
**Aprendizado de padrões EEG para prognóstico
precoce de pacientes em coma usando redes
echo state e redes neurais convolucionais**

Lucas Zago Bissaro



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Uberlândia
2021

Lucas Zago Bissaro

**Aprendizado de padrões EEG para prognóstico
precoce de pacientes em coma usando redes
echo state e redes neurais convolucionais**

Dissertação de mestrado apresentada ao Programa de Pós-graduação da Faculdade de Computação da Universidade Federal de Uberlândia como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Área de concentração: Ciência da Computação

Orientador: Murillo G. Carneiro

Coorientador: João B. Destro-Filho

Uberlândia

2021

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema de Bibliotecas da UFU, MG, Brasil.

B6231 Bissaro, Lucas Zago, 1994-
2021 Aprendizado de padrões EEG para prognóstico precoce de pacientes em coma usando redes echo state e redes neurais convolucionais [recurso eletrônico] / Lucas Zago Bissaro. - 2021.

Orientador: Murillo Guimarães Carneiro.

Co-orientador: João B. Destro-Filho.

Dissertação (mestrado) - Universidade Federal de Uberlândia.

Programa de Pós-Graduação em Ciência da Computação.

Modo de acesso: Internet.

Disponível em: <http://doi.org/10.14393/ufu.di.2022.5019>

Inclui bibliografia.

Inclui ilustrações.

1. Computação. I. Carneiro, Murillo Guimarães, 1988-, (Orient.). II. Destro-Filho, João B., 1970-, (Co-orient.). III. Universidade Federal de Uberlândia. Programa de Pós-Graduação em Ciência da Computação. IV. Título.

CDU: 681.3

André Carlos Francisco
Bibliotecário - CRB-6/3408



ATA DE DEFESA - PÓS-GRADUAÇÃO

Programa de Pós-Graduação em:	Ciência da Computação				
Defesa de:	Mestrado Acadêmico, 19/2021, PPGCO				
Data:	11 de agosto de 2021	Hora de início:	9h05min	Hora de encerramento:	12h10min
Matrícula do Discente:	11912CCP018				
Nome do Discente	Lucas Zago Bissaro				
Título do Trabalho:	Aprendizado de padrões EEG para prognóstico precoce de pacientes em coma usando redes echo state e redes neurais convolucionais				
Área de concentração:	Ciência da Computação				
Linha de pesquisa:	Inteligência Artificial				
Projeto de Pesquisa de vinculação:	-				

Reuniu-se, por videoconferência, a Banca Examinadora, designada pelo Colegiado do Programa de Pós-graduação em Ciência da Computação, assim composta: Professores Doutores: Marcelo Zanchetta do Nascimento - FACOM/UFU; Zhao Liang - FFCLRP/USP; João Batista Destro Filho - FEELT/UFU (coorientador) e Murillo Guimarães Carneiro - FACOM/UFU, orientador do candidato.

Os examinadores participaram desde as seguintes localidades: Zhao Liang - Ribeirão Preto/SP; Marcelo Zanchetta do Nascimento, João Batista Destro Filho e Murillo Guimarães Carneiro - Uberlândia/MG. O discente participou da cidade de Uberlândia/MG.

Iniciando os trabalhos o presidente da mesa, Prof. Dr. Murillo Guimarães Carneiro, apresentou a Comissão Examinadora e o candidato, agradeceu a presença do público, e concedeu ao Discente a palavra para a exposição do seu trabalho. A duração da apresentação do Discente e o tempo de arguição e resposta foram conforme as normas do Programa.

A seguir o senhor presidente concedeu a palavra, pela ordem sucessivamente, aos examinadores, que passaram a arguir o candidato. Ultimada a arguição, que se desenvolveu dentro dos termos regimentais, a Banca, em sessão secreta, atribuiu o resultado final, considerando o candidato:

Aprovado.

Esta defesa faz parte dos requisitos necessários à obtenção do título de Mestre.

O competente diploma será expedido após cumprimento dos demais requisitos, conforme as normas do Programa, a legislação pertinente e a regulamentação interna da UFU.

Nada mais havendo a tratar foram encerrados os trabalhos. Foi lavrada a presente ata que após lida e

achada conforme foi assinada pela Banca Examinadora.



Documento assinado eletronicamente por **Murillo Guimarães Carneiro, Professor(a) do Magistério Superior**, em 12/08/2021, às 10:53, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Zhao Liang, Usuário Externo**, em 12/08/2021, às 12:55, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **João Batista Destro Filho, Professor(a) do Magistério Superior**, em 12/08/2021, às 15:47, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Marcelo Zanchetta do Nascimento, Professor(a) do Magistério Superior**, em 15/08/2021, às 10:47, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site https://www.sei.ufu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **2942748** e o código CRC **AEC8E869**.

Dedico esta dissertação aos meus pais por todo esforço e apoio incondicional e por me ensinarem a valorizar a ciência e o estudo. Aos meus amigos que tentaram sempre me atrapalhar nas noites de estudo. Aos meus colegas de curso Vinícius e Lusmar, que me acompanharam por todo o processo. Por último mas não menos importante, sou eternamente grato ao meu orientador e professor Murillo G. Carneiro, pelo incentivo durante todo o projeto. Sua motivação foi mais que essencial para a conclusão desta dissertação.

Agradecimentos

Na realização da presente dissertação, contei com o apoio direto ou indireto de múltiplas pessoas e instituições às quais estou profundamente grata. Correndo o risco de injustamente não mencionar algum dos contributos quero deixar expresso os meus agradecimentos:

Ao meu orientador Murillo G. Carneiro, pelo incentivo, contribuição, apoio e disponibilidade que sempre demonstrou. Aqui lhe exprimo a minha gratidão.

Aos meus pais por sempre estarem presentes por mim, sempre me encorajando nos momentos de dificuldades, e me dando forças para seguir sempre em frente.

Aos meus colegas de curso, em especial, Vinicius e Lusmar por caminharem juntos comigo nesta jornada. Ao apoio e ajuda nos momentos menos fáceis, facilitando todo o processo com suas companhias.

Aos meus amigos Didi e Ramon, por mesmo com toda a distância, nunca deixarem de marcar minha vida com conversas motivadoras e alegres.

Aos meus amigos de Viçosa Bolota, Coala, Guttaishom, Ives, Pele, Despacho, por estarem comigo em todo caminho de toda minha jornada acadêmica. Sempre aconselhando (nem sempre os melhores conselhos) e nunca ficarem distantes mesmo quando a distância era grande.

A minha namorada um grande agradecimento, por ter segurado a barra quando eu estava pirando. Me fazendo ter o foco certo, me guiando na melhor direção.

