na equação 3.3, a função  $F$  corresponde a função de ativação do neurônio, geralmente utilizada a função tangente hiperbólica por adicionar não linearidade às ativações e possuir uma faixa de valores que vão de -1 a 1. O  $Lr$ , denominado *Leaking rate* é um dos hiperparâmetros discutidos na próxima seção. Na equação 3.4,  $F^{out}$  pode ser uma função de ativação ou a função identidade, sendo esta última utilizada nesse trabalho.

$$\mathbf{x}(n) = (1 - Lr)\mathbf{x}(n - 1) + LrF(\mathbf{W}^{in}[1; \mathbf{u}(n)] + \mathbf{W}\mathbf{x}(n - 1) + \mathbf{W}^{back}\mathbf{y}(n - 1)). \quad (3.3)$$

$$\hat{\mathbf{y}}(n) = F^{out}(\mathbf{W}^{out}[1; \mathbf{x}(n)]). \quad (3.4)$$

### 3.4.1 Hiperparâmetros de uma RCR

O reservatório é o núcleo de processamento da RCR. Em geral, sua construção é realizada de forma aleatória seguindo um conjunto de hiperparâmetros (HP) ou parâmetros globais. Na RCR, apenas a camada de saída é treinada, portanto, a criação de uma RCR adequada para uma determinada aplicação, depende diretamente do ajuste de sua topologia e dos hiperparâmetros (Ferreira, 2011). Na tabela 1, são apresentados os valores aplicáveis a cada hiperparâmetro da RCR:

Tabela 1 – Valores aplicáveis aos hiperparâmetros da RCR

Parâmetros	Valores
Lr	0,0 a 1,0
Sr	0,0 a $+\infty$
N	$\{x   x \in \mathbb{Z}_+^*\}$
lss	0,0 a $+\infty$
Seed	$\{x   x \in \mathbb{Z}_+^*\}$
Activ	$\{x   -1 \leq x \leq 1\}$
Rc_con	0,0 a 1,0
Noise_rc	0,0 a $+\infty$

Fonte: Autor, 2022.

O hiperparâmetro N define o número de nós ou neurônios do reservatório e seu aumento implica em maior capacidade de processamento, porém também aumenta a demanda computacional. O Lr (*Leaking rate*) define a taxa de amostragem temporal dos sinais, que deve ser modificado conforme variar a frequência da tarefa que a RCR irá executar, se a frequência da amostra temporal for alta, o Lr também deverá ser maior, olhar equação 3.3. O raio espectral, Sr (*Spectrum radius*), é o maior autovalor da matriz  $\mathbf{W}$  do reservatório e afeta os valores de seus pesos, os quais se forem muito grandes para a tarefa, podem distorcer o sinal na saída, se forem muito pequenos, a RCR perde capacidade de reproduzir alguns dados, a literatura sugere que o uso de um  $Sr < 1$  converge para a maioria dos problemas (Lukoševičius, 2012; Jaeger, 2001; Nakajima, 2021). A função de ativação (*Activ*) influencia na resposta não-linear da RCR e nos tipos de tarefas que a RCR é capaz de resolver, normalmente é utilizada a função  $\tanh$  que mapeia os valores da soma ponderada de entrada do neurônio para a faixa de  $[-1, 1]$  em sua saída. O HP Escala de Entrada, da sigla em inglês lss (*Input Scaling*), amplia ou reduz os valores da matriz de pesos  $\mathbf{W}_{in}$ , aplicando um ganho ou atenuação nos sinais de entrada da RCR. A semente *Seed* é um parâmetro utilizado especificamente para permitir que seja possível reproduzir a RCR, ele adiciona um número que será utilizado como chave para gerar a RCR. O fator de Tikhonov

( $\beta$ ) ou *ridge* é utilizado para impor uma restrição aos valores dos pesos obtidos para  $\mathbf{W}_{out}$ , sua função é garantir que durante a regressão linear, os valores dos pesos da matriz não cresçam ou decresçam de forma abrupta, é o parâmetro  $\beta$  que aparece na equação 3.2. O *rc\_con* altera a densidade das conexões entre os neurônios do reservatório, quanto maior, mais interconectados são os neurônios do reservatório. O *noise\_rc* modifica a quantidade de ruído dentro do reservatório, que afeta a precisão da RCR na realização das tarefas. Neste trabalho o ruído foi mantido nulo com o intuito de avaliar a RCR em condições ideais

Sobre a topologia, dois quesitos devem ser avaliados, a distribuição dos neurônios na rede, influenciado pelo HP *Sr* e a densidade das conexões entre eles, HP *Rc\_con*. A escolha do tipo de rede modifica simultaneamente estes dois quesitos. Como exemplo, na rede do tipo regular 2k, Figura 10(b), cada nó possui 2k links, num total de  $M=kN$  links, onde  $N$  é o número total de nós na rede (Eroglu, Lamb e Pereira, 2017). Por outro lado, na rede do tipo small-world, o comprimento do caminho médio é curto e a clusterização é alta, embora seja construída a partir de uma rede do tipo regular 2k, seus links são estabelecidos por probabilidades de conexão (Eroglu, Lamb e Pereira, 2017), o que torna este tipo de rede mais esparsa e mais robusta diante de uma tarefa que envolve muito ruído.

### 3.4.2 Otimização dos Hiperparâmetros da RCR

As matrizes  $\mathbf{W}^{in}$  e  $\mathbf{W}$  não são modificadas após a produção da RCR, portanto devem ser produzidas favorecendo a resolução do problema. Como sua construção é consequência da correta escolha dos hiperparâmetros, torna-se fundamental otimizar os oito hiperparâmetros da tabela 1<sup>3</sup>. Há diversas opções de otimização de parâmetros, porém segundo Bergstra (Bergstra *et al.*, 2013), o algoritmo TPE (*Tree-structured Parzen Estimator*) supera as estratégias randômicas e aquelas de escolha manual de parâmetros, pois sua busca de parâmetros é realizada utilizando probabilidades condicionais, especificamente, aplica o teorema de Bayes para selecionar iterativamente o melhor parâmetro durante a otimização. Assim, o algoritmo de otimização TPE foi escolhido para uso neste trabalho.

### 3.4.3 Treinamento da RCR

O treinamento da RCR consiste em determinar a matriz dos pesos sinápticos na saída  $\mathbf{W}_{out}$ , que minimiza o erro quadrático médio (MSE), o qual é equivalente a minimizar o RMSE, equação 3.1. Ao minimizar o MSE, a equação matricial 3.5 é obtida,

<sup>3</sup> Como a função do hiperparâmetro Semente (Seed) é garantir a reprodutibilidade da RCR, pode ser mantido fixo.

$$\mathbf{W}^{out} = \mathbf{Y}^{alvo} \mathbf{X}^T (\mathbf{X} \mathbf{X}^T)^{-1}. \quad (3.5)$$

onde a matriz  $\mathbf{Y}^{alvo} \in R^{N_y \times T}$ , em que  $N_y$  representa o número de elementos da matriz alvo e  $T$  representa o número de pontos no conjunto de treinamento. A matriz  $\mathbf{X} \in R^{(1+N_x) \times T}$  consiste no resultado do processamento de todos os dados de treinamento, onde 1 é o valor do neurônio reservado para o viés (*bias*) e  $N_x$  é a dimensão dos dados na saída do reservatório. Esse procedimento equivale à aplicar o método dos mínimos quadrados. A equação 3.5, no entanto, apresenta uma série de problemas no processo de aprendizagem, como o excesso de aprendizagem, do inglês *overfitting*, que impede a RCR de generalizar as suas previsões.

Visando evitar o problema de *overfitting* durante o cálculo da regressão linear, o termo  $\beta \mathbf{I}$  é adicionado à equação 3.5, também conhecido como o fator de Tikhonov ou *ridge regression*. Dessa forma, o resultado será a equação 3.6.

$$\mathbf{W}^{out} = \mathbf{Y}^{alvo} \mathbf{X}^T (\mathbf{X} \mathbf{X}^T + \beta \mathbf{I})^{-1}. \quad (3.6)$$

O termo adicionado à equação 3.5 é chamado de fator de regularização. Esse fator ajusta a matriz  $\mathbf{W}^{out}$ , buscando um compromisso entre obter um erro pequeno no treinamento com os pesos da matriz de saída também pequenos (Lukoševičius, 2012). Portanto, este fator restringe os valores que podem ser obtidos na regressão linear, impedindo que fiquem muito grandes ou extremamente pequenos, evitando assim que a matriz  $\mathbf{W}_{out}$  seja obtida com valores distorcidos. Por exemplo, se  $\mathbf{W}_{out}$  tiver valores muito grandes, ao aplicar uma transformação linear em  $\mathbf{x}(n)$ , poderá amplificá-lo provocando uma saída instável na RCR. É importante observar que a inversão da matriz  $(\mathbf{X} \mathbf{X}^T + \beta \mathbf{I})^{-1}$  é uma tarefa computacional custosa, e existe uma série de métodos numéricos disponíveis para realizar essa inversão.

#### 3.4.4 Métricas de qualidade das previsões

Nesta seção serão detalhadas as métricas para avaliar a qualidade das previsões, serão utilizados três métricas: o coeficiente de determinação  $\mathbf{R}^2$ , o erro médio quadrático normalizado, em inglês, *normalized root mean squared error* **NRMSE** e o alcance temporal das previsões baseado em múltiplos do tempo de Lyapunov (TL).

O coeficiente de determinação  $\mathbf{R}^2$  é definido pela equação 3.7, sendo uma medida do grau de aproximação entre duas variáveis. Em termos percentuais, sua faixa de valores está entre  $0 \leq \mathbf{R}^2 \leq 100\%$ , onde o 0% significa que as duas variáveis são completamente descorrelacionadas e 100% são completamente correlacionadas.

$$\mathbf{R}^2[\%] = \left(1 - \frac{\sum_{i=0}^{N-1} (y_i - \hat{y}_i)^2}{\sum_{i=0}^{N-1} (y_i - \bar{y})^2}\right) 100 \quad (3.7)$$

O **NRMSE** consiste em normalizar a raiz quadrada do erro quadrado médio (RMSE) por algum parâmetro. O RMSE é uma medida da distância entre o valor predito e o valor de referência, assim, quanto menor este valor, melhor serão as previsões obtidas. Neste trabalho, o fator de normalização foi adotado como a distância entre o valor mínimo e o valor máximo do conjunto de dados, conforme equação 3.8.

$$NRMSE = \frac{RMSE}{y_{max} - y_{min}} \quad (3.8)$$

$$TL = \frac{1}{\lambda} = 10u.a. \quad (3.9)$$

Segundo o estudo (Racca e Magri, 2022), o inverso do expoente de Lyapunov global positivo fornece um parâmetro para avaliar o alcance temporal de previsibilidade em sistemas caóticos. O expoente de Lyapunov pode ser entendido considerando-se dois sistemas idênticos, que partem de condições iniciais ligeiramente diferentes. O expoente de Lyapunov,  $\lambda$ , é então determinado linearizando a equação diferencial para o erro, definido como a diferença das equações dos dois sistemas, ponto-a-ponto no espaço de fases. Para cada ponto da trajetória são determinados autovalores, dos quais é escolhido o maior valor ao longo de uma trajetória longa do sistema. Neste estudo, o valor de  $\lambda$  determinado é igual a 0,1, obtido a partir do algoritmo de (Dabrowski, 2014). O tempo de Lyapunov é definido como o inverso de  $\lambda$ , sendo igual a 10u.a., conforme equação 3.9, que servirá como parâmetro de tempo para avaliar a capacidade de previsão da RCR. Como o expoente de Lyapunov é uma medida da divergência entre dois sinais próximos no espaço de fase, o seu inverso, permite estabelecer uma escala de tempo de previsibilidade (Racca e Magri, 2022; Strogatz, 2018). Note que o tempo de Lyapunov indica um horizonte de previsibilidade, entretanto em sistemas caóticos é possível realizar a previsão somente até alguns TL. Todas as medidas de tempo serão proporcionais ao TL, logo se uma previsão tem alcance de 3TL, isso significa que a previsão antecipou 30 unidades arbitrárias de tempo, ou  $\frac{30}{10} = 3TL$ .

#### 4 PROPOSTA DE USO DA REDE DE COMPUTAÇÃO DE RESERVATÓRIO NA PREVISÃO DE SÉRIES CAÓTICAS E DESSINCRONIZAÇÕES

Neste capítulo, serão apresentados os passos para treinar de forma supervisionada a RCR, visando prever as séries caóticas de um circuito GB e as dessincronizações, incluindo a detecção de eventos extremos, observados quando dois circuitos GB são acoplados em  $V_2$ , no modo Diretor-Resposta unidirecional, aqui designado sistema DR. O plano geral consiste em obter os sinais professores, do circuito GB e do sistema DR, por simulação numérica, aplicá-los à RCR e utilizando um algoritmo de otimização, encontrar hiperparâmetros (HP) que maximizem a capacidade da RCR de realizar as previsões. Então gerar a RCR a partir dos HP otimizados e testá-la com novos dados para avaliar a qualidade do treinamento e das previsões.

As RCR podem ser realizadas em programas ou podem ser produzidas diretamente em hardware (Cucchi *et al.*, 2022). Neste trabalho foi utilizada a biblioteca *reservoirpy* (Trouvain *et al.*, 2020) (RPY), desenvolvida para realizar experimentos com arquiteturas de computação de reservatório em ambiente *Python*.

Para implementar a RCR, isto é, as equações matemáticas 3.3 e 3.4, a RPY utiliza álgebra matricial. Como detalhado na subseção 3.4.1, a capacidade da RCR de resolver problemas depende fortemente da correta escolha de seus hiperparâmetros. Além disso, a cada ciclo de execução do programa, a RPY cria uma RCR com diferentes matrizes  $\mathbf{W}_{in}$ ,  $\mathbf{W}$  e  $\mathbf{W}_{out}$ . Uma vez criada a RCR, as matrizes  $\mathbf{W}_{in}$  e  $\mathbf{W}$  não são mais modificadas, enquanto a matriz  $\mathbf{W}_{out}$  vai sendo alterada, em função do treinamento aplicado a camada de saída da RCR.

Portanto, para a RCR realizar as previsões, duas ações devem ser realizadas. A primeira é otimizar os parâmetros globais ou hiperparâmetros(HP) com enfoque para a tarefa escolhida. Este aspecto é fundamental, pois estudos indicam que a capacidade da RCR de resolver um problema específico depende diretamente da correta escolha de seus HP (Lukoševičius, 2012; Nakajima, 2021). A segunda consiste em treinar a RCR, especificamente a camada de saída do reservatório, para minimizar o erro médio quadrático, equação 3.1 da seção 3.4.

Assim, a proposta de treinamento e uso da RCR será dividida em três etapas:

1. Gerar os dados necessários para realizar as previsões das séries caóticas do circuito GB e das dessincronizações do sistema DR;
2. Realizar a otimização para selecionar os HP que aumentam a qualidade das previsões; e
3. Testar a RCR obtida no passo 2 variando o intervalo de previsão.

Estas etapas serão aplicadas à previsão de séries caóticas e de dessincronizações do sistema DR. Cabe observar que os experimentos serão realizados para descobrir como a resposta do sistema depende dos fatores (Barros Neto, Scarminio e Bruns, 2007). Neste caso, a resposta é a capacidade da RCR de prever séries caóticas e eventos extremos (dessincronizações de maior amplitude), enquanto os fatores, tabela 2, foram separados em três grupos, as variáveis ou combinação de variáveis de entrada; o conjunto de hiperparâmetros e o atraso aplicado ao sinal utilizado como professor no treinamento da RCR. Importante frisar que o objetivo é descobrir quais destes fatores influenciam na capacidade da RCR de realizar as previsões de séries caóticas (variáveis dos circuitos GB) e do  $\|\mathbf{X}_\perp\|$  (medida do estado de sincronização do sistema DR) com NRMSE mínimo,  $R^2$  máximo e o maior número de tempos de Lyapunov (TL) possível.

Tabela 2 – Fatores que influenciam na qualidade das previsões da RCR

<b>Grupo</b>	<b>Fatores</b>
Variáveis do sistema DR	$V_{1d}, V_{2d}, I_d, V_{1r}, V_{2r}, I_r, \ \mathbf{X}_\perp\ $
HP da RCR	N, Lr, Sr, Rc_con, Activ, Iss, Noise_rc, Iss, Ridge
Intervalos de previsão em TL	0,5; 1; 1,5; 2; 3; 4; 5; 10; 15; 20

Fonte: Autor, 2023.

#### 4.1 SIMULAÇÃO NUMÉRICA DAS VARIÁVEIS DO CIRCUITO GB

A dinâmica de dois circuitos GB sincronizados, aqui denominado sistema DR, acoplados em  $V_2$ , oscilando na região caótica, foi simulada através da resolução das equações 2.6 e 2.7 pelo método de Runge-Kutta de 4<sup>o</sup>. Visando aplicações experimentais e tomando como base os parâmetros adimensionais  $R_1 = 1,2$ ,  $R_2 = 3,44$  e  $R_3 = 0,193$ , que colocam o sistema GB para operar na região caótica. Em cada circuito GB foram gerados aleatoriamente valores adimensionais de seus componentes com uma divergência máxima de 1% (Feitoza, 2022) entre os componentes do circuito diretor e do resposta. Para a força de acoplamento, constante  $c$ , foi utilizado o valor igual a 4, que é o mesmo valor utilizado no estudo de Gauthier (Gauthier e Bienfang, 1996), com o intuito de garantir uma sincronização robusta, porém as instabilidades na origem quebram essa sincronização gerando eventos extremos. Evita-se valores pequenos, pois geram uma grande quantidade de pequenos eventos de dessincronização que não são evento extremos. Na tabela 3 estão organizados os principais parâmetros que serão utilizados para simular o sistema DR.

Esta simulação produziu  $10^6$  pontos relativos às seis variáveis,  $V_{1d}, V_{2d}, I_d, V_{1r}, V_{2r}, I_r$ , envolvidas no sistema DR.