A minha colega Camila, por toda a ajuda durante o desenvolvimento do projeto, auxiliando em todos os momentos, nos guiando para as melhores decisões, sendo assim, meu muito obrigado.

Ao CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico), pela bolsa de estudos e auxílio financeiro que possibilitou a dedicação integral ao programa de pós-graduação e a operacionalização do estudo.

À NVIDIA e ao seu programa *Accelerated Data Science GPU Grant*, o qual premiou nosso projeto com uma placa de vídeo de última geração para as nossas simulações computacionais.

*“A ciência de hoje é a tecnologia de amanhã.”
(Edward Teller)*

Resumo

O exame de Eletroencefalograma (EEG) capta os sinais elétricos do cérebro utilizando eletrodos posicionados ao longo do escalpo. Tais sinais podem ser utilizados na medicina para diversos fins, tais como o prognóstico de doenças. Especificamente, este trabalho investiga arquiteturas de aprendizado recorrente e profundo para o prognóstico de pacientes em coma (PPC) considerando dados EEG coletados em um hospital público brasileiro. Duas formulações para o problema são consideradas aqui. A primeira divide a base de dados em dois desfechos: favorável e desfavorável. A segunda considera também um terceiro desfecho relacionado ao prognóstico de morte encefálica, o qual pode contribuir com protocolos clínicos voltados para doação e gestão de órgãos. Diferente das abordagens mais utilizadas na literatura, as quais são baseadas na extração de atributos dos sinais EEG a partir de quantificadores, nessa dissertação, nós desenvolvemos modelos de aprendizado baseados em redes *echo state* (ESN) capazes de considerar os padrões sequenciais do exame a partir de arquiteturas de processamento mono e multi-sequenciais do EEG. Além disso, foi desenvolvido um modelo de representação espacial que se aproveita da posição dos eletrodos no exame enquanto mantém a resolução temporal e, os quais foram posteriormente utilizados nas arquiteturas de aprendizado profundo baseadas em redes neurais convolucionais (CNN). Vários experimentos foram realizados para avaliar as arquiteturas desenvolvidas. Do ponto de vista da análise de padrões sequenciais, os resultados mostram que nossa abordagem baseada em ESN profunda possui o melhor desempenho preditivo médio para dois e três desfechos quando comparada com as outras arquiteturas ESN desenvolvidas. Em relação à análise da representação espacial, os resultados evidenciam um progresso considerável dessa abordagem, sendo que as CNNs foram capazes de superar abordagens do estado-da-arte tanto para dois quanto para três desfechos. Assim, o presente trabalho estende a literatura relacionada ao PPC e abre novas perspectivas de apoio a equipe médica nessa tarefa.

Palavras-chave: Aprendizado profundo, aprendizado recorrente, computação de reser-

vatório, eletroencefalograma, prognóstico de coma, morte encefálica..

Abstract

The electroencephalogram (EEG) exam registers the brain's electrical signals using positioned electrodes over the scalp. The signals can be used in many medicine applications, such as the prognosis of diseases. In this research we investigate recurrent neural networks and deep learning architectures for the prognosis of comatose patients (PPC) from their EEG data, which were collected in a Brazilian public hospital. We considered two formulations for such a problem. The first aims to divide the target class into two outcomes: favorable and unfavorable. The second also considers a third outcome related to brain death, which have great potential to contribute to organ donation procedures. Unlike most common approaches in the literature, which use quantifiers to extract features of the EEG signals, in this dissertation we develop learning models based on echo state networks (ESN) capable of considering the sequential patterns of the exam by processing architectures mono and multi-sequential of EEG signals. In addition, a spatial representation model that takes advantage of the position of the electrodes during the exam while maintaining temporal resolution was developed and later used in our deep learning architectures based on convolutional neural networks (CNN). Several experiments were carried out to evaluate the developed architectures. From the viewpoint of sequential pattern analysis, the results showed that our deep ESN-based approach has the best average predictive performance for two and three outcomes when compared to other developed ESN architectures. Regarding the analysis of spatial representation, the results demonstrated considerable progress as our CNNs were able to outperform state-of-the-art approaches for both two and three outcomes. Moreover, the present work extends the literature related to PPC and opens new perspectives for supporting the medical team in this task.

Keywords: Deep Learning, recurrent learning, reservoir computing, electroencephalogram, comatose outcome, brain death..

Lista de ilustrações

Figura 1 – Esquema de um neurônio - perceptron. Fonte: Retirada de (HAYKIN, 2021)	35
Figura 2 – Esquema de rede da Perceptron multicamadas Fonte: Retirada de (RODRÍGUEZ, 2020)	36
Figura 3 – Funcionamento do filtro na camada convolucional. Fonte: Retirada de (DESHPANDE, 2020a)	41
Figura 4 – Seleção de um pedaço da imagem para passagem do filtro. Fonte: Retirada de (DESHPANDE, 2020a)	41
Figura 5 – Transformação da imagem em uma matriz numérica. Fonte: Retirada de (DESHPANDE, 2020a)	41
Figura 6 – Aplicação do filtro na matriz numérica que representa uma imagem semelhante a desejada pelo filtro. Fonte: Retirada de (DESHPANDE, 2020a)	42
Figura 7 – Aplicação do filtro na matriz numérica que representa uma imagem diferente da desejada pelo filtro. Fonte: Retirada de (DESHPANDE, 2020a)	42
Figura 8 – Funcionamento do <i>stride</i> de tamanho 1, a seleção exibida na matriz da esquerda, se torna o ponto de mesma cor na matriz da direita. A distancia entre os filtros é o <i>stride</i> . Fonte: Retirada de (DESHPANDE, 2020b)	42
Figura 9 – Funcionamento do <i>stride</i> de tamanho 2, a seleção exibida na matriz da esquerda, se torna o ponto de mesma cor na matriz da direita. A distancia entre os filtros é o <i>stride</i> . Fonte: Retirada de (DESHPANDE, 2020b)	42
Figura 10 – Demonstração do padding, par alcançar todos os pontos nos extremos, a matriz é rodeada de valores 0. Fonte: Retirada de (DESHPANDE, 2020b)	43

Figura 11 – Camada <i>Pooling</i> , a imagem mostra um <i>max pooling</i> , na matriz da esquerda, de cada 4 pontos de mesma cor, somente o maior é utilizado, resultando na matriz da direita. Fonte: Retirada de (DESHPANDE, 2020b)	44
Figura 12 – Exemplo do traçado EEG Fonte: Retirada de (MAESTROVIRTUALE, 2021)	46
Figura 13 – Localização de eletrodos segundo o sistema 10-20. Fonte: Retirada de Retirada de (21. . . . , 2021).	47
Figura 14 – Estrutura espacial para modelagem da matriz a partir do sistema de organização 10-20.	52
Figura 15 – Passos desde o captura do exame até a geração da imagem final utilizada pela rede.	53
Figura 16 – Exemplo das estruturas do reservoir geradas com modelos de rede distintos. a) Na regular, os neurônios são conectados com seus k-vizinhos mais próximos. b) Na pequeno mundo, cada neurônio tem uma probabilidade $p = 0.1$ de trocar seus links depois de terem sido conectadas pelos seus k-vizinhos mais próximos; e c) Na aleatória, as conexões são totalmente aleatórias.	63
Figura 17 – Diagrama de diferença crítica obtida pelo teste post-hoc de Nemenyi, considerando as diferentes configurações de parâmetros na estrutura do reservoir.	66
Figura 18 – Diagrama de diferença crítica obtida pelo teste post-hoc de Nemenyi, considerando as diferentes técnicas de classificação para a camada readout.	68
Figura 19 – Diagrama de diferença crítica obtida a partir do teste post-hoc de Nemenyi considerando a randômica, a pequeno mundo e a regular.	68
Figura 20 – Estrutura da rede MS-ESN-1, onde os dados são colocados em sequência no <i>Reservoir</i> , e utilizado nos algoritmos de classificação.	77
Figura 21 – Estrutura da rede MS-ESN-2, o sinal de cada eletrodo é quebrado em pedaços separadamente e utilizado no <i>Reservoir</i> , depois é agrupado e usado pelos algoritmos de classificação.	78
Figura 22 – Estrutura da rede MS-ESN-3, a serie de cada eletrodo é dividida em pedaços, os pedaços são agrupados de acordo com seu índice, na ordem dos eletrodos, em seguida passam pelo <i>Reservoir</i> e são classificados pelos algoritmos.	80
Figura 23 – Estrutura da rede MS-ESN-4, cada eletrodo é dividido em pedaços que em sequência é transformado cada um pelo seu próprio <i>Reservoir</i> , as saídas são agrupadas e utilizadas nos classificadores.	81

Figura 24 – Estrutura da rede MS-DeepESN, os eletrodos divididos em pedaços são transformados em cascata por diferentes reservatórios, cada reservatório recebe a saída do reservatório anterior agrupada com o novo eletrodo, a saída do ultimo reservatório é utilizada pelos classificadores.	82
Figura 25 – Representação dos eletrodos na matriz 5x5	90
Figura 26 – Representação dos eletrodos na matriz 7x7	90
Figura 27 – Modelo de rede convolucional C1 utilizada pelos CNN-Padrão e CNN-Concat	92
Figura 28 – Modelo de rede convolucional C2 utilizada pelos CNN-Padrão e CNN-Concat	93
Figura 29 – Modelo de rede convolucional utilizada pelo CNN-Multi.	94

Lista de tabelas

Tabela 1 – Nomes e casos das diferentes frequências capturadas pelo exame EEG.	48
Tabela 2 – Demografia dos pacientes comatosos analisados.	56
Tabela 3 – Lista de parâmetros testados para cada técnica no readout e o parâmetro selecionado com melhor resultado.	59
Tabela 4 – Resultados utilizando os métodos considerados estado-da-arte com o EEG utilizando 2 desfechos na base de dados	59
Tabela 5 – Resultados utilizando os métodos considerados estado da arte com o EEG utilizando 3 desfechos na base de dados	60
Tabela 6 – Metadados das bases de dados consideradas no estudo.	64
Tabela 7 – Lista de parâmetros testados para cada técnica no readout e o parâmetro selecionado com melhor resultado.	65
Tabela 8 – F1 Macro para os oito classificadores com as cinco bases de dados reais. Em negrito estão os melhores resultados para cada base, e sublinhado estão os melhores resultados para configuração de rede em cada método.	67
Tabela 9 – P-valores encontrados para o teste de Wilcoxon, quando considera os resultados dos três melhores métodos na readout: GSVM, kNN, LSVM.	69
Tabela 10 – F1 Macro para os 3 melhores classificadores na camada readout para as cinco bases de dados reais, considerando dois modelos: Com (Res.) e sem (No-Res.) a estrutura do reservoír. Em negrito estão os melhores resultados para cada base, e sublinhado denota qual modelo prove o melhor resultado.	69
Tabela 11 – Resultados do reservoír utilizando um único eletrodo e com a base de dados dividida em dois desfechos.	71
Tabela 12 – Resultado comparando o melhor método de aprendizado de máquina sem o uso de aprendizado profundo, e a melhor execução da rede recorrente utilizando somente um eletrodo para dois desfechos.	72
Tabela 13 – Resultados do reservoír utilizando um único eletrodo e com a base de dados dividida em três desfechos.	72

Tabela 14 – Resultado comparando o melhor método de aprendizado de máquina sem o uso de aprendizado profundo, e a melhor execução da rede recorrente utilizando somente um eletrodo para três desfechos.	73
Tabela 15 – Acurácia, precision, recall F1-macro com a melhor configuração de cada método para os testes sem <i>reservoir</i> e para as 5 estruturas analisadas, com a base de dados dividida em 2 desfechos.	84
Tabela 16 – P-values do teste Wilcoxon considerando os três classificadores readout avaliados para dois desfechos. Considerando um nível de confiança de 95% ($\alpha = 0.05$), os símbolos $>$, $<$ e \approx indicam que a variável da coluna é respectivamente melhor, pior ou equivalente à variável da linha. . . .	85
Tabela 17 – P-values do teste Wilcoxon considerando os diferentes valores de divisões do exame para dois desfechos. Considerando um nível de confiança de 95% ($\alpha = 0.05$), os símbolos $>$, $<$ e \approx indicam que a variável da coluna é respectivamente melhor, pior ou equivalente à variável da linha.	85
Tabela 18 – Acurácia, precision, recall F1-macro com a melhor configuração de cada método para os testes sem <i>reservoir</i> e para as 5 estruturas analisadas, com a base de dados dividida em 3 desfechos.	86
Tabela 19 – P-values do teste Wilcoxon considerando os três classificadores readout avaliados para três desfechos. Considerando um nível de confiança de 95% ($\alpha = 0.05$), os símbolos $>$, $<$ e \approx indicam que a variável da coluna é respectivamente melhor, pior ou equivalente à variável da linha. . . .	87
Tabela 20 – P-values do teste Wilcoxon considerando os diferentes valores de divisões do exame para três desfechos. Considerando um nível de confiança de 95% ($\alpha = 0.05$), os símbolos $>$, $<$ e \approx indicam que a variável da coluna é respectivamente melhor, pior ou equivalente à variável da linha. 87	87
Tabela 21 – Acurácia, precision, recall F1-macro com a melhor configuração de cada arquitetura anterior e para a melhor configuração dos testes com os modelos convolucionais, sem a heurística e com(*), com a base de dados dividida em dois desfechos.	98
Tabela 22 – Resultado do teste Wilcoxon comparando o uso ou não da estratégia de vizinhança com a base de dados dividida em 2 desfechos.	98
Tabela 23 – Resultado do teste Wilcoxon comparando o tamanho do Mapa com a base de dados dividida em 2 desfechos.	99
Tabela 24 – Resultado do teste Wilcoxon comparando o valor utilizado na heurística com a base de dados dividida em 2 desfechos.	99
Tabela 25 – Resultado do teste Wilcoxon comparando os modelos de rede C1 e C2 para a estrutura CNN-Padrão com base de dados dividida em 2 desfechos. 99	99