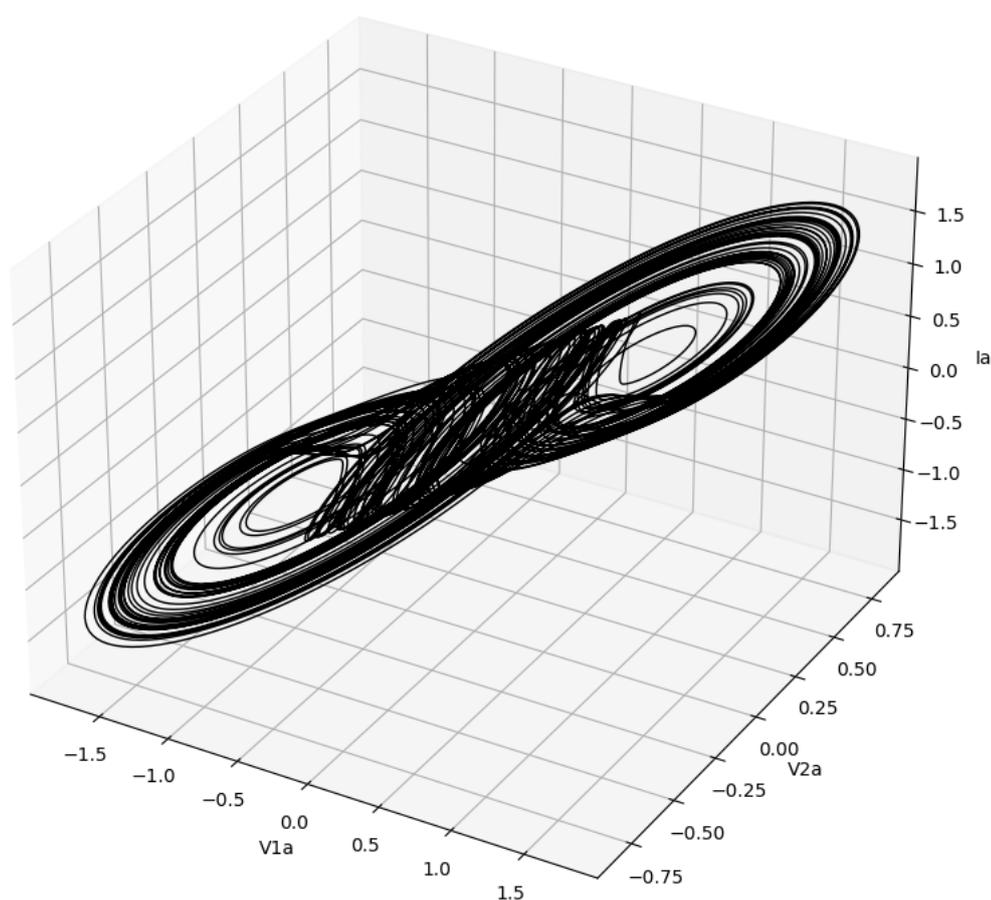
Na figura 14 está o atrator estranho obtido a partir dos dados simulados, demonstrando a natureza caótica do circuito GB simulado.

Tabela 3 – Valores utilizados para simular numericamente o acoplamento do sistema DR em  $V_2$

Parâmetro	Valores [u.a.]
$\Delta t$	$10^5$
dt	0,1
V1i, V2i, li	valores aleatórios entre 0,0 a 0,1
R1, R2, R3	1,2; 3,44; 0,193
c	4,0

Fonte: Autor, 2023.

Figura 14 – Atrator estranho do circuito GB-a obtido na simulação numérica



Fonte: Autor, 2023

## 4.2 OTIMIZAÇÃO DOS HIPERPARÂMETROS DA RCR

Para otimizar os HP da RCR, serão executados os passos apresentados no fluxograma da figura 15, os quais foram realizados para otimizar a RCR nas duas tarefas. Os conjuntos utilizados ou obtidos em cada etapa receberão nomes especiais, HP01 para os hiperparâmetros iniciais, HP02 para os hiperparâmetros obtidos variando todos os parâmetros, ao mesmo tempo, HP03 para os hiperparâmetros obtidos variando um parâmetro por vez. Para as dessincronizações também foi buscado HP04, que consiste em utilizar o método de busca de HP03, porém partindo dos hiperparâmetros HP01.

Esta otimização foi realizada para avaliar se a estratégia do teste 02 do fluxograma da figura 15 alcança a otimização ótima sem passar pela etapa do teste 01 do fluxograma da figura 15.

#### 4.2.1 Hiperparâmetros para Previsão de Séries Caóticas

O teste inicial será gerar a RCR utilizando HP01, tabela 4. Em relação a HP01, o ruído no reservatório (Noise\_rc) foi escolhido nulo para avaliar a RCR em condições ideais, a função de ativação (Activ) utilizada foi  $\tanh$  (Pyragas e Pyragas, 2020; Jaeger, 2001; Lukoševičius, 2012) que produz saídas em uma faixa de valores de  $[-1,1]$  e permite a aprendizagem de padrões mais complexos, o raio espectral (Sr) foi escolhido seguindo a referência da literatura, que sugere que um valor menor que 1 faz a RCR convergir mais rapidamente para a condição de eco de estado, onde alcança a melhor capacidade de resolver problemas (Jaeger, 2001; Pathak *et al.*, 2018; Lukoševičius, 2012). Inicialmente o número de neurônios N, foi escolhido pequeno com o intuito de avaliar a resposta da RCR com poucos neurônios. Para os outros HP (Lr, Rc\_con, lss, Ridge e Seed) foram escolhidos valores iniciais arbitrários, pois, em geral, a literatura não aponta valores especiais para esses HP, que devem ser obtidos conforme a tarefa que a RCR deve executar (Lukoševičius, 2012).

Tabela 4 – Hiperparâmetros HP01 para gerar a RCR inicial

Parâmetro	Valores
N	30
Lr	0,5
Sr	0,9
lss	1,0
Rc_con	0,1
Seed	10000
Activ	$\tanh$
Noise_rc	0,0
Ridge	1e-7

Fonte: Autor, 2023.

No teste inicial, serão avaliadas quatro condições de treinamento, *online* com *loopback* (ON\_CL), *online* sem *loopback* (ON\_SL), *offline* com *loopback* (OF\_CL), *offline* sem *loopback* (OF\_SL) (Racca e Magri, 2022; Trouvain *et al.*, 2020), conforme detalhes abaixo:

- Online (ON) - Neste modo, a RCR é treinada somente com informação local. Este método permite atualizar o modelo um passo por vez. A coleta e o treinamento podem ocorrer de forma simultânea;

- Offline (OF) - Previsões onde a RCR é treinada a partir de um conjunto completo de informação, isto é, entradas e suas respectivas saídas formam um conjunto completo. Uma vez que a aprendizagem é concluída, o modelo será alterado somente se o treinamento ocorrer em todo o conjunto de dados novamente. A coleta e o treinamento ocorrem em fases separadas;
- Com loopback (CL) - A RCR recebe na entrada uma das variáveis caóticas, ou o  $|\mathbf{X}_\perp|$  somente durante o treinamento. Na fase de previsão o sinal de entrada é retirado e o sinal da saída é conectado à entrada para que a RCR, oscilando de forma autônoma, produza as previsões;
- Sem loopback (SL) - Quando a RCR recebe na entrada uma ou mais variáveis e aprende a produzir na saída as mesmas variáveis, adiantadas alguns intervalos de tempo à frente, sem conectar o sinal da saída à entrada. Nesta condição, o sinal do sistema externo deve ser inserido constantemente na RCR.

Antes de ser aplicado à RCR, o sinal externo proveniente da simulação do sistema GB irá passar por uma normalização, equação 4.1, assim o conjunto de pontos terá média 0 e variância unitária. Este ajuste é realizado visando facilitar a comparação do desempenho da RCR para diferentes sinais (Pyragas e Pyragas, 2020). Importante notar que este passo não é obrigatório para o funcionamento da RCR.

$$u_i(t) = \frac{[s_i(t) - \langle s_i(t) \rangle]}{\sigma_i}, i = 1, \dots, d. \quad (4.1)$$

Cada teste com a RCR-HP01 consiste em utilizar 3000 pontos para treino e mais 2000 pontos para avaliar o treinamento da RCR. Além disso, será realizado nas condições ON\_CL, ON\_SL, OF\_CL e OF\_SL para os intervalos de previsão de 0,5; 1; 1,5; 2; 2,5; 3; 4; 5; 10; 15 e 20TL nas variáveis ou conjuntos de variáveis  $V_{1d}$ ,  $V_{2d}$ ,  $I_d$ ,  $\mathbf{V}_{1d\_V_{2d}}$ ,  $\mathbf{V}_{1d\_I_d}$ ,  $\mathbf{V}_{2d\_I_d}$ ,  $\mathbf{V}_{1d\_V_{2d\_I_d}}$  e '**Todas**'. O símbolo de *underline* (  ) entre as variáveis significa que aquelas variáveis serão colocadas em conjunto na entrada da RCR, por exemplo,  $\mathbf{V}_{1d\_V_{2d}}$  indica que serão colocadas simultaneamente as variáveis  $V_{1d}$  e  $V_{2d}$  na entrada da RCR. A combinação '**Todas**' corresponde às seis variáveis, três do circuito GB diretor e três do circuito GB resposta aplicadas simultaneamente na RCR.

Os testes com a RCR-HP01 visam identificar a condição e o intervalo de previsão onde a RCR apresenta o erro médio e o pior erro. A razão para isso é identificar dois pontos para intencionalmente testar se a RCR, após otimização de parâmetros, consegue melhorar os resultados. A qualidade da previsão obtida na RCR será avaliada a partir das métricas  $\mathbf{R}^2$  alto, **NRMSE** baixo e quantos TL a previsão antecipou.

Buscando um compromisso entre o intervalo de previsão maior com melhor precisão, dos testes com HP01, devem ser obtidas as informações sobre a melhor

condição de teste (ON\_CL, ON\_SL, OF\_CL e OF\_SL) e os intervalos de previsão com precisão média e pior precisão. Estas informações serão utilizadas no próximo passo que será a busca por HP02, teste 01, na figura 15. Esta busca será realizada por meio da biblioteca *hyperopt* (Bergstra *et al.*, 2013) rodando o algoritmo de busca de parâmetros estimador Parzen estruturado em árvore ou *Tree-structured Parzen Estimator* (TPE) em ambiente *Python*. A biblioteca *hyperopt* executa um algoritmo que minimiza o erro de uma função seguindo, ou a estratégia de busca randômica, ou a estratégia de busca TPE. Na randômica, os parâmetros são escolhidos dentro de uma faixa pré-definida, onde são testados vários pontos de forma aleatória. Na estratégia TPE, a escolha dos parâmetros também é realizada em uma faixa pré-definida, porém a seleção do próximo parâmetro é função da aplicação do teorema de Bayes ao resultado anterior, que permite uma convergência mais acelerada para o parâmetro ótimo. Para a busca de HP02 serão produzidas 500 combinações diferentes de HP, onde *Rc\_con* e *Activ* serão mantidos constantes e iguais aos de HP01 e os sete parâmetros restantes irão variar todos ao mesmo tempo, olhar tabela 4. Este teste será aplicado às variáveis  $V_{1d}$  e  $V_{2d}$ , pois no estudo de Gauthier-Bienfang (Gauthier e Bienfang, 1996) foi observado um comportamento distinto nessas variáveis, o sistema DR sincronizava com  $V_{1d}$ , mas não com  $V_{2d}$ , indicando que estas duas variáveis apresentam alguma distinção em suas dinâmicas.

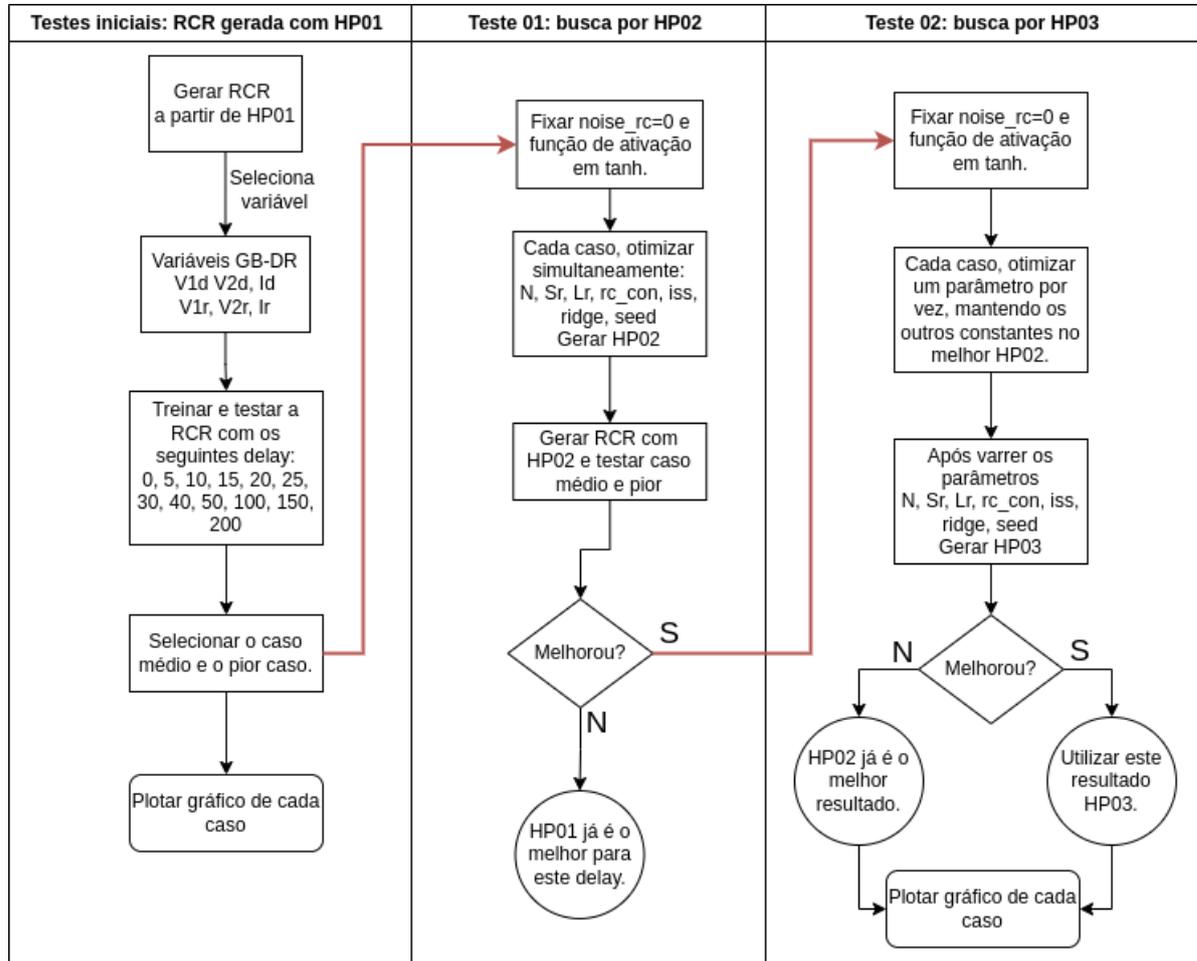
Importante mencionar que a otimização será realizada em um conjunto pequeno de dados, tendo em consideração as seguintes premissas, primeiro, a literatura aponta que os HP obtidos em uma pequena amostra dos dados do observável, em geral, funcionam de forma adequada para um conjunto maior de dados (Lukoševičius, 2012); segundo, testes em pequena escala permitem quantidade de ciclos superiores àqueles realizados em grande volume de dados, permitindo acelerar a obtenção dos HP.

O melhor HP obtido na busca de HP02, teste 01 na Figura 15, será refinado em uma nova otimização, a busca por HP03, teste 02 na Figura 15, onde o *Rc\_con* e a *Activ* seguem iguais aos de HP01, enquanto os outros HP irão variar 500 vezes, porém um de cada vez, de tal forma que o melhor resultado entre os 500 de um parâmetro, será utilizado com valor fixo na otimização do próximo HP, sucessivamente.

Concluída esta etapa, a qualidade do HP03 obtido será avaliada em relação ao  $R^2$  alto e **NRMSE** baixo. Novamente, os testes nos intervalos de previsão iniciais, olhar tabela 2, serão realizados, porém, utilizando HP03 para avaliar se houve melhora tanto no intervalo de previsão onde ocorreu a otimização quanto nos outros intervalos.

Também será realizado o treinamento da RCR construída com HP03, utilizando um conjunto maior de dados,  $10^5$  pontos e variando as proporções de dados, entre treinamento e teste, nas proporções percentuais 10/90, 20/80, 30/70, 40/60, 50/50, 60/40, onde o primeiro número é a quantidade de dados de treinamento e o segundo a quantidade utilizada para validar o treinamento. Este teste visa avaliar a influência da

Figura 15 – Fluxograma dos passos que serão executados pelo algoritmo de otimização dos HP. Partindo dos testes realizados com a RCR-HP01, o caso médio e o pior caso são selecionados para o teste 01, otimizações e busca de HP02, que consiste em obter e testar 500 RCR construídas com 500 combinações de HP diferentes, variando todos os HP ao mesmo tempo, em cada combinação. Partindo do HP02 obtido, o teste 02 irá produzir 3500 RCR diferentes, gerando 500 combinações por HP, variando um HP por vez e mantendo o melhor resultado para o próximo HP.



Fonte: Autor, 2023

quantidade de treinamento na capacidade da RCR de realizar as previsões.

Concluídos os testes com as séries caóticas, o próximo passo será treinar e testar a RCR para previsão das dessincronizações.

#### 4.2.2 Hiperparâmetros para Previsão das Dessincronizações

Para obter uma medida adequada das dessincronizações do sistema DR acoplado em  $V_2$ , foi utilizada uma nova variável,  $\|\mathbf{X}_\perp\|$ , equação 4.2, onde cada termo da equação corresponde a um vetor, obtido a partir da simulação do sistema DR, seção 4.1. Quando esta variável se aproxima de 0, indica que os dois circuitos GB estão sincronizados, caso o valor seja diferente, indica a ocorrência de uma dessincronização.

$$\|\mathbf{X}_\perp\| = \sqrt{(\mathbf{V}_{1d} - \mathbf{V}_{1r})^2 + (\mathbf{V}_{2d} - \mathbf{V}_{2r})^2 + (\mathbf{I}_d - \mathbf{I}_r)^2}. \quad (4.2)$$

Seguindo a mesma estratégia adotada na previsão de séries caóticas, ver figura 15, deve ser construída uma RCR com HP01, na qual são realizados testes de previsão nas quatro condições de treinamento, ON\_CL, ON\_SL, OF\_CL e OF\_SL.

Para as dessincronizações, cada teste com HP01 consistirá em utilizar 17000 pontos para treino e mais 11000 para validar o treinamento da RCR. A quantidade de pontos para os testes com  $\|X_{\perp}\|$  foi superior a utilizada com as variáveis caóticas porque na faixa de pontos que vão até 30000, já são observados alguns eventos extremos. Além disso, cada teste será realizado nos intervalos de previsão 0; 0,1; 0,2; 0,3; 0,4; 0,5; 0,6; 0,7; 0,8; 0,9; 1; 1,5; 2; 2,5; 3; 3,5; 4; 5; 5,5; 6; 10; 15 e 20TL nas variáveis ou conjunto de variáveis  $V_{1d}$ ,  $V_{2d}$ ,  $I_d$ ,  $V_{1d\_V_{2d}}$ ,  $V_{1d\_I_d}$ ,  $V_{2d\_I_d}$ ,  $V_{1d\_V_{2d\_I_d}}$  e **Todas**.

A partir do primeiro bloco de testes com HP01, devem ser selecionadas a melhor condição de teste, as variáveis ou combinações e as duas condições, média e pior precisão, para realizar as buscas de HP02 e HP03. Para cada variável selecionada, na primeira busca por HP02, serão realizados 500 testes, com todos os parâmetros variando, ao mesmo tempo, em cada teste, e na busca por HP03, serão realizados 500 testes por parâmetro, num total de 3500 testes. Estas otimizações também serão realizadas com o algoritmo TPE, executado na biblioteca hyperopt (Bergstra *et al.*, 2013) com as RCR geradas no RPY.

Uma vez obtidos os HP03, com o intuito de avaliar estes HP, serão realizadas previsões nos mesmos intervalos de previsão utilizados com HP01. Além de avaliar a melhora no intervalo da otimização, o objetivo é avaliar se houve melhora nos outros intervalos.

Após estes passos, o conjunto HP03 obtido será utilizado para gerar a RCR que será treinada para realizar as previsões dos eventos extremos.

#### 4.2.3 Previsões dos Eventos Extremos do Sistema DR Acoplado em $V_2$

Utilizando os melhores HP obtidos na seção 4.2.2, será produzida uma RCR para realizar os testes de previsão dos eventos extremos do sistema DR. Note que os eventos extremos são os maiores valores que o  $\|X_{\perp}\|$  alcança, porém é necessário estabelecer uma amplitude onde a dessincronização será considerada um evento extremo. Para o sistema DR será produzida a função de densidade de probabilidade a partir dos dados obtidos por simulação numérica, que permitirão a identificação da amplitude onde um evento de dessincronização será considerado extremo. Tendo a amplitude definida, durante as previsões, será utilizado um segundo programa para marcar os pontos, amplitude e tempo de antecipação em TL, onde a RCR identificar a ocorrência dos eventos extremos. Estes dados serão guardados em um banco de dados para análise posterior.

Os testes de previsão de eventos extremos serão realizados em dois conjuntos de dados, o primeiro com 30000 pontos, onde 17000 serão utilizados para treinamento e 13000 para previsão. O segundo conjunto terá 100000 pontos e será realizado o treinamento com 40000 pontos iniciais e 60000 para previsão. Nos dois conjuntos de dados, a qualidade da previsão será avaliada através do  $R^2$ , do NRMSE e buscando um TL de previsão longo. Além disso, também será avaliada a precisão em relação ao tempo e a amplitude dos eventos extremos previstos.

Com este passo, a proposta de uso da RCR otimizada para prever séries caóticas, dessincronizações e eventos extremos é concluída. No próximo capítulo, serão apresentados os resultados da implementação desta proposta.

## 5 RESULTADOS E DISCUSSÃO

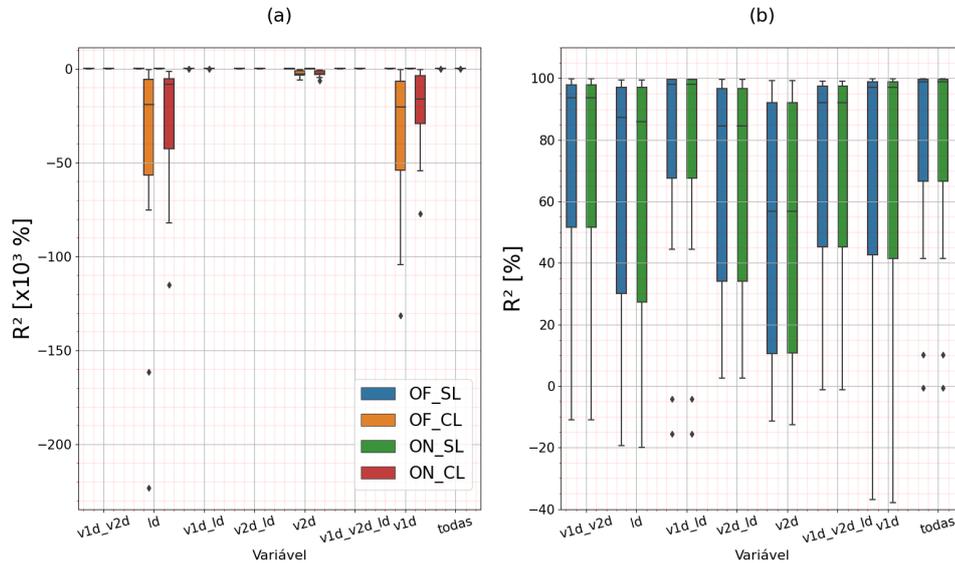
Neste capítulo estão os resultados da otimização dos hiperparâmetros (HP) das RCRs e àqueles relacionados com a qualidade das previsões obtidas com as séries caóticas e as dessincronizações de  $\|\mathbf{X}_\perp\|$  do sistema DR quando acoplado em  $V_2$ .

### 5.1 PREVISÕES DAS SÉRIES CAÓTICAS

Inicialmente foi realizada uma triagem utilizando a RCR construída com os parâmetros HP01, que estão na tabela 4 da seção 4.2.1, sob quatro condições diferentes, *online* com *loopback* (ON\_CL) e sem *loopback* (ON\_SL), *offline* com *loopback* (OF\_CL) e sem *loopback* (OF\_SL), ver Figura 16. Nesta triagem foi observado que a previsão obtida pela RCR com *loopback* (ON\_CL e OF\_CL) tem qualidade inferior à condição sem *loopback* (ON\_SL e OF\_SL). Com *loopback* em nenhuma variável foi obtido  $R^2$  positivo, olhar diagrama de caixa (*boxplot*) nas cores laranja e vermelho da Figura 16(a). Uma hipótese para explicar este resultado está relacionado com o objeto de aprendizagem da RCR, o sistema caótico. A teoria indica que em sistemas caóticos não é possível realizar previsões muito longas devido a sua típica divergência exponencial no espaço de fases (Strogatz, 2018). Assim, quando a RCR é colocada para funcionar de forma autônoma, ao reproduzir a dinâmica do sistema caótico, uma pequena divergência em suas condições iniciais modifica completamente a natureza do sinal obtido em um tempo mais longo. Assim, as condições ON\_CL e OF\_CL não foram otimizadas nas fases de busca de HP02 e HP03.

Ainda na triagem inicial, foram comparados os modos *online* e *offline* da condição sem *loopback*, ver Figuras 16(b) e 17. Neste caso, as previsões nos dois modos não apresentaram grandes diferenças. A justificativa para esse comportamento é que o conjunto de pontos de treinamento utilizado foi igual para os dois modos. Assim, após o treinamento, a RCR convergiu para o mesmo resultado nos dois modos. Entretanto, foi observado que no modo *online*, a demanda de processamento computacional alcançou níveis próximos de 100%, tornando lenta a busca pelos hiperparâmetros (HP), e em alguns casos causando travamento no computador onde o algoritmo foi executado. A explicação para esta demanda computacional decorre do processo de inversão da matriz  $(\mathbf{X}\mathbf{X}^T + \beta\mathbf{I})^{-1}$ , que é um processo com alto custo computacional e no caso do modo *online* deve ocorrer para cada ponto de treinamento, olhar seção 3.4.3 do capítulo 3 para mais detalhes. No modo *offline*, a inversão da matriz  $(\mathbf{X}\mathbf{X}^T + \beta\mathbf{I})^{-1}$  ocorre uma única vez para todo o conjunto de dados ao mesmo tempo, o que justificaria a demanda computacional menor. Devido a esta questão do processamento e também ao fato de que os testes nos dois modos apresentaram valores similares, os experimentos subsequentes foram realizados no modo OF\_SL.

Figura 16 – Diagramas de caixa (*boxplot*) dos resultados das previsões iniciais obtidas com a RCR produzida com HP01. Na cor vermelha, *online com loopback* (ON\_CL), na cor verde, *online sem loopback* (ON\_SL), na cor laranja, *offline com loopback* (OF\_CL) e na cor azul, *offline sem loopback* (OF\_SL). Em (a) estão destacados os resultados para os testes com *loopback*, cores vermelha e laranja, observe que estes resultados são todos negativos. Em (b) estão ampliados os resultados para os testes sem *loopback*, cores verde e azul, note a similaridade *online* e *offline*.



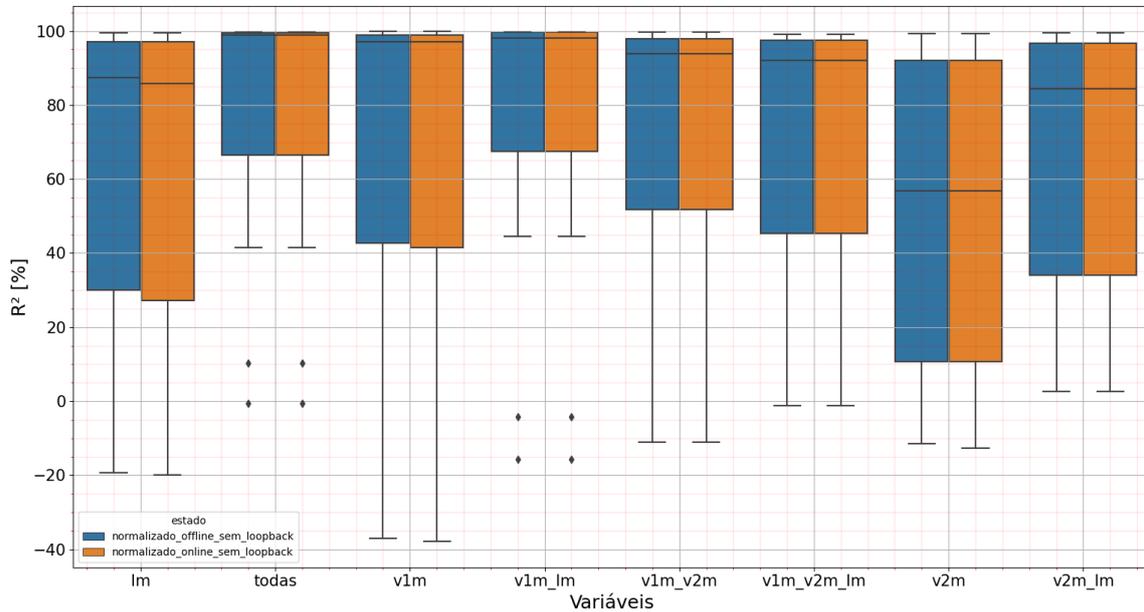
Fonte: Autor, 2023

No modo *offline sem loopback*, Figuras 18 e 19, com o intervalo de previsão abaixo ou igual a  $1,5TL$ , quaisquer variáveis ou combinações previstas pela RCR-HP01 apresentaram um  $R^2$  acima de 85%. O primeiro aspecto deste resultado está relacionado com o intervalo de previsão, pois para um  $TL$  pequeno é esperado obter boas previsões em sistemas caóticos. O segundo aspecto deste resultado tem relação com a escolha de HP01, que no intervalo de previsão, permitiu que a RCR produzisse uma boa aproximação da dinâmica das variáveis caóticas.

O melhor resultado foi obtido quando a RCR-HP01 realizou a previsão tendo como entrada as variáveis dos circuitos diretor e resposta ao mesmo tempo, denominado *todas* nas Figuras 18 e 19. Nestas condições, com um intervalo de previsão até  $5TL$ , apresentou um  $R^2$  superior a 90%. Uma interpretação para este resultado é que o conjunto de variáveis *todas* contém as seis variáveis que fornecem a informação completa sobre dinâmica do sistema DR, então tendo esta informação completa, a RCR alcança um aprendizado mais preciso da dinâmica deste sistema. Um segundo aspecto é a combinação HP01, que neste caso permitiu a construção de uma RCR eficiente para aprender a dinâmica na faixa considerada.

Para as variáveis isoladas, o melhor resultado é observado em  $V_{1d}$ , que com o intervalo de previsão até  $5TL$ , alcança um  $R^2$  acima de 85%. Novamente, a escolha de HP01 favoreceu a RCR capturar a dinâmica da variável na faixa considerada. Já para intervalos longos, maiores ou iguais a  $10TL$ , todas as variáveis apresentaram um  $R^2$  abaixo de 50%. Este resultado pode ser consequência da natureza caótica do

Figura 17 – Diagramas de caixa dos resultados das previsões iniciais obtidas pela RCR produzida com HP01. Nesta figura são apresentados somente os modos *offline sem loopback* OF\_SL (cor laranja) e *online sem loopback* ON\_SL (cor azul). Observe a grande similaridade nos resultados para os dois modos.



Fonte: Autor, 2023

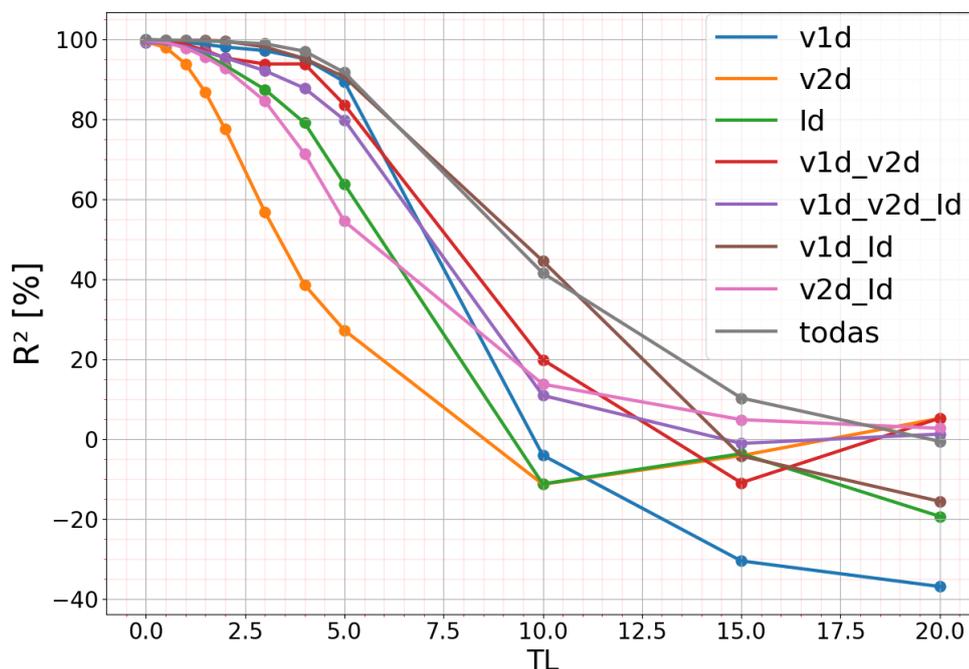
fenômeno a ser aprendido pela RCR, que diverge rapidamente para TL longo e também pode ser consequência da escolha de HP01, ou seja, espera-se que otimizando HP01 pode-se melhorar o resultado nas faixas mais longas. Também chama atenção, o comportamento com  $V_{2d}$ , pois nesta variável a RCR-HP01 apresentou o decréscimo mais acelerado de  $R^2$ , ficando abaixo de 80% em um intervalo maior ou igual a 2TL. Uma justificativa para este resultado é o fato de  $V_{1d}$  e  $V_{2d}$  apresentarem dinâmicas diferentes na faixa de teste, principalmente em relação à amplitude dos sinais, olhar Figura 20(a) e (b).

Nas próximas subseções estão alguns resultados específicos das variáveis caóticas do circuito diretor,  $V_{1d}$  e  $V_{2d}$ . Para ambas variáveis foi aplicado o fluxograma de testes da Figura 15 localizado na seção 4.2.1. Todos os testes consistiram em colocar a variável na entrada da RCR, que foi treinada para prever esta a mesma variável alguns TL à frente na saída. Todas as previsões foram realizadas em um conjunto diferente daqueles utilizados no treinamento.

### 5.1.1 Resultados para $V_{1d}$

Para  $V_{1d}$ , os resultados com HP01, traço de cor azul nas Figuras 18 e 19, apontam que para intervalos de previsão menores ou iguais a 5TL, o  $R^2$  é maior ou igual a 90%. O que sugere que a RCR construída com HP01 capturou de forma ótima a dinâmica de  $V_{1d}$  nesta faixa. Tendo em vista o enfoque em melhorar as previsões para intervalos mais longos, os intervalos de previsão 10TL, com  $R^2$  próximo a -5%, e 20TL

Figura 18 – Valores de  $R^2$  para as previsões da RCR-HP01 ao variar TL no modo *offline* sem *loopback* para as diversas combinações de variáveis.

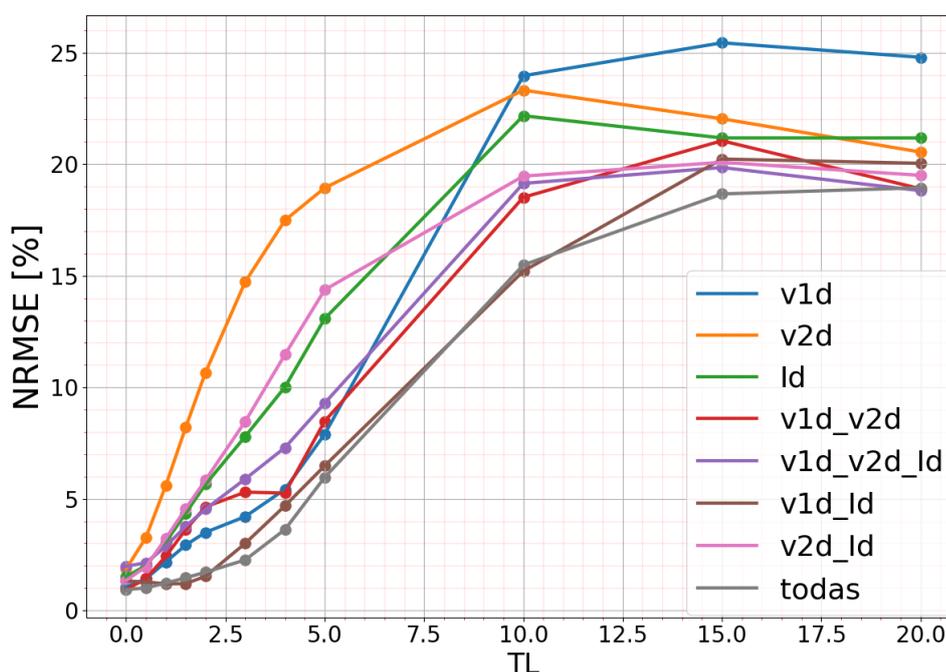


Fonte: Autor, 2023

com  $R^2$  de -36% foram selecionados para a fase de busca do HP02 e HP03. Cabe observar que valores negativos de  $R^2$  indicam que a previsão não apresenta qualquer relação com o sinal esperado. Portanto, espera-se que otimizar os HP da RCR corrija este problema.