Tabela 26 – Resultado do teste Wilcoxon comparando os algoritmos para a estrutura CNN-Concat com base de dados dividida em 2 desfechos.	100
Tabela 27 – Resultado do teste Wilcoxon comparando 10 e 20 divisões para a estrutura CNN-Concat com base de dados dividida em 2 desfechos.	100
Tabela 28 – Acurácia, precision, recall F1-macro com a melhor configuração de cada arquitetura anterior e para a melhor configuração dos testes com os modelos convolucionais, com a base de dados dividida em 3 desfechos.	100
Tabela 29 – Resultado do teste Wilcoxon comparando o uso ou não da estratégia de vizinhança com a base de dados dividida em 3 desfechos.	101
Tabela 30 – Resultado do teste Wilcoxon comparando o tamanho do Mapa com a base de dados dividida em 3 desfechos.	101
Tabela 31 – Resultado do teste Wilcoxon comparando os modelos de rede C1 e C2 com a base de dados dividida em 3 desfechos.	101
Tabela 32 – Resultado do teste Wilcoxon comparando os algoritmos com a base de dados dividida em 3 desfechos.	102
Tabela 33 – Resultado do teste Wilcoxon comparando 10 e 20 divisões para a estrutura CNN-Concat com base de dados dividida em 3 desfechos.	102

Sumário

1	INTRODUÇÃO	25
1.1	Hipóteses e Objetivos	28
1.2	Organização da Dissertação	29
2	FUNDAMENTAÇÃO E REVISÃO BIBLIOGRÁFICA	31
2.1	Aprendizado de Máquina Supervisionado	31
2.1.1	Aprendizado tradicional	33
2.1.2	Aprendizado profundo	37
2.2	Análise de Sinais de Eletroencefalograma (EEG)	45
2.2.1	Definições e conceitos	46
2.2.2	Abordagens convencionais de análise	47
2.2.3	Prognóstico de pacientes em coma	48
2.2.4	Aprendizado de sinais EEG de pacientes em coma	49
2.2.5	Aprendizado profundo em sinais EEG	51
3	APRENDIZADO TRADICIONAL PARA O PROGNÓSTICO DE PACIENTES EM COMA (PPC)	55
3.1	Descrição da Base de Dados	55
3.2	Ambiente Experimental e Métricas de Desempenho Preditivo .	57
3.3	Uma Abordagem Benchmark para o PPC	58
3.3.1	Resultados PPC para Dois Desfechos	59
3.3.2	Resultados PPC para Três Desfechos	59
4	REDES <i>ECHO STATE</i> REGULARES PARA O PPC	61
4.1	Análise comparativa de ESNs em termos de reservóir e readout	61
4.1.1	Modelos de rede de reservatório	62
4.1.2	Classificadores readout	62
4.1.3	Parâmetros considerados	63

4.2	Resultados em Bases de Dados Reais	63
4.2.1	Análise dos classificadores readout	65
4.2.2	Análises do modelo de rede	68
4.2.3	Análise da contribuição do reservoir	69
4.3	Aplicação para o PPC usando sinais de EEG	70
4.3.1	Descrição do Modelo	70
4.3.2	Resultados PPC (por eletrodo) para Dois Desfechos	71
4.3.3	Resultados PPC (por eletrodo) para Três Desfechos	72
5	REDES ECHO STATE MULTI-SEQUÊNCIAS PARA O PPC	75
5.1	Visão Geral	75
5.2	Descrição do Modelo	76
5.2.1	MS-ESN-1	76
5.2.2	MS-ESN-2	77
5.2.3	MS-ESN-3	79
5.2.4	MS-ESN-4	80
5.2.5	MS-DeepESN	80
5.3	Resultados PPC para Dois Desfechos	82
5.4	Resultados PPC para Três Desfechos	84
6	REDES NEURAIIS CONVOLUCIONAIS PARA O PPC	89
6.1	Resolução Espacial dos Sinais EEG	89
6.2	Arquiteturas de CNN para uma matriz de entrada	91
6.3	Arquitetura de CNN para múltiplas matrizes de entrada	91
6.4	Descrição dos modelos de CNN	95
6.4.1	CNN-Padrão	95
6.4.2	CNN-Concat	95
6.4.3	CNN-Multi	96
6.4.4	Heurística de atenção	96
6.5	Resultados PPC para Dois Desfechos	97
6.6	Resultados PPC para Três Desfechos	100
7	CONCLUSÃO	103
7.1	Principais Contribuições	105
7.2	Trabalhos Futuros	105
7.3	Contribuições em Produção Bibliográfica	106
REFERÊNCIAS		107

Introdução

De acordo com o Ministério da Saúde, em 2019 existiam por volta de 36000 pessoas no Brasil ativas na lista de transplante de órgãos. Infelizmente 2851 pessoas morreram em 2018 à espera de um transplante. De acordo com a Associação Brasileira de Transplante de Órgãos (ABTO), em 2018 o Brasil teve somente 3531 doadores dos 10778 potenciais (morte cerebral ou cardíaca) notificados (ABTO, 2021).

A maior parte dos transplantes atualmente utiliza órgãos de doadores que tem batimentos cardíacos após morte cerebral (MCKEOWN; BONSER; KELLUM, 2012). De acordo com a Resolução 2.173/2017 do Conselho Federal de Medicina, morte cerebral é uma condição clínica de coma irreversível de causa conhecida com ausência de reflexos no tronco cerebral e apneia. Entretanto, uma variedade de alterações fisiológicas ocorrem durante e após o processo de morte cerebral, tais como alterações inflamatórias e hormonais generalizadas, mudanças agudas, que podem afetar a função do órgão doador e aumentar sua propensão à rejeição (PRATSCHE et al., 1999).

Assim, a gestão da qualidade dos órgãos é crucial para um transplante bem-sucedido. Além disso, a abordagem da família doadora deve ser planejada e conduzida por profissionais bem treinados, seguindo um protocolo médico clínico. É senso comum na literatura que o prognóstico precoce da morte encefálica pode melhorar a quantidade e a qualidade dos órgãos doados (MCKEOWN; BONSER; KELLUM, 2012).