A primeira otimização, busca por HP02, não resultou em melhoria dos parâmetros nos dois intervalos, portanto o algoritmo TPE não conseguiu encontrar o melhor conjunto de HP somente com 500 testes, variando todos os HP ao mesmo tempo. A alternativa para resolver este problema é aumentar o número de testes até alcançar uma convergência ou modificar a natureza dos testes. Optou-se pela segunda alternativa, ver árvore de decisões na coluna do teste 01 da Figura 15 localizado na seção 4.2.1. Assim, a segunda otimização, que corresponde a busca por HP03, foi realizada. A lógica por trás da segunda otimização é a seguinte, suponha, por exemplo, que a busca se iniciou por  $Lr$ , assim, foram realizados 500 testes variando somente este parâmetro selecionado e mantendo os outros fixos. Após encontrar o melhor conjunto entre os 500 testes neste parâmetro, o algoritmo fixa o  $Lr$  do melhor conjunto e passa a realizar mais 500 testes com o próximo parâmetro. Este procedimento é realizado sete vezes, resultando num total de 3500 testes (500 testes/parâmetro x 7 parâmetros). Esta segunda otimização se mostrou mais eficiente, pois produziu uma melhoria significativa no intervalo de previsão (IP) de 10TL, o  $R^2$  foi de -5% para 81,6%, embora o  $NRMSE$  tenha piorado, foi de 24% para 34,7%. Observe que a previsão obtida com a RCR-HP01 estava completamente diferente do sinal correto, Figura 21(a), porém com a RCR-HP03

Figura 19 – Valores de NRMSE para as previsões da RCR-HP01 ao variar TL no modo offline.



Fonte: Autor, 2023

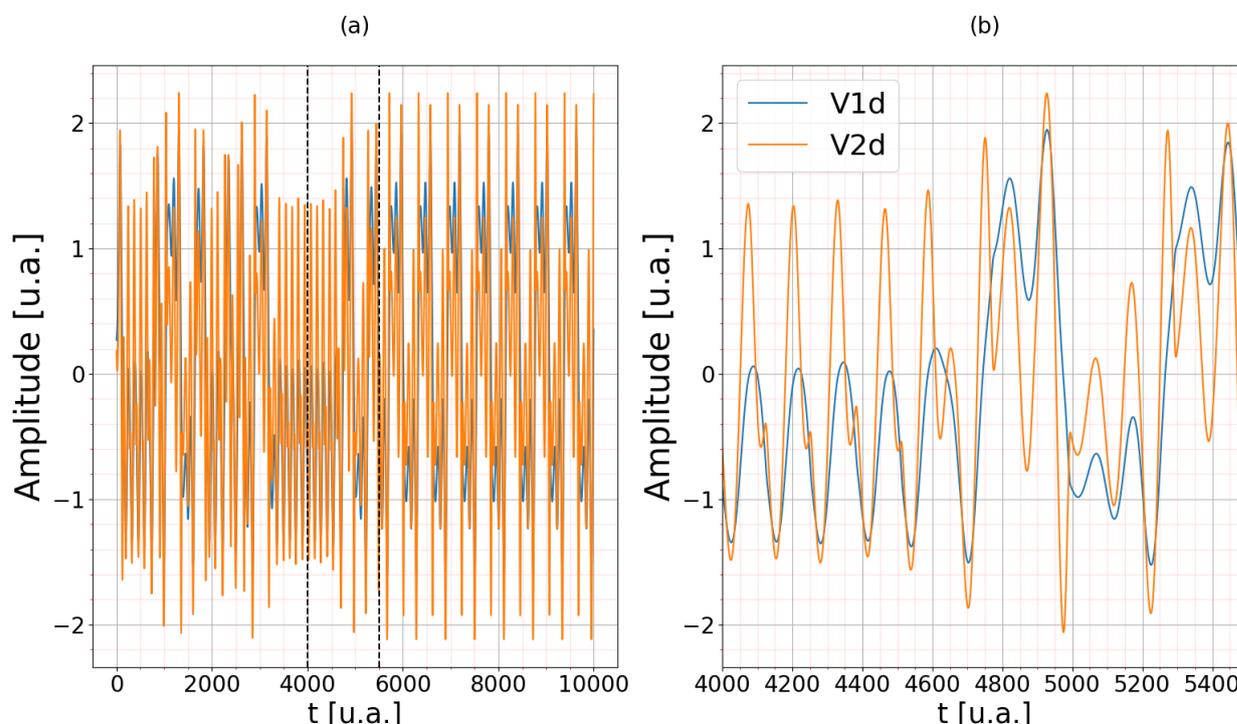
houve uma aproximação melhor, Figura 21(b), ainda assim, a RCR-HP03 produziu alguns erros em amplitude e frequência. Uma hipótese para explicar porque estes erros ainda ocorreram está relacionada com a quantidade de testes de otimização. Espera-se que ao aumentar a quantidade de testes utilizando o método de HP03, as melhorias devam continuar avançando.

Na tabela 5 estão os resultados da otimização para cada parâmetro da RCR no IP 10TL. Cabe destacar três pontos, o aumento do número de neurônios, que foram de 30 para 136, está relacionado com a complexidade do sinal caótico, que exigiu maior capacidade de processamento da RCR. O segundo ponto é o raio espectral ( $Sr$ ) igual a 1,358, indicando que para prever  $V_{1d}$ , a condição de eco de estados, necessária para o correto funcionamento da RCR (Jaeger, 2001; Lukoševičius, 2012) é obtida com uma matriz  $\mathbf{W}$  com  $Sr > 1$ , portanto com pesos maiores que aqueles sugeridos na literatura (Nakajima, 2021; Chowdhury *et al.*, 2022). O terceiro é a densidade das conexões do reservatório, o  $Rc\_con$ , que foi reduzido pela metade em HP03, indicando que uma matriz  $\mathbf{W}$  mais dispersa foi mais adequada para capturar a dinâmica de  $V_{1d}$ .

Em relação aos resultados para o IP 20TL, olhar tabela 6, mesmo com o aumento do número de neurônios, a RCR-HP03 não produziu um  $R^2 > 0$ . A justificativa para este resultado é a complexidade de se antecipar uma série caótica em um intervalo de 20 tempos de Lyapunov. Apesar disso, é possível que uma quantidade maior de tentativas durante a otimização com o método de HP03 resulte em convergência para este IP.

Comparando as previsões de  $V_{1d}$  obtidos com a RCR-HP01, RCR-HP03 de 10TL

Figura 20 – Comparação entre  $V_{1d}$ , traço na cor azul, e  $V_{2d}$ , traço na cor laranja, do circuito diretor do sistema DR. Em (a) estão os sinais completos de toda a faixa de teste de previsão para as duas tensões. Em (b) está uma ampliação dos dois sinais para melhor visualização, note a diferença de amplitude entre as tensões. A hipótese é que  $V_{2d}$  possui amplitudes maiores, as quais a RCR produzida com HP01 não conseguiu aprender.



Fonte: Autor, 2024

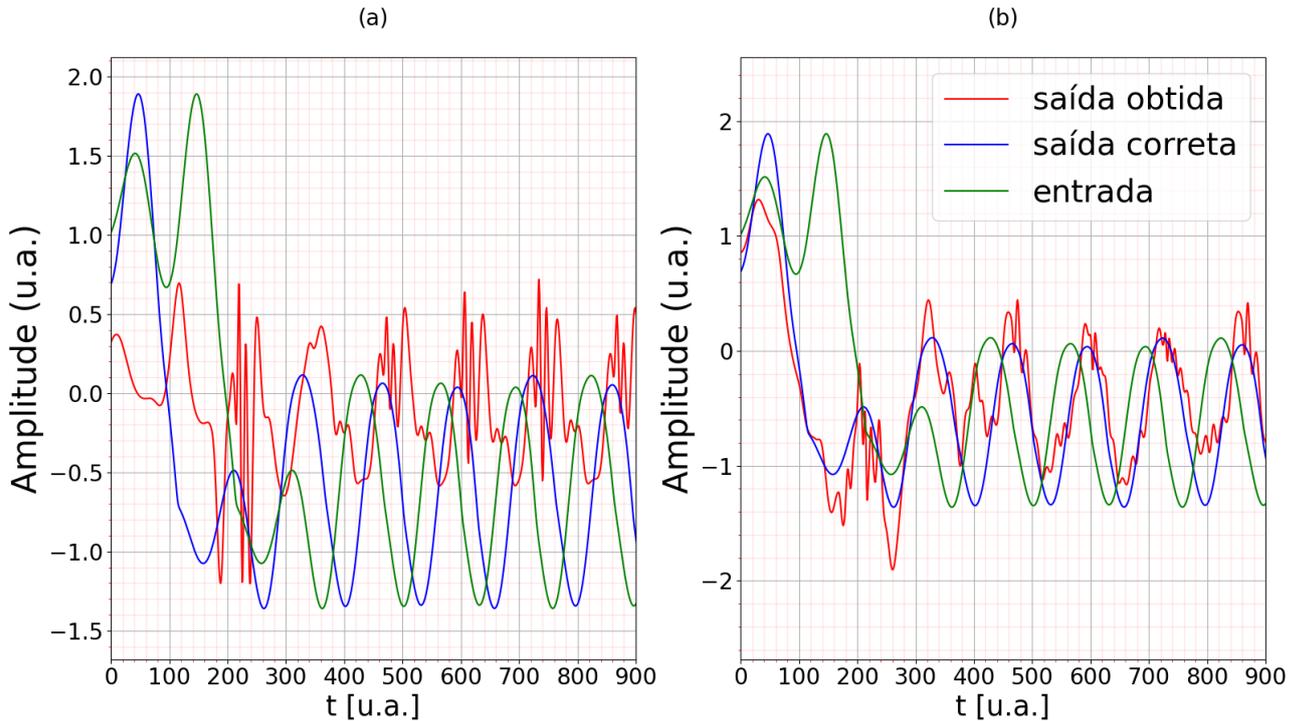
e RCR-HP03 de 20TL, Figura 22, testadas em IPs diferentes, nota-se que a RCR-HP03 de 10TL, traço na cor azul, apresentou resultados melhores para todo  $IP < 20TL$ . A explicação para este resultado é que otimizar os HP da RCR em um IP, neste caso, em 10 TL, possibilitou obter resultados melhores em IPs diferentes.

### 5.1.2 Resultados para $V_{2d}$

Em relação à variável  $V_{2d}$ , nos testes com a RCR-HP01, observar o traço laranja nas Figuras 18 e 19, foram selecionados os IP de 3TL e 20TL para otimização, pois no primeiro o  $R^2$  está próximo de 50% e o  $NRMSE$  de 15%, caracterizando o IP de erro médio desta variável, e no segundo, se observa o pior erro, com  $R^2$  de 5% e um  $NRMSE$  próximo de 21%.

Na tabela 7 estão os HP obtidos com as otimizações realizadas no intervalo de 3TL e as medidas de qualidade  $R^2$  e  $NRMSE$  das previsões obtidas com as RCR utilizando estes HP. Os resultados de HP02 e HP03 ficaram ambos acima de 90%. A explicação para esses resultados está relacionada com o aumento da capacidade de processamento da RCR, decorrente do incremento no número de neurônios e também com as reduções aplicadas em  $L_r$ ,  $S_r$  e  $I_{ss}$ , pois o  $L_r$ , fator de amostragem temporal, menor ajustou a RCR para uma frequência maior, o  $S_r$  menor implicou em uma matriz

Figura 21 – Comparação entre as previsões obtidas para  $V_{1d}$  com (a) RCR-HP01 e (b) RCR-HP03 otimizada em 10TL. Note que em (b), a RCR-HP03 produz uma previsão, traço na cor vermelha, que está mais próxima da saída correta, traço na cor azul, porém ainda apresenta divergências.



Fonte: Autor, 2024

Tabela 5 – Comparação entre HP01, HP02 e HP03 obtidos na otimização onde a entrada da RCR é a variável  $V_{1d}$  e a saída é treinada para antecipar a variável  $V_{1d}$  em um IP de 10TL

Parâmetro	HP01	HP02	HP03
N	30	49	136
Lr	0,5000	0,6060	0,4737
Sr	0,9000	1,502	1,358
lss	1,000	2,005	0,8856
Rc_con	0,1000	0,0427	0,0552
Seed	10000	47	47
Activ	tanh	tanh	tanh
Noise_rc	0	0	0
Ridge	1,000e-7	1,744e-06	2,235e-09
$R^2$	-5%	-6%	81,6%
NRMSE	24%	24%	34,7%

Fonte: Autor, 2023.

**W** com valores de pesos menores, aumentando a influência do sinal anterior na saída atual do reservatório, olhar equação 3.3 na seção 3.4, e o lss reduziu para a metade o sinal de entrada utilizado pela RCR, neste caso, a hipótese é que as amplitudes de  $V_{2d}$  deveriam estar saturando os valores da RCR, com a a redução de lss, essa saturação foi evitada.

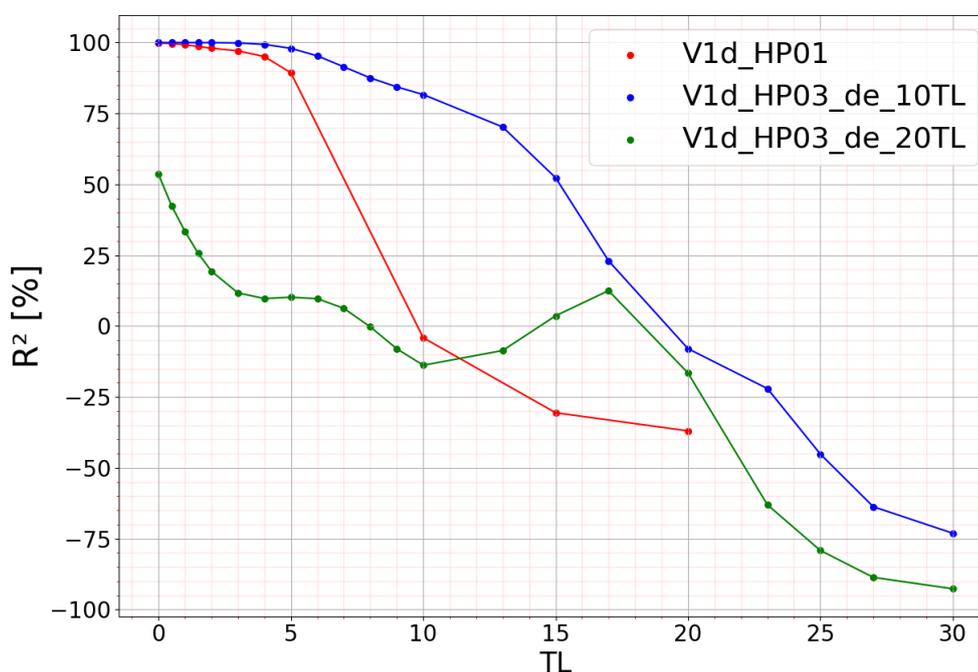
Na Figura 23 estão amostras dos resultados das previsões temporais de  $V_{2d}$  no

Tabela 6 – Comparação entre HP01, HP02 e HP03 obtidos na otimização onde a entrada da RCR é a variável  $V_{1d}$  e a saída é treinada para antecipar a variável  $V_{1d}$  em um IP de 20TL

Parâmetro	HP01	HP02	HP03
N	30	110	200
Lr	0,5000	3,03e-07	0,0086
Sr	0,9000	2,557e-05	0,0010
lss	1,000	4,504	9,682
Rc_con	0,1000	7,622e-09	0,0081
Seed	10000	89	28
Activ	tanh	tanh	tanh
Noise_rc	0	0	0
Ridge	1,000e-7	3,845e-08	1,001e-09
$R^2$	-36%	-35%	-10,6%
NRMSE	25%	24,6%	22,2%

Fonte: Autor, 2023.

Figura 22 – Comparação dos valores de  $R^2$  obtidos nas previsões da série caótica de  $V_{1d}$  utilizando a RCR-HP01, traço vermelho, a RCR-HP03 otimizada no intervalo de 10TL, traço na cor azul e a RCR-HP03 otimizada no intervalo de 20TL, traço na cor verde. Comparando o traço vermelho com o traço azul, observe que a RCR-HP03, que foi obtida em uma otimização no IP de 10TL, melhorou os valores de  $R^2$  em todos os outros IP. Já a RCR-HP03 de 20TL, que não convergiu, apresentou resultados piores que a RCR-HP01 na maioria dos IP.



Fonte: Autor, 2023

IP de 3TL, traços na cor vermelha. Utilizando HP01, Figura 23(a), pode-se observar uma previsão ainda ruidosa decorrente dos HP não ajustados. Na Figura 23(b), observe a melhoria na frequência e amplitude prevista com a RCR-HP02, onde foram aplicadas as reduções de Lr, Sr e lss. Com HP03, Figura 23(c), o ajuste fino produziu um aumento da dispersão dos neurônios do reservatório, obtido com a redução do HP rc\_con, que

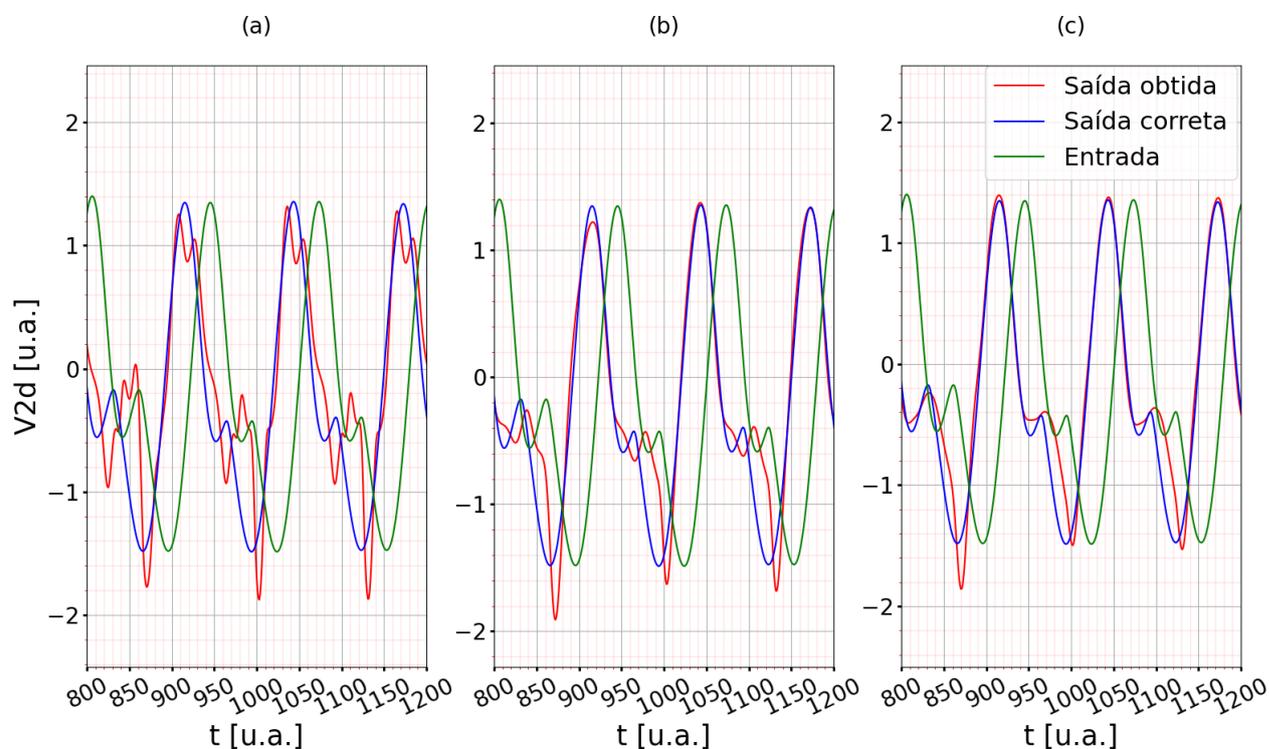
resultou em maior precisão tanto na amplitude quanto na frequência.

Tabela 7 – Comparação entre HP01, HP02 e HP03 obtidos na otimização onde a entrada da RCR é a variável  $V_{2d}$  e a saída é treinada para antecipar a variável  $V_{2d}$  em um IP de 3TL

Parâmetro	HP01	HP02	HP03
N	30	124	200
Lr	0,5000	0,1690	0,1376
Sr	0,9000	0,6623	0,7138
lss	1,000	0,5098	0,5008
Rc_con	0,1000	0,4558	0,1264
Seed	10000	89	89
Activ	tanh	tanh	tanh
Noise_rc	0	0	0
Ridge	1,000e-7	6,970e-09	7,357e-09
$R^2$	56%	92%	95%
NRMSE	15%	27%	23%

Fonte: Autor, 2023.

Figura 23 – Amostras das previsões de  $V_{2d}$  no IP de 3TL utilizando (a) RCR-HP01 sem ajuste de HP, neste caso, a comparação entre saída obtida, traço na cor vermelha e saída correta, traço na cor azul, gerou um  $R^2 = 56\%$ , (b) RCR-HP02, com HP N, Lr, Sr e lss reduzidos após ajuste, a mesma comparação produziu  $R^2 = 92\%$  e (c) RCR-HP03, que reduziu rc\_con aumentando a dispersão do reservatório, a mesma comparação produziu  $R^2 = 95\%$ .



Fonte: Autor, 2024

Em relação ao IP de 20TL os resultados das otimizações dos HP estão organizados na tabela 8. A otimização não encontrou uma combinação de HP que permitisse boas previsões, a razão disto é que o  $R^2$  melhorou, mas, o  $NRMSE$  piorou, alcançando

um valor de 82% com HP03. Há ao menos uma hipótese para explicar este resultado e está relacionada com a natureza do sinal caótico. Em um intervalo longo de 20TL, sinais caóticos apresentam alto grau de complexidade e divergência, os quais não foram capturados pela RCR nas condições de otimização e teste utilizados. Um caminho para encontrar a solução pode estar nas características da otimização, espera-se que aumentando a quantidade de testes de otimização e a faixa de valores possíveis para a quantidade de neurônios, primeiro permitirá ao algoritmo de otimização varrer uma faixa de HP superior e segundo, haverá um incremento dos níveis de processamento possíveis, favorecendo a busca por uma região de HP onde a RCR converge com melhor qualidade para a saída correta.

Utilizando a RCR-HP01, Figura 24(a), a saída obtida pela RCR, , sinal na cor vermelha, apresenta pouca relação com a saída correta,  $R^2$  de 5%. Utilizando a RCR-HP02, Figura 24(b), a RCR melhora a amplitude da previsão, porém seu erro  $NRMSE$  escala para 79%. A Figura 24(c) sugere um sinal menos ruidoso, na frequência correta, porém ainda necessitando de correções na amplitude. Supõe-se que utilizar maior quantidade de neurônios e executar uma quantidade maior de testes deve levar a RCR a produzir resultados melhores.

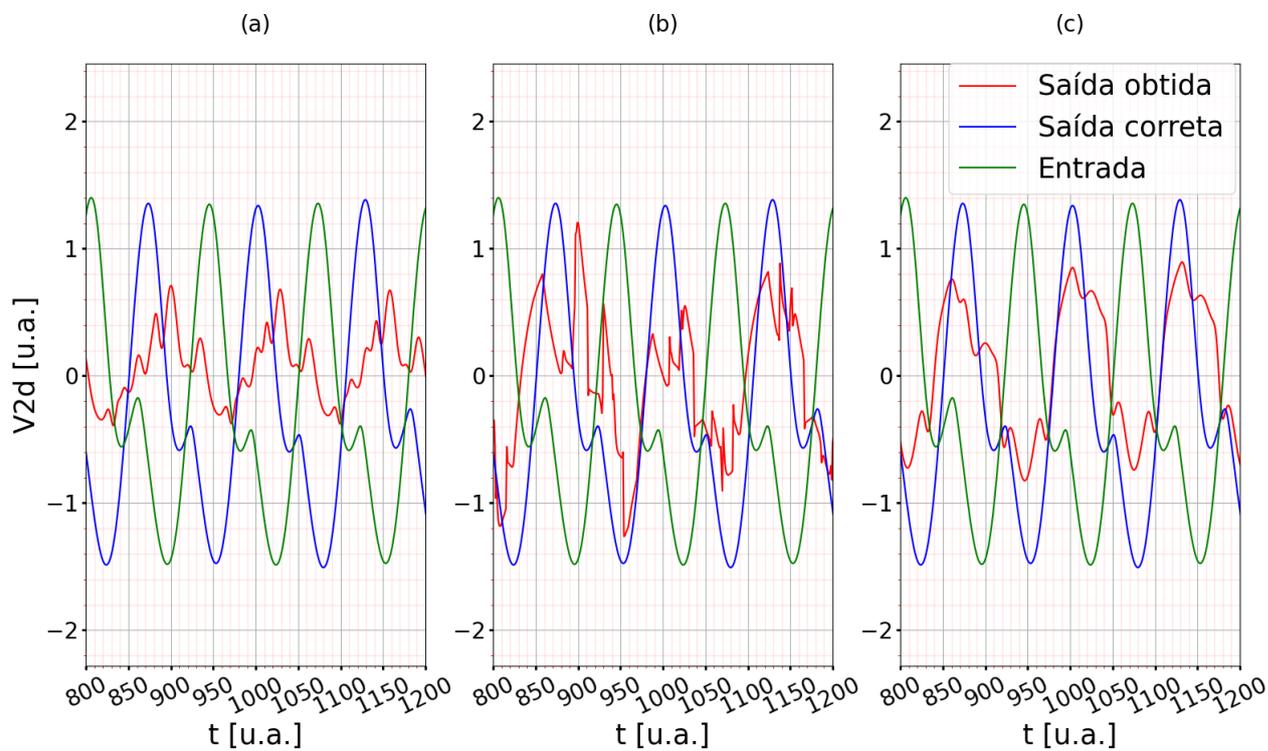
Tabela 8 – Comparação entre HP01, HP02 e HP03 obtidos na otimização onde a entrada da RCR é a variável  $V_{2d}$  e a saída é treinada para antecipar a variável  $V_{2d}$  em um IP de 20TL

Parâmetro	HP01	HP02	HP03
N	30	79	56
Lr	0,5000	0,0469	0,0552
Sr	0.9000	0.4088	0.4235
lss	1,000	1,875	1,829
Rc_con	0,1000	0,0088	0,0248
Seed	10000	86	86
Activ	tanh	tanh	tanh
Noise_rc	0	0	0
Ridge	1,000e-7	1,288e-07	7.625e-08
$R^2$	5%	23%	18%
NRMSE	20%	79%	82%

Fonte: Autor, 2023.

Um aspecto importante da aprendizagem supervisionada é a proporção entre dados de treino e teste, pois uma escolha incorreta nesta proporção pode influenciar diretamente na capacidade da RCR de aprender o fenômeno, seja por não apresentar pontos suficientes para que a RCR capture a dinâmica do fenômeno, seja por uma escolha de conjunto de testes muito pequena, não suficiente para avaliar o fenômeno ou a aprendizagem da RCR. Nos testes realizados com  $V_{2d}$  foi utilizada a proporção 60/40 em um conjunto de 10.000 pontos, onde 60% da amostra foi para o treino da RCR e 40% foi para o teste de previsão. Além disso, também é importante avaliar o

Figura 24 – Amostras das previsões de  $V_{2d}$  no IP de 20TL utilizando (a) RCR-HP01, com  $R^2 = 5\%$  e previsão apresentando pouca relação com a saída correta, (b) RCR-HP02, com  $R^2 = 23\%$  e um  $NRMSE = 79\%$  e (c) RCR-HP03, com  $R^2 = 18\%$  e um  $NRMSE = 82\%$ . Tanto com HP02 como com HP03 a RCR ainda não produziu a amplitude e frequência corretas. A hipótese de que um aumento no processamento, com mais neurônios e no número de testes de otimização devem permitir a convergência da RCR.

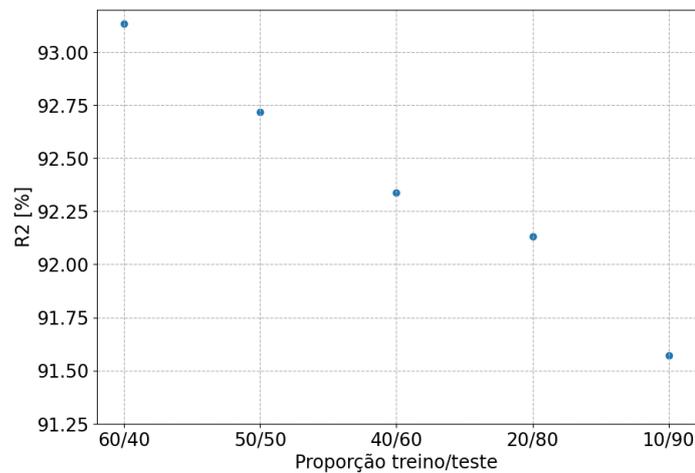


Fonte: Autor, 2024

volume de dados utilizados, no caso da proporção 60/40, o conjunto utilizado foi de 10.000 pontos. Assim, visando testar a proporção e o volume dos dados para treino e testes com variáveis caóticas, a RCR-HP03 otimizada no IP de 3TL, que produziu o melhor resultado de previsão com  $R^2 = 95\%$ , foi testada em um conjunto de 100.000 pontos, onde foram avaliadas também as proporções 10/90, 20/80, 40/60 e 50/50. Em cada proporção foram aferidos  $R^2$  e  $RMSE$ . Os resultados indicam que ao reduzir a quantidade de dados para treino na proporção, ocorre um pequeno aumento do erro  $R^2$ , que permanece acima de 91% e do erro  $RMSE$ , que permanece abaixo de 0,290, olhar ponto 10/90 nas Figuras 25 e 26. Isto significa que 10.000 pontos para treino são suficientes para a RCR-HP03 otimizada em 3TL aprender a dinâmica da variável  $V_{2d}$  e realizar a previsão de 90.000 pontos de teste. Este resultado também confirma a premissa de que resultados obtidos em um pequeno conjunto de dados se reproduzem em um conjunto maior ao utilizar RCR (Lukoševičius, 2012).

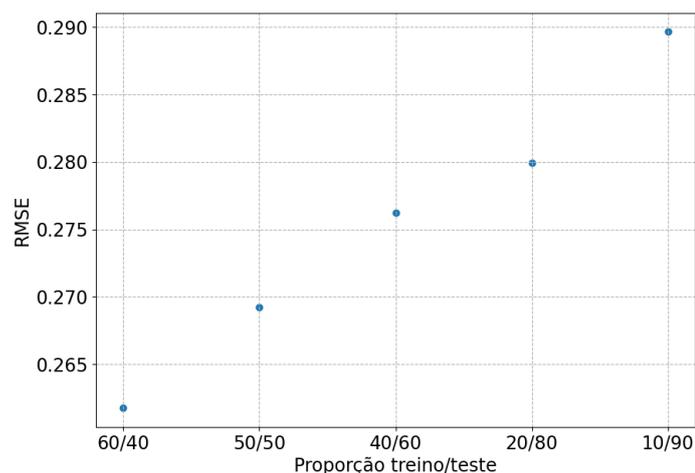
Na Figura 27 está uma comparação da flutuação do erro  $R^2$  na previsão de  $V_{2d}$ , obtido com a RCR-HP01, traço na cor vermelha, com aquele obtido na RCR-HP03 otimizada em 3TL, traço na cor azul, e aquele obtido na RCR-HP03 otimizada em 20TL, traço na cor verde. Para esta comparação, cada RCR foi testada em todos os IP. Observe que a RCR-HP03 de 3TL, traço azul, apresentou resultados melhores

Figura 25 – Variação de  $R^2$  em função da proporção utilizada para treino e teste no conjunto de  $10^5$  pontos da variável  $V_{2d}$ . Observe que a redução no conjunto de treino implica em uma pequena piora no  $R^2$ , que na menor proporção, 10/90, permanece acima de 91%.



Fonte: Autor, 2023

Figura 26 – Variação de  $RMSE$  em função da proporção utilizada para treino e teste em  $10^5$  pontos da variável  $V_{2d}$ . Note que ocorre um pequeno aumento do erro, porém permanece abaixo de 0,3 na menor proporção, 10/90.



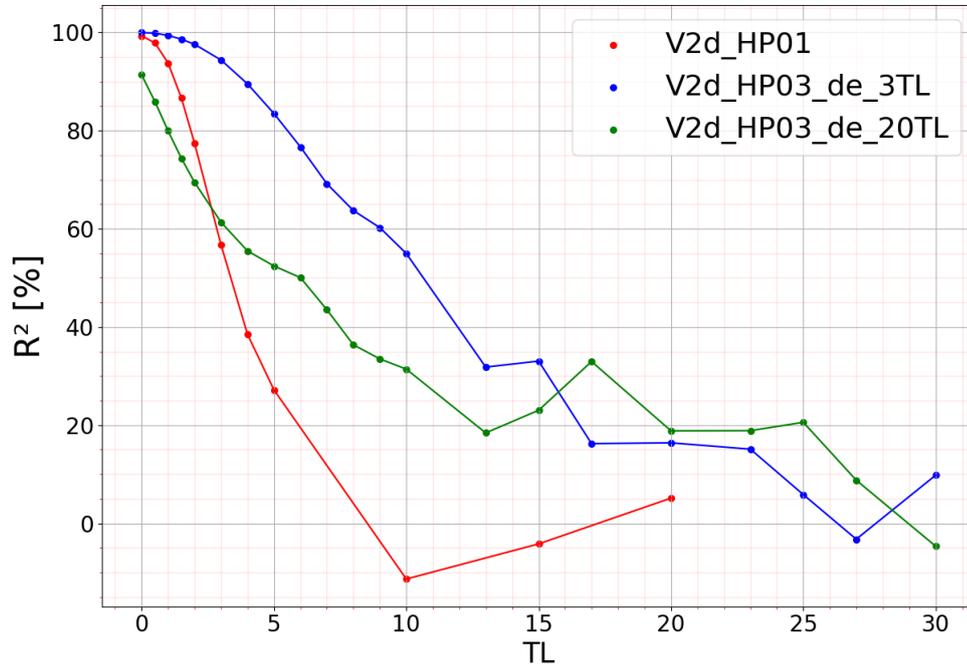
Fonte: Autor, 2023

para qualquer  $IP < 15TL$ , indicando que a melhoria em um IP específico, também afeta outros IPs. Este é um resultado esperado, pois com  $V_{1d}$  também foi observado tal comportamento, olhar Figura 22.

## 5.2 PREVISÕES DA VARIÁVEL $\|\mathbf{X}_{\perp}\|$ DO SISTEMA DR QUANDO ACOPLADO EM $V_2$

Nos resultados desta seção, a RCR foi testada quanto a habilidade de antecipar o comportamento da variável  $\|\mathbf{X}_{\perp}\|$  no maior número de TL. Aqui também foram aplicados os passos do fluxograma 15 localizado na seção 4.2.1. Na próxima seção serão apresentados resultados específicos voltados para os eventos extremos do sistema DR.

Figura 27 – Comparação da flutuação do erro  $R^2$  na previsão de  $V_{2d}$ , obtido com (a) RCR-HP01, traço na cor vermelha, (b) RCR-HP03 otimizada em 3TL, traço na cor azul, e (c) RCR-HP03 otimizada em 20TL, traço na cor verde. Observe que para  $IP < 15TL$  a RCR-HP03 de 3TL apresenta os melhores resultados.