Existem diversos exames que podem apoiar os médicos em um prognóstico de morte encefálica. Um dos exames mais sofisticados é a ressonância magnética funcional (fMRI), cuja principal vantagem é prover uma representação espacial do nível de oxigenação em diferentes regiões do cérebro. Por outro lado, é um equipamento incapaz de monitorar a atividade cerebral em função do tempo e de custo bastante elevado, sendo assim considerado fora de realidade para a grande maioria dos hospitais brasileiros (MULERT; LEMIEUX, 2009).

Outro exame conhecido é o eletroencefalograma (EEG), que monitora e grava atividades elétricas captando sinais de diferentes frequências em várias regiões do cérebro (HERWIG; SATRAPI; SCHÖNFELDT-LECUONA, 2003; RAMOS, 2017). Entre as prin-

principais vantagens do EEG destaca-se a natureza não invasiva do exame, a resolução temporal da atividade cerebral e o baixo custo, os quais tornam esse tipo de exame amplamente disponível, inclusive no Sistema Único de Saúde (SUS) do Brasil. As principais desvantagens do exame incluem a falta de uma representação espacial (MULERT; LEMIEUX, 2009) bem como a suscetibilidade a diferentes tipos de ruído nos registros capturados (ROMO-VAZQUEZ et al., 2007).

Nesse estudo estamos particularmente interessados na análise das atividades cerebrais de pacientes em estado de coma. Tal condição se caracteriza pela falta ou baixo nível de consciência de um paciente (GERRARD; ZAFONTE; GIACINO, 2014; PERRI et al., 2014). Diferente da morte encefálica, a qual se caracteriza como um quadro de morte cerebral irreversível, a condição de coma pode evoluir para dois grupos de desfechos: favorável e desfavorável. O grupo favorável diz respeito a pacientes com bons resultados neurológicos como recuperação completa ou incapacidade moderada, enquanto o grupo desfavorável representa incapacidade severa, vigília sem resposta (estado vegetativo) ou morte (BANSAL et al., 2005; JENNETT, 2005). Nesse cenário, o exame EEG é de grande relevância, pois permite uma análise funcional das atividades cerebrais dos pacientes, podendo inclusive contribuir em decisões e intervenções terapêuticas bem como na prevenção de complicações.

Atualmente esse processo de avaliação dos sinais EEG é normalmente realizado por um médico que busca padrões conhecidos. Existem vários desafios relacionados à tarefa, como a duração do exame, em que 20 minutos pode ser equivalente a uma série temporal com representação de mais de 100 mil pontos. Logo pode ser extremamente cansativo para o médico que realiza a avaliação. Além disso, vários exames são avaliados por dia, o que pode resultar em maior dificuldade devido ao cansaço (KANDA et al., 2014). Outra dificuldade diz respeito à quantidade de ruído que o exame pode conter, podendo ser amenizado utilizando recursos computacionais avançados que também não estão disponíveis para os médicos. Por fim os padrões também nem sempre estão nítidos e podem ser complexos demais para serem percebidos a olho nu (CAPRON; KASS, 1972).

O uso de Inteligência Artificial (IA) é cada vez maior na área da saúde, com diferentes abordagens voltadas para o treinamento de algoritmos capazes de identificar e reconhecer os padrões e apoiar diagnósticos a partir de imagens e exames (TRAN et al., 2019). As técnicas atuais de aprendizado de máquina possibilitam lidar com uma variedade de formatos de entradas de dados, como imagens (JIANG; TRUNDLE; REN, 2010), vetores de dados (ZHAO et al., 2019), sinais (YILDIRIM et al., 2018), etc.

Em relação ao reconhecimento de padrões a partir de sinais, várias abordagens têm sido desenvolvidas baseadas nas saídas dos mais variados equipamentos e exames (BEGUM; BARUA; AHMED, 2014; BALOGLU et al., 2019). Nessa pesquisa, buscamos especificamente os sinais EEG capturados a partir da análise das atividades cerebrais, os quais já se mostraram aplicáveis para auxiliar em vários problemas da literatura, tais

como o diagnóstico e acompanhamento da evolução de doenças, como Parkinson (VANEGAS et al., 2018), Alzheimer (TRAMBALIO et al., 2011), esquizofrenia (SABETI et al., 2011), epilepsia (SHOEB; GUTTAG, 2010), etc. Estes mesmos sinais podem ser utilizados para apoiar os médicos no prognóstico de pacientes em coma (CHEN et al., 2008).

Abordagens relacionadas às análises de EEG de pacientes comatosos na literatura costumam se basear majoritariamente no desenvolvimento de procedimentos e quantificadores para extração de atributos dos sinais, além de análises estatísticas convencionais (KUSTERMAN et al., 2019; LEHEMBRE et al., 2012; HERMANS et al., 2016; NOIRHOMME et al., 2014). Nesse sentido, vários quantificadores foram projetados e avaliados por trabalhos correlatos, tais como coerência (SCHORR et al., 2016), potência absoluta e relativa (TOLONEN et al., 2018), potência do sinal EEG em bandas (GOLKOWSKI et al., 2017; RAMOS, 2017) e padrão surto-supressão (BAGNATO et al., 2010), especialmente considerando a sua correlação com escalas mundialmente adotadas para avaliar o nível de consciência em pacientes com lesões cerebrais agudas, como a escala de coma Glasgow (JENNETT, 2005).

Para análise desses quantificadores, a maioria das abordagens considerou métodos estatísticos como o teste t e a correlação com o desfecho do paciente. Mais recentemente, vários algoritmos tradicionais de aprendizado de máquina também foram avaliados a partir de uma base de dados obtida a partir de um conjunto de quantificadores (RAMOS; CARNEIRO; DESTRO-FILHO, 2021).

Apesar do progresso recente alcançado no problema do Prognóstico de Pacientes em Coma (PPC), o processo de projetar e desenvolver novos quantificadores é demorado e dependente dos dados disponíveis, com poucas análises voltadas para questões de generalização desses métodos para outros domínios. Ademais, a maioria dos quantificadores não tira proveito do aspecto temporal sequencial dos dados, o qual é considerado uma característica saliente do exame.

Nesse sentido, o presente trabalho traz uma investigação de arquiteturas de aprendizado baseadas em redes *echo state* (ESN) para contribuir com a análise dos padrões sequenciais dos sinais EEG no problema de PPC. A ESN é uma estrutura de aprendizado recorrente, composta por um *reservoir*, normalmente denotado por uma rede de pesos aleatórios, e uma função de leitura (*readout*) (LUKOŠEVIČIUS; JAEGER, 2009). O *reservoir* é responsável por mapear os dados de entrada para um espaço de alta dimensionalidade, enquanto a função de leitura é responsável por aprender os padrões nesse novo espaço. Uma das principais vantagens do *reservoir* em relação as outras redes recorrentes é a simplicidade, uma vez que os pesos internos da rede são fixos, ou seja, não é necessária qualquer etapa de treinamento ou procedimento para atualização de pesos (TANAKA et al., 2019).