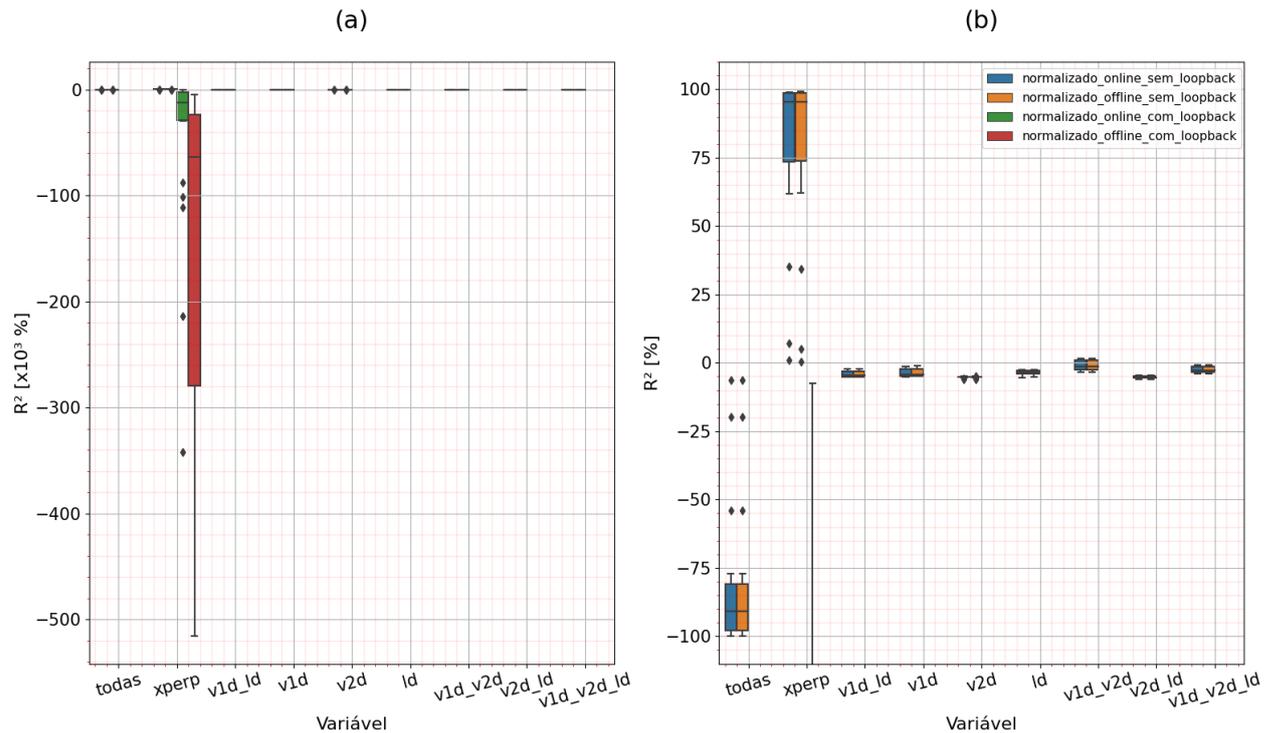
Fonte: Autor, 2023

Na Figura 28, estão os resultados das previsões da variável  $\|\mathbf{X}_\perp\|$  expressos em termos de flutuação do  $R^2$  e obtidos com a RCR gerada a partir dos hiperparâmetros HP01. Em relação aos testes com *loopback*, diagramas de caixa na cor verde, ON\_CL e vermelho, OF\_CL na Figura 28(a), a variável  $\|\mathbf{X}_\perp\|$  apresentou o mesmo comportamento das variáveis  $V_{1d}$  e  $V_{2d}$ , permaneceu sempre com valores de  $R^2$  negativos, indicando que a RCR não conseguiu aprender a dinâmica desta variável. A hipótese para explicar este comportamento é a natureza caótica da variável, que na condição com *loopback* diverge rapidamente, impedindo a RCR de produzir um resultado consistente. Cabe destacar também que ao utilizar as seis variáveis do sistema DR como entradas simultâneas, aqui designadas com o nome 'todas' (primeira coluna da esquerda para a direita na Figura 28(b)), as previsões obtidas também permaneceram com  $R^2$  negativo. Resultado que difere das previsões obtidas com 'todas' nas variáveis caóticas, olhar Figura 16 na seção 5.1. A explicação para esta diferença está relacionada com a variável a ser prevista pela RCR e com a adequação do HP01 a esta tarefa. No cenário da previsão 'todas' das variáveis caóticas, a RCR recebe na entrada as variáveis 'todas' e aprende a antecipar na saída as mesmas variáveis 'todas'. Já no cenário da previsão 'todas' das dessincronizações, a RCR recebe na entrada as variáveis 'todas', mas deve aprender a antecipar na saída a variável  $\|\mathbf{X}_\perp\|$ , que possui dinâmica distinta das variáveis caóticas exigindo mudanças nos hiperparâmetros.

Em relação às condições sem *loopback*, em azul ON\_SL e laranja OF\_SL na

Figura 28(b), os melhores resultados são obtidos quando a entrada é o próprio  $\|\mathbf{X} \perp\|$ , segunda coluna da esquerda para a direita na Figura 28(b). Isto acontece porque a RCR construída com HP01 e treinada com o próprio  $\|\mathbf{X} \perp\|$  adquire toda informação necessária para capturar a dinâmica desta variável até o IP de 6TL, olhar Figura 30.

Figura 28 – Diagramas de caixa comparando a flutuação de  $R^2$  na previsão das dessincronizações, que ocorrem quando  $\|\mathbf{X} \perp\| > 0$  no sistema DR acoplado em  $V_2$ . Como entrada da RCR foram utilizadas as variáveis, ou combinações de variáveis do sistema DR. Foram avaliadas as condições *online* com *loopback* - ON\_CL na cor verde, *online* sem *loopback* - ON\_SL na cor azul, *offline* com *loopback* - OF\_CL na cor vermelha e *offline* sem *loopback* - OF\_SL na cor laranja.



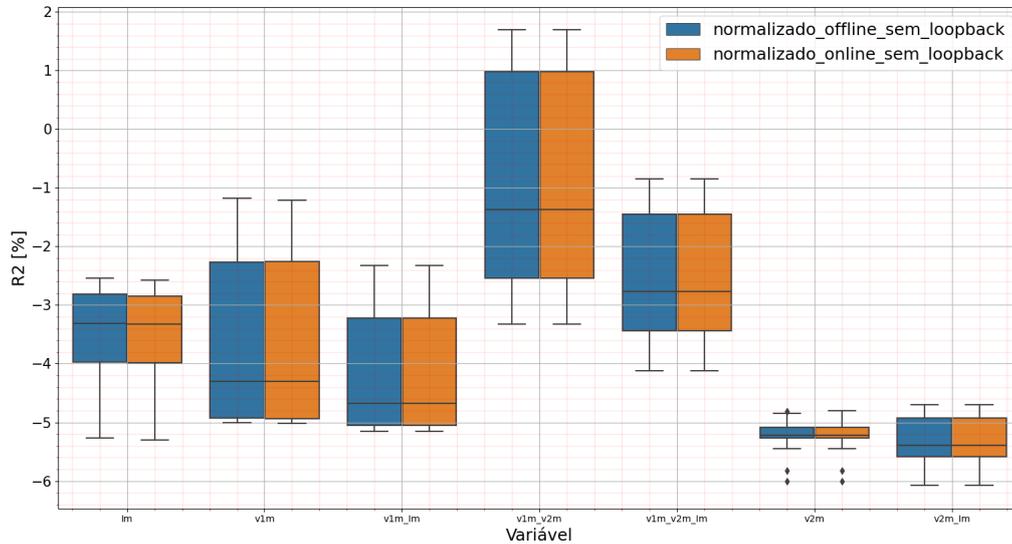
Fonte: Autor, 2023

Olhando com mais detalhes para os resultados onde a RCR é treinada com somente as variáveis do sistema DR, na Figura 29 destaca-se a previsão obtida com a RCR treinada a partir da combinação das variáveis  $V_{1d}V_{2d}$ . Na Figura 31 é possível observar que com esta combinação, o maior  $R^2$  fica abaixo de 2%, traço na cor vermelha. Neste caso, a RCR-HP01 produz um sinal pouco semelhante com o sinal correto devido à dinâmica de  $\|\mathbf{X} \perp\|$  ser diferente daquela das variáveis caóticas ou combinações de variáveis caóticas do sistema DR.

A partir dos resultados obtidos com a RCR-HP01 para a previsão de  $\|\mathbf{X} \perp\|$ , foram selecionadas as variáveis  $\|\mathbf{X} \perp\|$  e  $V_{1d}V_{2d}$  para realizar otimizações de parâmetros da RCR nos IPs de 6TL e 5TL, respectivamente.

Se na entrada da RCR for utilizada a combinação  $V_{1d}V_{2d}$ , otimizando os HP no IP de 5TL, o  $R^2$  obtido na previsão de  $\|\mathbf{X} \perp\|$  salta de 1,7% para 21,4% com HP02 e alcança o máximo de 22,4% com HP03, olhar tabela 9. Sobre HP03, quarta coluna da tabela 9, destaca-se o número de neurônios, 462, indicando uma alta demanda

Figura 29 – Resultados das previsões de  $\|\mathbf{X}_\perp\|$  em termos de variação de  $R^2$ , tendo na entrada da RCR-HP01 as variáveis do sistema DR, excluindo  $\|\mathbf{X}_\perp\|$  e "todas". Na cor azul estão os resultados dos testes *online* sem *loopback* - ON\_SL e na cor laranja estão os resultados dos testes *offline* sem *loopback* - OF\_SL. Note que o melhor resultado é observado com a combinação  $V_{1d}V_{2d}$  no centro da figura.



Fonte: Autor, 2023

de processamento para a RCR aprender o sinal de  $\|\mathbf{X}_\perp\|$  tendo como entrada o sinal  $V_{1d}V_{2d}$ . Além disto, o baixíssimo valor de  $Lr$  sugere que para a RCR conseguir aprender a dinâmica de  $\|\mathbf{X}_\perp\|$ , 99% do estado atual do reservatório é influência de seu passado, olhar termo multiplicado por  $1 - Lr$  na equação 3.3 da seção 3.4.

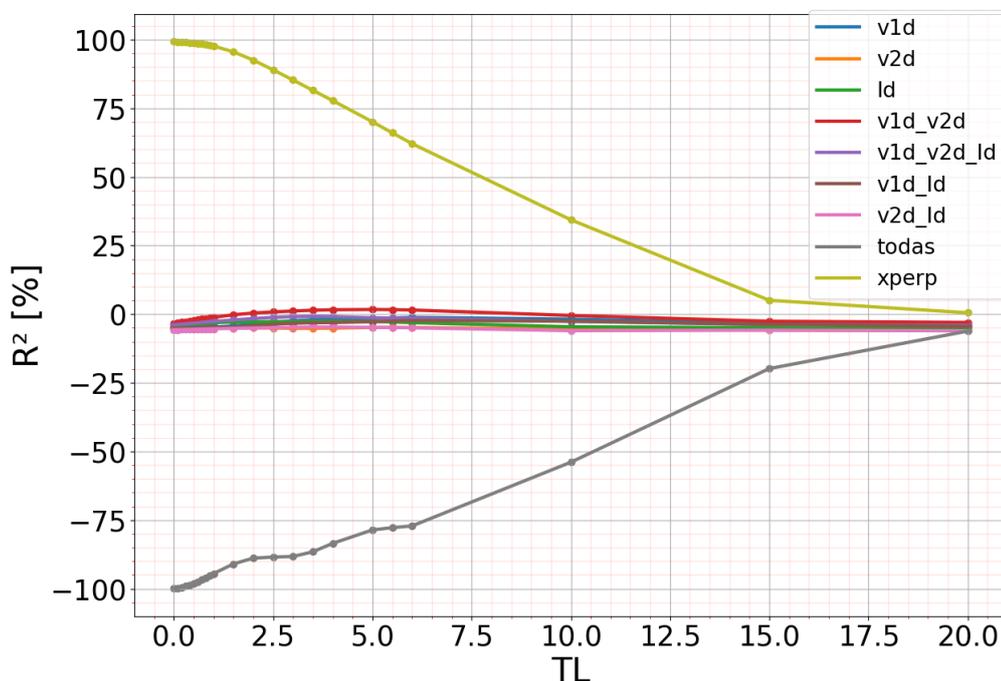
Tabela 9 – Comparação entre HP01, HP02 e HP03 obtidos na otimização onde a entrada da RCR é a combinação  $V_{1d}V_{2d}$  e a saída é treinada para antecipar a variável  $\|\mathbf{X}_\perp\|$  em um IP de 5TL

Parâmetro	HP01	HP02	HP03
N	30	342	462
Lr	0,5000	0,0143	0,0139
Sr	0,9000	0,5485	0,5431
lss	1,000	0,0567	0,0535
Rc_con	0,1000	0,1365	0,0277
Seed	10000	7	141
Activ	tanh	tanh	tanh
Noise_rc	0	0	0
Ridge	1,000e-7	2.460e-06	2.208e-06
$R^2$	1,7%	21,4%	22,4%
$NRMSE$	20,4%	18,2%	18,1%

Fonte: Autor, 2023.

Se na entrada da RCR for utilizada a variável  $\|\mathbf{X}_\perp\|$ , otimizando os HP no IP de 6TL, o  $R^2$  obtido na previsão de  $\|\mathbf{X}_\perp\|$  alcança 62,9% com HP03, olhar tabela 10. Os destaques deste HP03 obtido são  $Lr = 0,9981$ , demonstrando que  $\|\mathbf{X}_\perp\|$  tem uma dinâmica completamente diferente das variáveis caóticas,  $rc\_con$  de 43,58%, exigindo

Figura 30 – Resultado das previsões de  $\|X_{\perp}\|$  obtidas com a RCR-HP01 em função do intervalo de previsão em TL. Na entrada da RCR foram aplicadas as variáveis do sistema DR e  $\|X_{\perp}\|$ . Observe que  $\|X_{\perp}\|$ , traço na cor amarela, na entrada da RCR produz o melhor resultado, enquanto a combinação "todas", traço na cor cinza, não alcança valores mínimos de convergência.



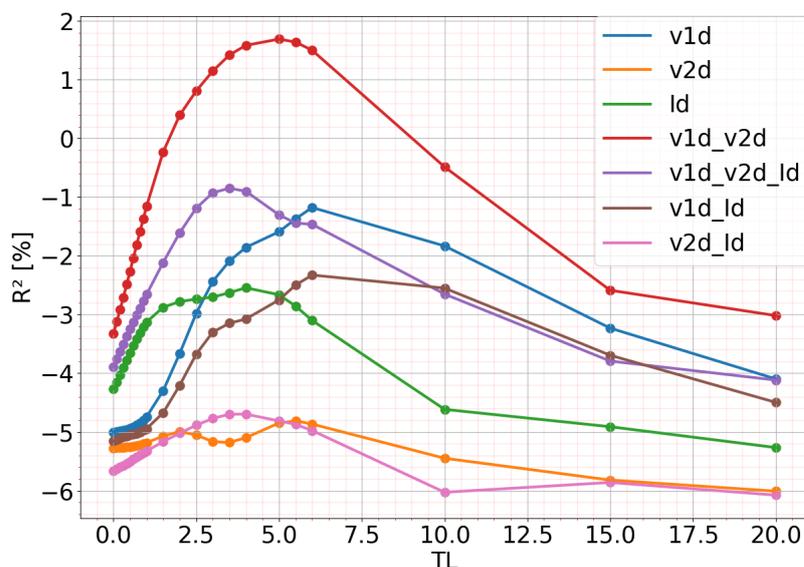
Fonte: Autor, 2023

um reservatório altamente conectado que combinado com o alto número de neurônios, 229, sugerem alta demanda de processamento para a RCR resolver a dinâmica da variável  $\|X_{\perp}\|$ .

Considerando que utilizar os parâmetros HP01 na RCR e o  $\|X_{\perp}\|$  em sua entrada, traço na cor amarela da Figura 30, obtém-se os melhores resultados para as previsões das dessincronizações num intervalo de previsão de 6TL. Optou-se por realizar uma otimização de hiperparâmetros, utilizando o método de HP03, porém tendo como parâmetro inicial o HP01 ao invés do HP02 proposto no fluxograma da Figura 15 da seção 4.2.1. Este novo conjunto foi identificado como HP04 na tabela 10 e foi obtido no intervalo de previsão de 6TL. Com a RCR-HP04 o resultado das previsões teve uma pequena melhora, o  $R^2$  foi para 62,2%. O ponto peculiar do resultado com a RCR-HP04 é o fato de ter produzido uma melhora de 0,2% com uma RCR de apenas 30 neurônios. Uma hipótese para explicar esta melhora são os ajustes de  $L_r$  e  $rc_{con}$ , pois ambos afetam a dinâmica das informações no reservatório. O incremento de  $L_r$  permitiu a RCR aproximar-se da dinâmica de  $\|X_{\perp}\|$  e o aumento do  $rc_{con}$  para 43,78% produziu uma rede de neurônios do reservatório densamente conectada.

Na Figura 32 está uma comparação entre as previsões de  $\|X_{\perp}\|$  obtidas com a RCR-HP01, RCR-HP03 e RCR-HP04 em vários IP. Sobre estes resultados deve-se notar que a RCR-HP04 é a opção mais eficiente, pois alcançou resultados similares àqueles obtidos com a RCR-HP01 e RCR-HP03, porém utilizando somente

Figura 31 – Resultados das previsões de  $\|\mathbf{X} \perp\|$  em termos de variação de  $R^2$ , tendo na entrada da RCR-HP01 as variáveis do sistema DR, excluindo  $\|\mathbf{X} \perp\|$  e "todas". Esta figura detalha os dados contidos nos diagramas de caixa da Figura 29.



Fonte: Autor, 2023

Tabela 10 – Comparação entre HP01, HP02, HP03 e HP04 obtidos na otimização onde a entrada da RCR é a variável  $\|\mathbf{X} \perp\|$  e a saída é treinada para antecipar a variável  $\|\mathbf{X} \perp\|$  em um IP de 6TL

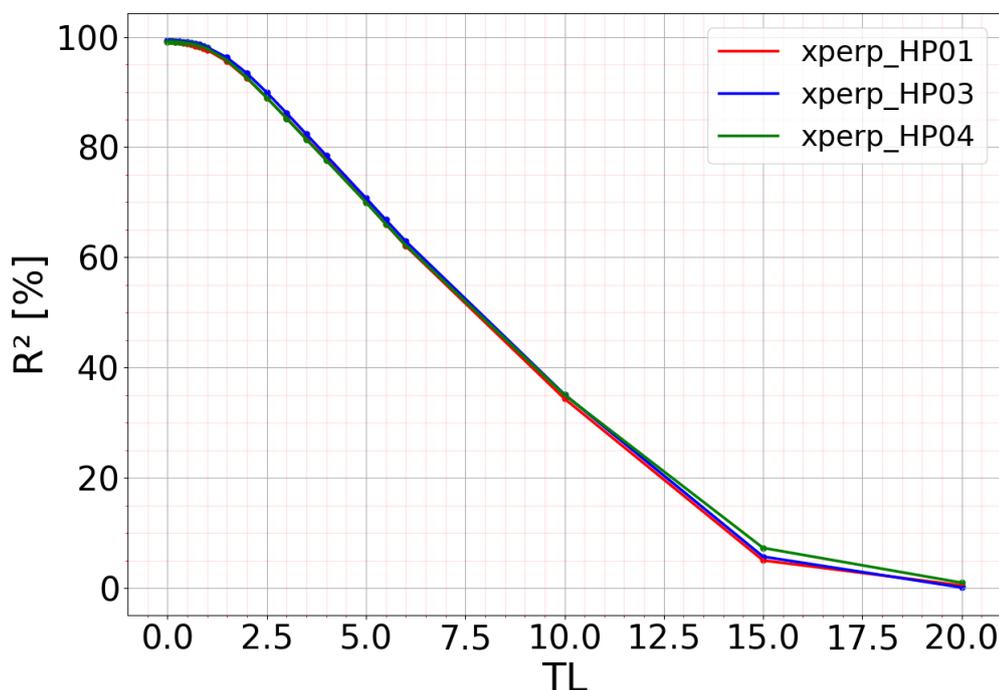
Parâmetro	HP01	HP02	HP03	HP04
N	30	249	229	30
Lr	0,5000	0,9679	0,9981	0,7216
Sr	0,9000	0,3567	0,3548	0,8761
lss	1000	0,9152	0,9136	0,8914
Rc_con	0,1000	0,0522	0,4358	0,4378
Seed	10000	66	67	277
Activ	tanh	tanh	tanh	tanh
Noise_rc	0	0	0	0
Ridge	1,000e-7	2,217e-08	2,078e-08	1,518e-07
$R^2$	62%	62,5%	62,9%	62,2%
$NRMSE$	12,7%	12,6%	12,5%	12,6%

Fonte: Autor, 2023.