Outra lacuna que pretendemos cobrir com esse trabalho diz respeito a uma própria

limitação do EEG relacionada à falta de uma representação espacial dos dados. Se por um lado o EEG possui vantagem sobre vários outros exames clínicos por prover uma resolução temporal das atividades neurológicas do paciente, por outro lado, ele está em desvantagem quando se considera a falta de uma resolução espacial capaz de representar essas atividades. Nesse sentido, esta pesquisa investiga abordagens capazes de prover uma representação espacial do exame a partir da resolução temporal e da posição dos eletrodos no escalpo, os quais normalmente são dispostos considerando um modelo de organização chamado sistema 10-20 (HERWIG; SATRAPI; SCHÖNFELDT-LECUONA, 2003; RAMOS, 2017). Para tirar proveito de tal representação, também são investigadas arquiteturas de aprendizado baseadas em redes neurais convolucionais (do inglês, Convolutional Neural Network - CNN). Tais modelos são considerados o estado-da-arte para a extração de representações que são robustas à rotação parcial e deformação de dados de entrada (LECUN; BENGIO; HINTON, 2015), tendo sido inclusive aplicadas com sucesso para predição de convulsões usando sinais EEG (MIROWSKI et al., 2009).

Em síntese, até onde sabemos, este trabalho representa uma das primeiras pesquisas para a modelagem preditiva do prognóstico de pacientes em coma utilizando técnicas de aprendizado profundo a partir de sinais de EEG. Especificamente, o aprendizado recorrente dos sinais a partir de arquiteturas de ESN tem sua motivação principal na possibilidade de extrair padrões sequenciais do exame de maneira simples, a partir da expansão não-linear dos dados, enquanto o aprendizado a partir de arquiteturas de CNN tem sua motivação na possibilidade de aprender representações robustas para o desfecho do coma a partir da resolução espacial do exame. Ademais, também estendemos a literatura por tratar a morte encefálica como um terceiro desfecho, separado dos outros desfechos do grupo desfavorável. Dessa forma, o presente trabalho realiza um esforço importante para contribuir com o prognóstico de morte encefálica de pacientes em coma, especialmente no intuito de fornecer uma indicação precisa e antecipada desse desfecho, com vistas a aumentar o número de doadores e também permitir uma melhor gestão da qualidade dos órgãos doados.

1.1 Hipóteses e Objetivos

Duas hipóteses são investigadas nessa dissertação, a saber:

1. A análise das características sequenciais dos sinais EEG a partir de arquiteturas de aprendizado recorrente, baseadas em redes *echo state* contribui para o prognóstico de pacientes em coma em termos de desempenho preditivo;
2. O prognóstico de pacientes em coma usando sinais EEG é mais preciso utilizando uma resolução espacial do exame, modelada a partir de arquiteturas de aprendizado profundo baseadas em redes neurais convolucionais.

O objetivo geral deste trabalho é o desenvolvimento de modelos de aprendizado profundo recorrentes e convolucionais para o prognóstico precoce de pacientes em coma a partir do exame de eletroencefalograma considerando os grupos de desfecho favorável, desfavorável e morte encefálica. Dessa forma, os objetivos específicos são:

- ❑ Projetar e desenvolver arquiteturas de aprendizado recorrente, baseadas em redes *echo state* capazes de considerar as características sequenciais e multi-sequenciais dos sinais EEG no processo de modelagem preditiva para o PPC.
- ❑ Desenvolver métodos capazes de prover uma resolução espacial dos sinais EEG tanto a partir da resolução temporal quanto da posição dos eletrodos no escalpo;
- ❑ Desenvolver arquiteturas de aprendizado profundo baseadas em redes neurais convolucionais capazes de aprender representações robustas para o PPC a partir das resoluções espaciais do exame.
- ❑ Avaliar comparativamente as soluções tradicionais, recorrentes e convolucionais desenvolvidas para o PPC em função de dois (favorável e desfavorável) e três (favorável, desfavorável e morte encefálica) desfechos.

1.2 Organização da Dissertação

Este documento é dividido em sete capítulos que exibem todo o desenvolvimento do projeto. O primeiro capítulo é a introdução, que apresenta as hipóteses e objetivos do trabalho e as contribuições realizadas. O segundo capítulo é a fundamentação teórica e revisão bibliográfica que explica conteúdos chave para o entendimento da pesquisa e alguns trabalhos relacionados que foram utilizados de inspiração nos modelos propostos. O terceiro capítulo apresenta uma descrição da base de dados utilizada em todo o trabalho, exhibe a visão geral e resultados de um método proposto na literatura e utilizado como baseline para as outras arquiteturas. O quarto capítulo realiza um estudo sobre a rede *echo state*, sua estrutura e os classificadores utilizados na camada *readout*, utilizando a mesma rede *echo state* são realizados testes e apresentados resultados analisando separadamente os principais eletrodos. O quinto capítulo define as arquiteturas que utilizam a rede *echo state* com todos os eletrodos e discute seus resultados. O sexto capítulo define uma nova representação para os exames, três modelos de redes convolucionais diferentes, uma heurística para escolha de resultado final e discute os resultados dos testes. O sétimo capítulo apresenta a conclusão com as descobertas da pesquisa, os trabalhos futuros a serem desenvolvidos e as contribuições em produção bibliográfica.

Fundamentação e Revisão bibliográfica

O eletroencefalograma é um exame comumente utilizado na medicina (CASSON et al., 2018), com diversas aplicações práticas, e já começou a ser utilizada junto da inteligência artificial para a detecção de diversas doenças como Alzheimer (SIMPRAGA et al., 2017), Parkinson (OH et al., 2020), demência (IERACITANO et al., 2020), epilepsia (KUMAR; KOLEKAR, 2014) e o coma (TJEPKEMA-CLOOSTERMANS et al., 2019).

Neste capítulo apresentamos uma explicação mais detalhada sobre aprendizado de máquina e análise de sinais de eletroencefalograma. Também apresentaremos alguns trabalhos relacionados, utilizados como motivação ou comparação com os modelos desenvolvidos na dissertação.