30 neurônios. Segundo ponto é a necessidade de corrigir algum aspecto da técnica utilizada, possivelmente aumentando a quantidade de testes na otimização, pois os conjuntos de testes de HP02 e HP03 não foram suficientes para obter uma melhoria maior que 1%. Embora a otimização de parâmetros tenha melhorado a previsão de  $\|\mathbf{X} \perp\|$  tendo a variável  $\mathbf{V}_{1d\_V_{2d}}$  na entrada da RCR, o  $R^2$  abaixo de 50% ainda é um resultado com baixa confiabilidade para as previsões de  $\|\mathbf{X} \perp\|$ .

Quando  $\mathbf{V}_{1d\_V_{2d}}$  é utilizado na entrada da RCR, ver Figura 33, os testes de otimização com HP03 melhoraram o  $R^2$  em todos os IP, alcançando o valores próximos de 26% para os IP menores que 2,5TL.

Figura 32 – Comparação entre as previsões de  $\|\mathbf{X} \perp\|$  obtidas com a RCR-HP01, RCR-HP03 e RCR-HP04 em vários IP. Observe que os três HP foram otimizados no IP de 6TL.



Fonte: Autor, 2023

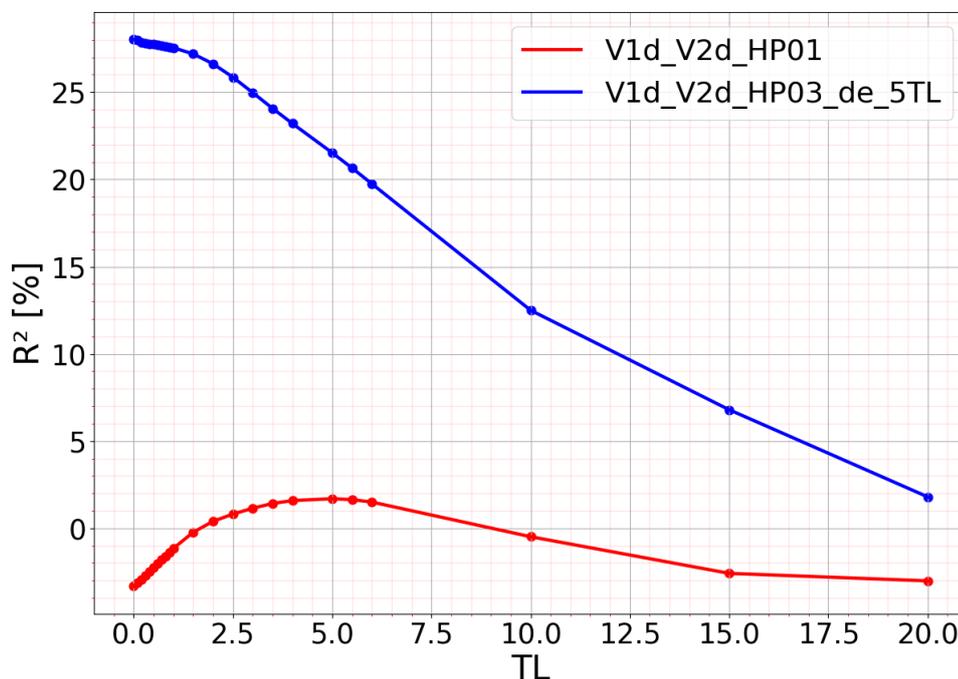
Os resultados desta seção apontam que os conjuntos HP03 e HP04 obtidos tendo na entrada da RCR o  $\|\mathbf{X} \perp\|$  e otimizando seus HP no IP de 6TL apresentam os melhores resultados. Portanto, as RCR-HP03 e RCR-HP04 foram selecionadas para testar as previsões de eventos extremos do sistema DR na próxima seção.

### 5.3 PREVISÕES DOS EVENTOS EXTREMOS DO SISTEMA DR

Quando a variável  $\|\mathbf{X} \perp\| > 0$ , são observadas as dessincronizações do sistema DR. Aquelas que alcançam as maiores amplitudes são os eventos extremos. Em termos práticos, a escolha do nível de evento extremo foi estabelecido a partir da observação do comportamento da função de densidade de probabilidade, ver Figura 34, gerada com os dados da simulação de  $10^6$  pontos do sistema DR, obtidos na seção 4.1 do capítulo 4. Observe que a região mais à direita no gráfico da função de densidade de probabilidade da Figura 34 apresenta a assinatura dos eventos extremos do tipo *Dragon-King* (Sornette, 2009; Cavalcante *et al.*, 2013), compare a Figura 34 com a Figura 1, da seção 2.1.

A partir do gráfico da Figura 34, foi estabelecida a amplitude igual a 1,5 unidades arbitrárias (u.a.) como referência para o evento extremo, identificado com um tracejado vertical na cor vermelha. Este ponto foi escolhido por estar próximo do início da cauda gorda, do inglês *fat tail*, da distribuição, onde se espera observar os eventos extremos. Assim, toda vez que o  $\|\mathbf{X} \perp\| > 1,5u.a.$ , a dessincronização será considerada um

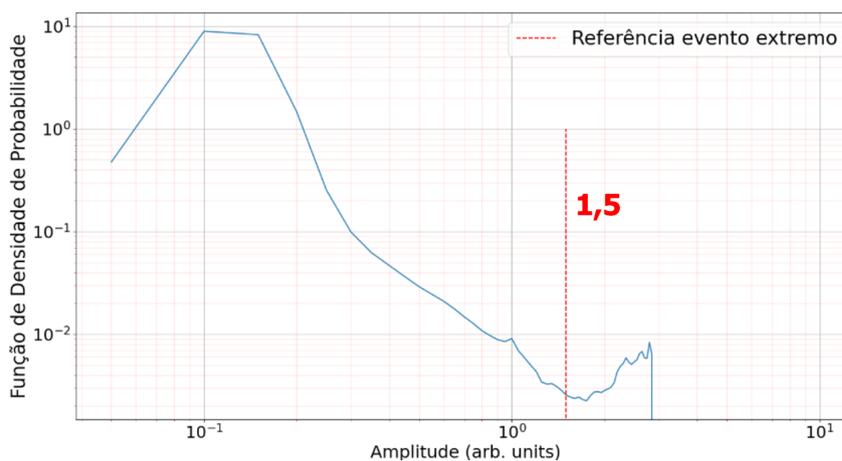
Figura 33 – Comparação entre as previsões de  $\|X_{\perp}\|$  obtidas com a RCR-HP01 e RCR-HP03, tendo como variável na entrada da RCR a combinação de variáveis  $V_{1d}, V_{2d}$ . Observe que as otimizações de HP01 e HP03 foram realizadas no IP 5TL. Note também a melhoria significativa obtida com a RCR-HP03, traço na cor azul.



Fonte: Autor, 2023

evento extremo do sistema DR.

Figura 34 – Gráfico da função de densidade de probabilidade produzida a partir dos dados simulados do sistema DR acoplado em  $V_2$ . O traço vertical na cor vermelha indica a amplitude de 1,5 unidades arbitrárias, a partir da qual o evento é considerado extremo. Observe que após o tracejado vertical, a função apresenta o comportamento característico dos *Dragon-King*, uma cauda "gorda" ou *fat tail*.

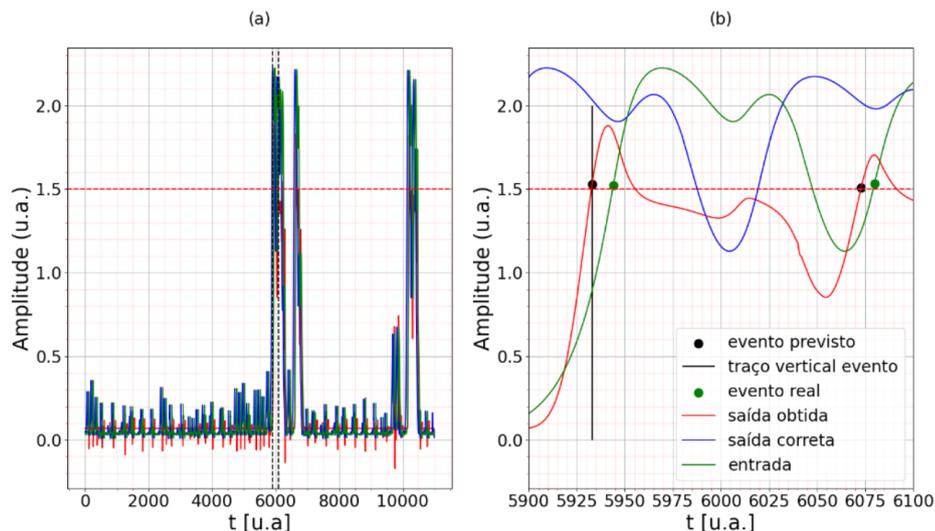


Fonte: Autor, 2023

Foram utilizados os hiperparâmetros HP03 e HP04, obtidos na seção 5.2, para construir as RCR que anteciparam os eventos extremos. As RCR realizaram previsões para os IP de 1TL, 3TL e 6TL. As medidas ficaram restritas a estes IP devido ao valor onde a RCR foi otimizada, que foi 6TL. No sinal gerado pela RCR, foram realizadas marcações e contagens dos valores que superaram a referência de 1,5u.a.

Na Figura 35(a) estão os 17.000 pontos utilizados para testar as previsões. Note que somente os eventos que superaram a marcação de 1,5 u.a. são tratados como extremos. No recorte da Figura 35(b) é apresentado um evento extremo antecipado pela RCR-HP03. A previsão obtida, ponto na cor preta marcado sobre o traço vermelho, não acertou amplitude, nem a duração do evento, porém acertou a ocorrência de um evento extremo 1,1TL antes da ocorrência real, ponto na cor laranja marcado sobre o traço na cor verde. A primeira consideração sobre este resultado é que a RCR-HP03 antecipou corretamente a ocorrência de um evento extremo 1,1TL à frente. Isto é um ganho em relação às técnicas estatísticas, pois tais métodos indicam as chances de ocorrência sem apontar quando acontecerá. A segunda consideração é que se trata da previsão de um evento extremo, cuja natureza implica que o evento real deve ser tratado como um evento na maior magnitude histórica do fenômeno em observação. Outra consideração relevante é que a RCR com  $R^2$  igual a 62,9% não foi suficiente para prever amplitude nem a duração do evento, portanto deve-se buscar uma nova otimização dos hiperparâmetros modificando algum aspecto da técnica para aumentar o  $R^2$  e, por consequência, melhorar as previsões.

Figura 35 – Previsão do  $\|\mathbf{X} \perp\|$  (na cor vermelha) do sistema DR acoplado em  $V_2$ , obtido pela RCR construída com o HP03 otimizado no IP de 6TL, treinada com 17000 pontos e testada em 11000 pontos. O traço horizontal em (a) e (b) na cor vermelha tracejado, representa o limite de 1,5 u.a. onde uma dessincronização será considerada evento extremo. O evento detectado pela RCR é o ponto na cor preta sobre o traço vermelho em (b) e o evento real é o ponto na cor verde sobre o traço verde em (b), a linha azul representa a saída correta esperada na previsão, neste caso, 6TL antes de ocorrer o evento real. Na primeira detecção, evento com um traço vertical na cor preta, a RCR antecipou o evento real em 1,1TL.



Fonte: Autor, 2024

A equação 5.1 (Racca e Magri, 2022), permite estimar a precisão ( $p$ ) da RCR considerando a contagem de eventos verdadeiros positivos (VP), isto é, são aqueles que a RCR indica e que de fato ocorrem e falso positivos (FP), que são aqueles que a RCR indica, mas não ocorrem efetivamente. Quando o sinal gerado pela RCR indicou um evento extremo, portanto acima de 1,5 u.a. e na região do eixo temporal entre o

evento real e o esperado, o evento extremo foi considerado VP, caso contrário, FP.

$$p = \frac{VP}{VP + FP}. \quad (5.1)$$

Na tabela 11 estão os resultados das contagens de eventos extremos obtidas com as RCR e os cálculos de precisão realizados com a equação 5.1. Para o IP de 1TL, as RCR-HP03 e RCR-HP04 apresentaram precisão similar. Neste caso, a RCR-HP04 foi mais eficiente, pois utilizou somente 30 neurônios em sua rede. No IP de 3TL, nas previsões realizadas no conjunto de  $30 \times 10^3$  pontos, as RCR-HP03 e RCR-HP04 apresentaram precisão igual a 85%, mas no conjunto de  $100 \times 10^3$  pontos, a RCR-HP04 apresentou precisão de 72,7%. No intervalo de 6TL, a RCR-HP03 teve precisão melhor, igual a 71% para o conjunto de  $30 \times 10^3$  pontos. No conjunto de  $100 \times 10^3$  pontos e IP de 6TL, as duas RCR apresentaram um resultado menor ou igual a 50% de precisão. De maneira geral,  $p$  cai conforme aumenta o IP, ou em função do aumento do conjunto de dados testados. É importante observar que os valores considerados VP são aqueles pontos da previsão que estão na região do eixo  $x$  entre o valor real e o esperado para o evento extremo.

Tabela 11 – Comparação entre as previsões de eventos extremos obtidas nos IP 1, 3, e 6TL com as RCR-HP03 e RCR-HP04

HP	IP [TL]	Pontos [ $\times 10^3$ ]	Treino/ Teste	Reais	VP	FP	$p$	$\bar{t}$ [TL]
HP03	1	30	17/ 13	6	6	0	100%	$0,70 \pm 0,28$
HP04	1	30	17/ 13	6	6	0	100%	$0,80 \pm 0,33$
HP03	1	100	40/ 60	16	16	3	84%	$1,06 \pm 0,32$
HP04	1	100	40/ 60	16	16	4	80%	$0,99 \pm 0,34$
HP03	3	30	17/ 13	6	6	1	85%	$0,95 \pm 0,43$
HP04	3	30	17/ 13	6	6	1	85%	$0,43 \pm 0,20$
HP03	6	30	17/ 13	6	5	2	71%	$0,88 \pm 0,43$
HP04	6	30	17/ 13	6	4	2	66%	$0,40 \pm 0,08$
HP03	3	100	40/ 60	16	15	9	62,5%	$1,48 \pm 0,62$
HP04	3	100	40/ 60	16	16	6	72,7%	$1,26 \pm 0,46$
HP03	6	100	40/ 60	16	16	16	50%	$1,10 \pm 0,52$
HP04	6	100	40/ 60	16	13	17	43%	$0,81 \pm 0,33$

Fonte: Autor, 2023.

Outro aspecto importante apresentado na tabela 11 são os tempos médios (última coluna) das previsões. A primeira consideração em relação a esses tempos é que se busca uma combinação entre precisão  $p$  alta (penúltima coluna) e um tempo médio  $\bar{t}$  que se aproxima ao máximo do valor do IP esperado (segunda coluna). Portanto, as previsões que apresentaram a melhor relação foram aquelas obtidas no IP de 1TL aplicadas ao conjunto de  $100 \times 10^3$  pontos, utilizando o esquema de treino/teste 40/60, olhar linhas 3 e 4 da tabela. Uma hipótese para explicar estes resultados está relacionada com a qualidade das previsões das RCR-HP03 e RCR-HP04, que tiveram

um  $R^2$  de 62,9% e 62,2% na fase de otimização, combinado com o IP de 1TL onde as RCR foram testadas, pois estes valores baixos de  $R^2$  não permitiram que a RCR alcançasse tempos melhores em IPs maiores. Outra hipótese está relacionada com o esquema de treino/teste, é possível que uma proporção de treino/teste maior, por exemplo, 50/50 ou 60/40 sejam necessários para permitir a RCR uma melhoria nos tempos médios em IPs maiores.

## 6 CONCLUSÕES E PERSPECTIVAS

Os objetivos desta dissertação foram prever as séries caóticas, as dessincronizações e os eventos extremos observados no sistema DR utilizando redes de computação de reservatório (RCR). Para alcançar estes objetivos, foram realizadas otimizações dos hiperparâmetros da RCR, os quais foram ajustados para prever as variáveis caóticas  $V_{1d}$ ,  $V_{2d}$ , as dessincronizações e os eventos extremos onde  $\|\mathbf{X} \perp\| > 0$ .

As RCR utilizadas neste trabalho somente convergiram quando foram treinadas e testadas no modo sem *loopback*. No modo com *loopback* todas as tentativas produziram um  $R^2$  negativo, que significa uma previsão da RCR completamente diferente do valor esperado. A principal hipótese para explicar este resultado é a natureza caótica do sinal utilizado, pois quando a RCR é colocada no modo com *loopback*, seu sinal de saída diverge rapidamente. Assim, utilizando o método deste trabalho, as melhores previsões são produzidas no modo sem *loopback*.

Sobre as variáveis caóticas, a RCR otimizada produziu previsões com  $R^2$  de 86% em 10 TL na variável  $V_{1d}$  e 95% em 3TL na variável  $V_{2d}$ , portanto, o método utilizado se mostrou eficiente para prever as variáveis caóticas do sistema DR.

Em relação às previsões de dessincronizações, na condição onde a RCR utiliza como entrada alguma variável do sistema DR, por exemplo, a combinação  $V_{1d\_V_{2d}}$ , previsões alcançaram um  $R^2$  máximo de 26,2%, logo, prever  $\|\mathbf{X} \perp\|$  somente com variáveis caóticas não é viável com o método proposto. É possível que incrementar o número de testes na otimização permita a RCR convergir para um resultado melhor ou pode ser que a RCR precise de mais informações além das variáveis caóticas para prever o  $\|\mathbf{X} \perp\|$ . Por outro lado, se a RCR for treinada com  $\|\mathbf{X} \perp\|$  na entrada para antecipar  $\|\mathbf{X} \perp\|$  na saída, se observa o melhor resultado, pois o  $R^2$  alcançou 62,9% em 6TL. A explicação para este resultado é que a RCR treinada com  $\|\mathbf{X} \perp\|$  tem mais informações sobre esta variável, o que se reflete na qualidade das previsões obtidas pela RCR.

Quanto à previsão de eventos extremos, a RCR-HP03 foi capaz de antecipar corretamente a ocorrência de eventos extremos em 1,06TL com uma precisão de 84% de acerto. Isto é um ganho em relação às técnicas estatísticas, pois neste caso o intervalo de tempo da ocorrência é corretamente identificado pela RCR, enquanto pelos métodos estatísticos o que se obtém são probabilidades de ocorrência dos eventos. Outra conclusão relevante é que a RCR com  $R^2$  igual a 62,9% não foi suficiente para prever amplitude nem a duração do evento. Supõe-se que modificar algum aspecto da técnica de otimização dos HP da RCR possa aumentar o  $R^2$  e, por consequência, melhorar os resultados. Entretanto, por se tratar da previsão de um evento extremo, em relação à amplitude, os eventos previstos pela RCR devem ser tratados como eventos da maior magnitude histórica do fenômeno em observação.

A dinâmica de  $\|X_{\perp}\|$  é completamente diferente da dinâmica das variáveis caóticas. A justificativa são os diferentes valores de  $L_r$ , pequeno nas variáveis caóticas e grande em  $\|X_{\perp}\|$ , obtidos durante a otimização dos hiperparâmetros. O  $L_r$  ajusta a RCR para a frequência do sinal que está sendo aprendido. Assim, é importante considerar este aspecto em trabalhos futuros de otimização da RCR.

Neste trabalho a influência dos ruídos, tanto no sinal de entrada quanto no reservatório não foram avaliados. Portanto, em condições ideais, a RCR prevê corretamente as variáveis caóticas, mas ainda precisa de ajustes para prever corretamente os eventos extremos. Considerando que sinais reais contém ruídos, é importante que em trabalhos futuros seja avaliada a influência dos ruídos nestas previsões.

Outro aspecto importante para aplicações práticas é o tempo de resposta para a RCR produzir uma previsão. É interessante implementar a RCR com reservatórios físicos para avaliar este tempo de resposta. A hipótese é que uma RCR baseada em hardware deve apresentar respostas mais rápidas que as RCR baseadas em software. Principalmente em aplicações que requerem respostas em tempo real. Além das previsões do sistema DR, algumas sugestões para trabalhos futuros são o uso do método proposto em testes com dados do mercado financeiro, plantas de fornecimento de energia, previsão de ataques epiléticos, previsão de chuvas torrenciais entre outros.

## REFERÊNCIAS

ALBEVERIO, Sergio; JENTSCH, Volker; KANTZ, Holger. **Extreme Events in Nature and Society**. Heidelberg: Springer Berlin, 2006. 352 p. (The Frontiers Collection). *E-book*.

ANTONIK, P.; GULINA, M.; PAUWELS, J.; MASSAR, S. Using a reservoir computer to learn chaotic attractors, with applications to chaos synchronization and cryptography. **Physical Review E**, v. 98, n. 1, p. 012215, 2018. Disponível em: <https://journals.aps.org/pre/abstract/10.1103/PhysRevE.98.012215>. Acesso em: 28 de jan. 2022.

ARCANGELIS, L. de. Are dragon-king neuronal avalanches dungeons for self-organized brain activity?. **The European Physical Journal Special Topics**, v. 205, p. 243–257, 2012. ISSN 1951-6355,1951-6401. Disponível em: <http://doi.org/10.1140/epjst%2Fe2012-01574-6>. Acesso em: 18 de ago. 2023.

BERGSTRA, J. *et al.* Making a science of model search: hyperparameter optimization in hundreds of dimensions for vision architectures. *In: PROCEEDINGS OF THE 30TH INTERNATIONAL CONFERENCE ON MACHINE LEARNING*, 30., 2013, Atlanta. **Proceedings of Machine Learning Research**. Atlanta, 2013. p. 115 – 123. Disponível em: <https://proceedings.mlr.press/v28/bergstra13.pdf>. Acesso em: 22 de jul. 2023.

BARROS NETO, Benício de; SCARMINIO, Ieda Spacino; BRUNS, Roy Edward. **Como fazer experimentos**. Campinas: Editora Unicamp, 2007.

BOCCALETTI, S.; KURTHS, J.; OSIPOV, G.; VALLADARES, D.; ZHOU, C. The synchronization of chaotic systems. **Physics Reports**, [s. l.], v. 366, n. 1, p. 1–101, 2002. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0370157302001370>. Acesso em: 13 de out. 2021.

BOCCALETTI, S.; LATORA, V.; MORENO, Y.; CHAVEZ, M.; HWANG, D.-U. Complex networks: Structure and dynamics. **Physics reports**, [s. l.], v. 424, n. 4-5, p. 175–308, 2006. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S037015730500462X>. Acesso em: 10 de set. 2023.

BOORE, D. M. The richter scale: its development and use for determining earthquake source parameters. **Tectonophysics**, [s. l.], v. 166, n. 1, p. 1–14, 1989. Disponível em: <https://www.sciencedirect.com/science/article/pii/004019518990200X>. Acesso em: 13 de out. 2021.

CAMPBELL, Murray; HOANE JR, A. Joseph; HSU, Feng-hsiung. Deep blue. **Artificial intelligence**, v. 134, n. 1-2, p. 57-83, 2002.. ISSN 0004-3702. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0004370201001291>. Acesso em: 21 de set. 2023.

CAVALCANTE, H. L. D. S.; ORIÁ, M.; SORNETTE, D.; OTT, E.; GAUTHIER, D. J. Predictability and suppression of extreme events in a chaotic system. **Physical Review Letters**, Maryland, v. 111, n. 19, p. 198701, 2013. Disponível em: <https://link.aps.org/doi/10.1103/PhysRevLett.111.198701>. Acesso em: 29 jan. 2021.

CHOWDHURY, S. N.; RAY, A.; DANA, S. K.; GHOSH, D. Extreme events in dynamical systems and random walkers: A review. **Physics Reports**, [s. l.], v. 966, p. 1–52, 2022. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0370157322001144>. Acesso em: 13 de nov. 2022.

CRISE varre meio trilhão da bolsa em 2008. **Folha de São Paulo**, São Paulo, 2 outubro 2008. Caderno Mercado. Disponível em: <https://m.folha.uol.com.br/mercado/2008/10/451291-crise-varre-meio-trilhao-da-bolsa-em-2008.shtml>. Acesso em: 08 jun. 2022.

CUCCHI, M.; ABREU, S.; CICCONE, G.; BRUNNER, D.; KLEEMANN, H. Hands-on reservoir computing: a tutorial for practical implementation. **Neuromorphic Computing and Engineering**, v. 2, n. 3, p. 032002, 2022. Disponível em: <https://iopscience.iop.org/article/10.1088/2634-4386/ac7db7/meta>. Acesso em: 24 de abr. 2023.

DABROWSKI, A. Estimation of the largest lyapunov exponent-like (llel) stability measure parameter from the perturbation vector and its derivative dot product (part 2) experiment simulation. **Nonlinear Dynamics**, v. 78, p. 1601–1608, 2014. Disponível em: <https://link.springer.com/article/10.1007/s11071-014-1542-7>. Acesso em: 27 de nov. 2023.

DOMINEY, P. F. Complex sensory-motor sequence learning based on recurrent state representation and reinforcement learning. **Biological cybernetics**, v. 73, n. 3, p. 265–274, 1995. Disponível em: <https://link.springer.com/article/10.1007/BF00201428>. Acesso em: 18 de ago. 2023

EROGLU, D.; LAMB, J. S.; PEREIRA, T. Synchronisation of chaos and its applications. **Contemporary Physics**, v. 58, n. 3, p. 207–243, 2017. Disponível em: <https://www.tandfonline.com/doi/abs/10.1080/00107514.2017.1345844>. Acesso em: 7 de nov. 2021.

FARAZMAND, M.; SAPSIS, T. P. Extreme events: Mechanisms and prediction. **Applied Mechanics Reviews**, v. 71, n. 5, 2019. Disponível em: <https://asmedigitalcollection.asme.org/appliedmechanicsreviews/article-abstract/71/5/050801/629878/Extreme-Events-Mechanisms-and-Prediction>. Acesso em: 22 de fev. 2022.

FEITOZA, Ricardson Alexandre Pereira. **Geração e distribuição de chaves criptográficas simétricas através da sincronização intermitente de circuitos eletrônicos caóticos acoplados**. Dissertação (Mestrado em Engenharia Física) – Unidade Acadêmica do Cabo de Santo Agostinho, Universidade Federal Rural de

Pernambuco, Cabo de Santo Agostinho, 2022. Disponível em: <http://tede2.ufpe.br:8080/tede/handle/tede2/8589>. Acesso em: 15 de ago. 2023.

FERREIRA, Aida Araújo. **Um método para design e treinamento de reservoír computing aplicado à previsão de séries temporais**. Tese (Doutorado em Ciência da Computação) - Universidade Federal de Pernambuco, 2011. Disponível em: <https://repositorio.ufpe.br/handle/123456789/1786>. Acesso em: 20 de ago. 2023.

FUJISAKA, H.; YAMADA, T. Stability theory of synchronized motion in coupled-oscillator systems. **Progress of Theoretical Physics**, [s. l.], v. 69, n. 1, p. 32–47, 1983. Disponível em: <https://academic.oup.com/ptp/article/69/1/32/1835380?login=false>. Acesso em: 14 out. 2021.

GAUTHIER, D. J.; BIENFANG, J. C. Intermittent loss of synchronization in coupled chaotic oscillators: Toward a new criterion for high-quality synchronization. **Physical Review Letters**, Maryland, v. 77, n. 9, p. 1751–1754, 1996. Disponível em: <https://link.aps.org/doi/10.1103/PhysRevLett.77.1751>. Acesso em: 5 de set. 2021.

GAUTHIER, D. J.; BOLLT, E.; GRIFFITH, A.; BARBOSA, W. A. Next generation reservoir computing. **Nature communications**, v. 12, n. 1, p. 5564, 2021. Disponível em: <https://www.nature.com/articles/s41467-021-25801-2>. Acesso em: 10 de mai. 2023.

HAYKIN, Simon. **Redes Neurais: princípios e prática**. 2. ed. São Paulo: Bookman, 2001. Tradução de Paulo Martins Engel. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9788577800865/pages/recent>. Acesso em: 25 ago. 2022.

IZBICKI, Rafael; SANTOS, Tiago Mendonça dos. **Aprendizado de máquina: uma abordagem estatística**. Rafael Izbicki, 2020. Disponível em: <https://www.rizbicki.ufscar.br/AME.pdf>. Acesso em: 12 de jan. 2023.

JAEGER, H. The “echo state” approach to analysing and training recurrent neural networks-with an erratum note. **German National Research Center for Information Technology GMD Technical Report**, Bonn, v. 148, n. 34, p. 13, 2001. Disponível em: <https://www.ai.rug.nl/minds/uploads/EchoStatesTechRep.pdf>. Acesso em: 26 de jul. 2023.

KIRBY, K. G.; DAY, N. The neurodynamics of context reverberation learning. *In*: PROCEEDINGS OF THE TWELFTH ANNUAL INTERNATIONAL CONFERENCE OF THE IEEE ENGINEERING IN MEDICINE AND BIOLOGY SOCIETY, 12., 1990, Philadelphia. **Proceedings** [...]. Philadelphia, 1990. p. 1781–1782. Disponível em: <https://ieeexplore.ieee.org/abstract/document/692015>. Acesso em: 27 de jul. 2023.

KOVALSKY MARCELO G.; HNILO, A. A. T. J. R. Extreme events in the ti:sapphire laser. **Optics Letters**, v. 36, n. 22, p. 4449-4451, 2011. Disponível em: <https://opg.optica.org/ol/abstract.cfm?uri=ol-36-22-4449>. Acesso em: 17 de out.

2023.

LIU, Y. *et al.* Summary of chatgpt-related research and perspective towards the future of large language models. **Meta-Radiology**, p. 100017, 2023. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2950162823000176>. Acesso em: 20 de out. 2023.

LU, Z.; HUNT, B. R.; OTT, E. Attractor reconstruction by machine learning. **Chaos: An Interdisciplinary Journal of Nonlinear Science**, v. 28, n. 6, 2018. Disponível em: <https://pubs.aip.org/aip/cha/article/28/6/061104/914157>. Acesso em: 25 de nov. 2023.

LUKOŠEVIČIUS, Mantas. A practical guide to applying echo state networks. *In: Neural networks: tricks of the trade*. Heidelberg: Springer Berlin Heidelberg, 2012. p. 659–686. Disponível em: [https://link.springer.com/chapter/10.1007/978-3-642-35289-8\\_36](https://link.springer.com/chapter/10.1007/978-3-642-35289-8_36). Acesso em: 4 de ago. 2023.

MAASS, Wolfgang. Liquid state machines: motivation, theory, and applications. **Computability in context: computation and logic in the real world**. [s. l.]: World Scientific, 2011. p. 275–296. Disponível em: [https://www.worldscientific.com/doi/abs/10.1142/9781848162778\\_0008](https://www.worldscientific.com/doi/abs/10.1142/9781848162778_0008). Acesso em: 4 de ago. 2023.

MAASS, W.; NATSCHLÄGER, T.; MARKRAM, H. Real-Time Computing Without Stable States: A New Framework for Neural Computation Based on Perturbations. **Neural Computation**, v. 14, n. 11, p. 2531–2560, 2002. ISSN 0899-7667. Disponível em: <https://ieeexplore.ieee.org/abstract/document/6789852>. Acesso em: 10 de ago. 2023.

NAKAJIMA, Kohei. Reservoir computing: Theory, physical implementations, and applications. **IEICE Technical Report; IEICE Tech. Rep.**, v. 118, n. 220, p. 149-154, 2018. Disponível em: <https://ken.ieice.org/ken/paper/20180921t15x/eng/>. Acesso em: 15 de jan. 2022.

OLIVEIRA, E. R.; JULIANI, F. Reservoir computing: uma abordagem conceitual. **Revista de Ciências Exatas e Tecnologia**, v. 13, n. 13, p. 09–12, 2018. Disponível em: <https://exatastecnologias.pgsscogna.com.br/rcext/article/view/5424>. Acesso em: 24 de out. 2023.

PARLITZ, U.; JUNGE, L.; LAUTERBORN, W.; KOCAREV, L. Experimental observation of phase synchronization. **Physical Review E**, v. 54, n. 2, p. 2115–2117, 1996. Disponível em: <https://journals.aps.org/pre/abstract/10.1103/PhysRevE.54.2115>. Acesso em: 4 mai. 2022.

PATHAK, J.; HUNT, B.; GIRVAN, M.; LU, Z.; OTT, E. Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach. **Physical review letters**, v. 120, n. 2, p. 024102, 2018. Disponível em: <https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.120.024102>. Acesso em:

13 de jul. 2023.

PECORA, L. M.; CARROLL, T. L. Synchronization in chaotic systems. **Physical review letters**, Maryland, v. 64, n. 8, p. 821–824, 1990. Disponível em: <https://link.aps.org/doi/10.1103/PhysRevLett.64.821>. Acesso em: 7 dez. 2022.

PIKOVSKY, A.; ROSENBLUM, M. G.; KURTHS, J. Synchronization: A universal concept in nonlinear science. **American Journal of Physics**, v. 70, n. 6, p. 655–655, 2002. Disponível em: <https://pubs.aip.org/aapt/ajp/article-abstract/70/6/655/1040631/Synchronization-A-Universal-Concept-in-Nonlinear?redirectedFrom=fulltext>. Acesso em: 21 dez. 2022.

PYRAGAS, V.; PYRAGAS, K. Using reservoir computer to predict and prevent extreme events. **Physics Letters A**, v. 384, n. 24, p. 126591, 2020. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0375960120304588>. Acesso em: 20 de mar. 2023.

RACCA, A.; MAGRI, L. Data-driven prediction and control of extreme events in a chaotic flow. **Physical Review Fluids**, v. 7, n. 10, p. 104402, 2022. Disponível em: <https://journals.aps.org/prfluids/abstract/10.1103/PhysRevFluids.7.104402>. Acesso em: 22 de mar. 2023.

RASHID, T. **Make your own neural network**. North Charleston: CreateSpace Independent Publishing Platform, 2016. v. 29. Disponível em: [https://github.com/harshitkgupta/StudyMaterial/blob/master/Make%20Your%20Own%20Neural%20Network%20\(Tariq%20Rashid\)%20-%20%7BCHB%20Books%7D.pdf](https://github.com/harshitkgupta/StudyMaterial/blob/master/Make%20Your%20Own%20Neural%20Network%20(Tariq%20Rashid)%20-%20%7BCHB%20Books%7D.pdf). Acesso em: 15 de set. 2023.

SILVER, D. *et al.* Mastering the game of go with deep neural networks and tree search. **Nature**, v. 529, n. 7587, p. 484–489, 2016. Disponível em: <https://www.nature.com/articles/nature16961%7D>. Acesso em: 20 de out. 2023.

SORNETTE, D. Probability distributions in complex systems. **ArXiv Physic**, 2007. Disponível em: <https://arxiv.org/abs/0707.2194>. Acesso em: 18 de jan. 2023.

SORNETTE, D. Dragon-kings, black swans and the prediction of crises. **ArXiv Physic**, 2009. Disponível em: <https://arxiv.org/abs/0907.4290>. Acesso em: 19 de jan. 2023.

STEINER, P.; JALALVAND, A.; STONE, S.; BIRKHOLZ, P. Pycrn: A toolbox for exploration and application of reservoir computing networks. **Engineering Applications of Artificial Intelligence**, v. 113, p. 104964, 2022. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0952197622001713>. Acesso em: 12 de mar. 2023.

STROGATZ, S. H. **Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering**. Boca Raton: CRC press, 2018.

TANAKA, G.; YAMANE, T.; HÉROUX, J. B.; NAKANE, R.; KANAZAWA, N.; TAKEDA, S.; NUMATA, H.; NAKANO, D.; HIROSE, A. Recent advances in physical reservoir computing: A review. **Neural Networks**, v. 115, p. 100–123, 2019.

Disponível em:

<https://www.sciencedirect.com/science/article/pii/S0893608019300784>. Acesso em: 10 de set. 2023.

TRAKAS, D. N.; PANTELI, M.; HATZIARGYRIOU, N. D.; MANCARELLA, P. Spatial risk analysis of power systems resilience during extreme events. **Risk Analysis**, 2018. ISSN 0272-4332,1539-6924. Disponível em:

<http://doi.org/10.1111/risa.13220>. Acesso em: 10 de mai. 2023.

TROUVAIN, N. *et al.* Reservoirpy: an efficient and user-friendly library to design echo state networks. *In*: INTERNATIONAL CONFERENCE ON ARTIFICIAL NEURAL NETWORKS, 29., 2020, Bratislava. **Proceedings** [...]. Bratislava, 2020. p. 494–505. Disponível em: <https://inria.hal.science/hal-02595026>. Acesso em: 10 de jan. 2023.

WHO Covid-19 dashboard. **WHO Data**. World Health Organization, 2020. Disponível em: <https://covid19.who.int/>. Acesso em: 30 set. 2023.

WMO Statement on the State of the Global Climate in 2019. **WMO e-Library**. World Meteorological Organization, 2020. Disponível em: [https://library.wmo.int/index.php?lvl=notice\\_display&id=21777#.YqZBndLMI3F](https://library.wmo.int/index.php?lvl=notice_display&id=21777#.YqZBndLMI3F).

Acesso em: 08 jun. 2022.