2.1 Aprendizado de Máquina Supervisionado

Aprendizado de máquina é uma área em constante evolução, que pode ser definida como o uso de informações passadas para fazer previsões precisas ou melhorar performances (MOHRI; ROSTAMIZADEH; TALWALKAR, 2018), onde algoritmos computacionais são projetados para emular a inteligência humana aprendendo com informações de exemplo. As técnicas de aprendizado de máquina tem sido utilizadas em diversas aplicações com sucesso desde reconhecimento de padrões, visão computacional, engenharia de naves espaciais, finanças, entretenimento e biologia computacional até aplicações biomédicas e médicas (NAQA; MURPHY, 2015).

O aprendizado de máquina é dividido em três categorias (SATHYA; ABRAHAM, 2013):

- **Aprendizado supervisionado:** cada objeto contém um vetor de atributos (x_1, x_2, \dots, x_n) e um rótulo y já predefinido. O aprendizado supervisionado aprende associando os rótulos aos objetos, criando um modelo, que se estenda a casos ainda não vistos com precisão. Onde novos objetos diferentes dos utilizados para treinamento, gerem um rótulo y_{target} que se aproxime ao máximo do rótulo y correto para

aquela instância. Por exemplo, é possível criar um modelo para identificar espécies de folhas a partir de um conjunto de imagens e seus respectivos rótulos.

□ **Aprendizado por reforço:** diferente dos outros aprendizados, o aprendizado por reforço interage com o ambiente, gerando ações (a_1, a_2, \dots, a_n) e essas ações afetam o estado, o ambiente, gerando recompensas (r_1, r_2, \dots, r_n), então não basta o método simplesmente dizer qual atitude tomar, mas deve aprender qual ação leva ao melhor resultado, a partir da tentativa e erro. Em um circuito-fechado, o método é executado diversas vezes tomando diferentes atitudes ao longo do tempo, onde cada uma leva a uma recompensa. A pontuação final equivale a qualidade da solução. Depois de diversas vezes a tendência é haver uma melhora do modelo, que consequentemente deve conseguir uma pontuação maior. Por exemplo, pode ser utilizado para ensinar um computador a jogar, penalizando as jogadas ruins e recompensando as boas (SUTTON; BARTO, 2018).

□ **Aprendizado não supervisionado:** quando se recebe vetores de objetos (x_1, x_2, \dots, x_n), sem que se um rótulo ou nem uma recompensa associada ao objeto, se tem o aprendizado não supervisionado. É difícil imaginar como se utiliza um método para aprender sem que se tenha como o guiar, mas ele tem objetivos que não necessitam desse auxílio. Por exemplo, quando se quer agrupar os dados de acordo com sua semelhança, ou usar método que diminui as dimensões dos dados, somente o próprio conjunto de dados é necessário (GHAHRAMANI, 2003). Pode ser utilizado para criar perfis de clientes a partir de características similares, por exemplo.

O aprendizado supervisionado é constituído de uma entrada $\mathbf{u}(n) \in \mathbb{R}^{N_u}$ e uma classe associada $y^{target}(n) \in \mathbb{R}^{N_y}$, $n = 1, 2, \dots, T$, sendo T o número de objetos da base de dados. A tarefa é utilizar um modelo representado pela função $\mathbf{f}(n)$, que considere os atributos de n para gerar um rótulo $\mathbf{y}(n)$ que se aproxime ao máximo de $\mathbf{y}^{target}(n)$. A qualidade do modelo é medida por um erro $E(\mathbf{y}, \mathbf{y}^{target})$.

No aprendizado supervisionado, também é possível separar na classificação da regressão. A diferença entre os dois grupos reside basicamente que na classificação mapeia sua entrada $\mathbf{u}(\mathbf{n})$ em uma saída y no domínio de classes pre-existentes, enquanto a regressão mapeia sua entrada $\mathbf{u}(\mathbf{n})$ em uma saída y pertencente ao domínio dos valores reais (NASTESKI, 2017).

Apesar dessa diferença, é comum que nos problemas de classificação, a saída y seja mapeada como um valor real, e convertida ao final em uma classe ou a saída final seja a probabilidade de pertencer a cada classe, sendo assim também um valor real que é convertido ao rótulo certo (AWAD; KHANNA, 2015).

2.1.1 Aprendizado tradicional

Quando se utiliza um algoritmo para aprendizado de máquina, existe o conjunto de dados utilizado como base para novos casos. Porém, a forma como esse conjunto é utilizado, pode variar sendo dividido em duas categorias: o aprendizado preguiçoso ou aprendizado ansioso (GALVÁN et al., 2009). O aprendizado preguiçoso diz respeito a técnicas que utilizam o conjunto de dados para classificar os novos casos. Ex: k vizinhos próximos (COVER; HART, 1967). O aprendizado ansioso cria um modelo a partir dos dados e utiliza este modelo para a classificação. O processo de criação deste modelo varia de técnica para técnica, e é chamado de treinamento. Ex: Máquina de vetores de suporte (BURGES, 1998).

2.1.1.1 K vizinhos mais próximos

Com um algoritmo preguiçoso, não é necessária criação de modelo na fase de treinamento. O próprio conjunto de dados é utilizado para classificação de novos objetos.

O algoritmo é relativamente simples e muito poderoso, conseguindo lidar com espaços não lineares (COVER; HART, 1967). O modelo tem um parâmetro K que define o número de vizinhos mais próximos a serem observados, e uma função de distância d que precisa respeitar o espaço métrico (SANTANA, 2012) definido por:

- $d(x, y) \geq 0$;
- $d(x, y) = d(y, x)$;
- $d(x, y) = 0$ se, e somente se, $x = y$;
- $d(x, z) \leq d(x, y) + d(y, z)$; (desigualdade triangular)

geralmente a métrica utilizada é a distância euclidiana (DANIELSSON, 1980) definida como:

$$D(x, y) = \sqrt{\sum_{k=1}^D (x_k - y_k)^2}, \quad (1)$$

sendo D o número de dimensões de cada ponto.

Definido o K . Quando um novo objeto chega, calcula-se a distância para todos os objetos contidos na base de dados pegando o K mais próximos. O rótulo dado ao novo objeto é a moda (ou seja, o elemento que mais aparece entre eles) dos K resultados.

2.1.1.2 Redes Neurais

As RNAs são sistemas computacionais baseados no funcionamento do cérebro humano (SHANMUGANATHAN, 2016), e a partir de conjunto de amostras é possível automatizar decisões sem que o mesmo seja explicitamente programado para tal, somente utilizando os exemplos de base.

As redes são feitas a partir de elementos que simulam os neurônios, e ligações que propagam os sinais imitando o funcionamento das sinapses entre neurônios cerebrais. Nas RNAs as conexões tem pesos que definem como as entradas são propagadas, uma função de ativação que diz como esses valores são recebidos e passados para frente, e os neurônios de saída, que definem o resultado da rede. (GARDNER; DORLING, 1998)

O conjunto de dados é utilizado para treinamento com a intenção de mapear cada objeto O para um rótulo pré-definido Y , a representação na rede neural é um vetor V de M valores, onde cada valor representa uma classe na qual a instância pode pertencer, o vetor contém valor 1 na posição referente o rótulo original do objeto e 0 para os outros.

A rede mapeia os objetos a partir de uma função $f : O \subset R^N \rightarrow R^M$, onde N é o número de atributos e M o número de possibilidades para o valor de Y (HECHT-NIELSEN, 1992). A saída da rede $f(O) = Y'$, é representada por M valores reais que representam a probabilidade do objeto pertencer à cada classe, a rede utiliza então a saída Y' em comparação com a Y para aprender, ajustando os pesos de acordo com a diferença dos resultados. A partir de vários exemplos, os padrões se formam e a rede é utilizada para prever o rótulo de novos objetos. (BLACK, 2001)

Um dos modelos de redes neurais mais conhecidos, Perceptron foi o nome dado ao esquema matemático criado baseado no funcionamento do neurônio em 1958 (ROSENBLATT, 1958). Ele é formado por um somatório e uma função de ativação descrito na equação:

$$f(n) = \begin{cases} 1, & \text{se } \varphi(\sum_{j=1}^N w_j x_j) \geq \text{threshold} \\ 0, & \text{se } \varphi(\sum_{j=1}^N w_j x_j) < \text{threshold}, \end{cases} \quad (2)$$

onde N é o número de atributos da entrada, e threshold é o valor limite para diferenciar as classes e depende da função utilizada.

A Figura 1 mostra um esquema geral do funcionamento. Existem diversas entradas, cada uma delas tem um peso associado, onde cada valor é multiplicado pelo seu respectivo peso, todos resultados são somados com uma constante 1 multiplicada por um peso chamada bias que contribui na translação da função para melhor ajuste ao modelo. O somatório é aplicado a uma função $\varphi()$ que determina a saída do neurônio. A correção do modelo é feita comparando-se essa saída com o resultado esperado e utiliza-se o erro para ajustar os pesos e o bias.

A primeira coisa que percebe-se é que um neurônio sozinho é uma estrutura limitada, visto que ele corresponde a uma separação linear no espaço dos atributos. Para dividir o espaço de uma forma ideal normalmente vários neurônios são aninhados, o que possibilita o aprendizado de padrões mais complexos.

O problema é, com um neurônio, os pesos de entrada são atualizados utilizando a diferença da saída desejada para real. Quando se tem um neurônio intermediário, não se sabe qual seria sua saída desejada, visto que ele é utilizado de entrada para outro neurônio. A questão está em como direcionar a atualização dos pesos de forma que o

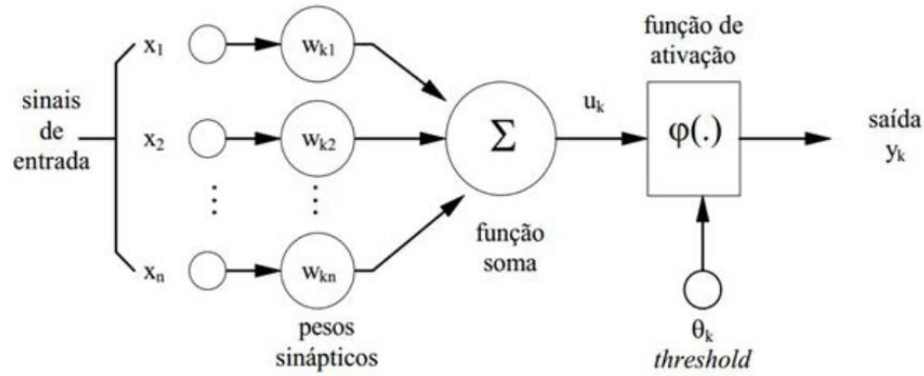


Figura 1 – Esquema de um neurônio - perceptron. Fonte: Retirada de (HAYKIN, 2021)

neurônio que não está na camada de saída também aprenda. Este foi um grande problema nas redes neurais e foi resolvido com a criação do método de treinamento chamado de *backpropagation* (STERN, 1992).

O *backpropagation* funciona utilizando o gradiente descendente a partir da derivada da função de ativação do neurônio em relação ao erro. Essa derivada é propagada no sentido inverso da rede, indo da camada final até a camada de entrada. Essa atualização com base no gradiente descendente faz com que a rede caminhe em direção ao erro mínimo (RUMELHART; HINTON; WILLIAMS, 1986).

Como o *backpropagation* utiliza a derivada da função de ativação do neurônio, logo a função não pode ser linear, para ter uma derivada diferente de uma constante, sendo assim. Duas funções não lineares comuns são as sigmoide e tangente hiperbólica.

A função sigmoide é dada pela equação 3. Sua derivada é dada pela equação 4. Ela assume valores entre 0 e 1, e é uma função saturante, o que significa que quando o valor do somatório é muito alto ou muito baixo, a variação de x altera pouco o resultado da função (HAN; MORAGA, 1995).

$$f(x) = (1 + e^{-x})^{-1} \quad (3)$$

$$f'(x) = f(x) * (1 - f(x)) \quad (4)$$

A tangente hiperbólica (\tanh) é uma variação da sigmoide, definida na equação 5, pode assumir valores entre -1 e 1 e sua derivada é dada pela equação 6. Assim como a sigmoide ela é saturante nos valores altos e baixos (SHARMA, 2017).

$$f(x) = (1 + e^{-x})^{-1} \quad (5)$$

$$f'(x) = 1 - f^2(x) \quad (6)$$

O Perceptron multi-camadas (MLP) foi criado utilizando vários *perceptrons* aninhados em paralelo e sequência, as camadas foram divididas em entrada, saída e escondida,

como mostrado na Figura 2. Os neurônios são organizados em sequência sem ciclos, ligados por conexões que contém pesos associados a ele, por onde os valores são propagados do neurônio anterior ao próximo. Esse modelo foi possível graças ao *backpropagation*, que permite que a rede seja atualizada para que os pesos se adequem aos dados de entrada e saídas desejadas (RUMELHART; HINTON; WILLIAMS, 1986). A MLP é considerada um dos métodos estado da arte para vetor de atributos (RUMELHART; HINTON; WILLIAMS, 1985).

A camada de entrada tem o tamanho dos atributos dos objetos. A camada de saída, chamada de softmax, o neurônio de saída com maior valor é definida como a classe daquela entrada. As camadas escondidas, podem ter um número indefinido de camadas, e quantos neurônios forem necessários em cada uma delas. Vale notar que quanto mais camada e mais neurônios tiverem, mais complexa será a separação, podendo ser necessários mais dados para fazer uma divisão que represente bem as classes (GARDNER; DORLING, 1998).

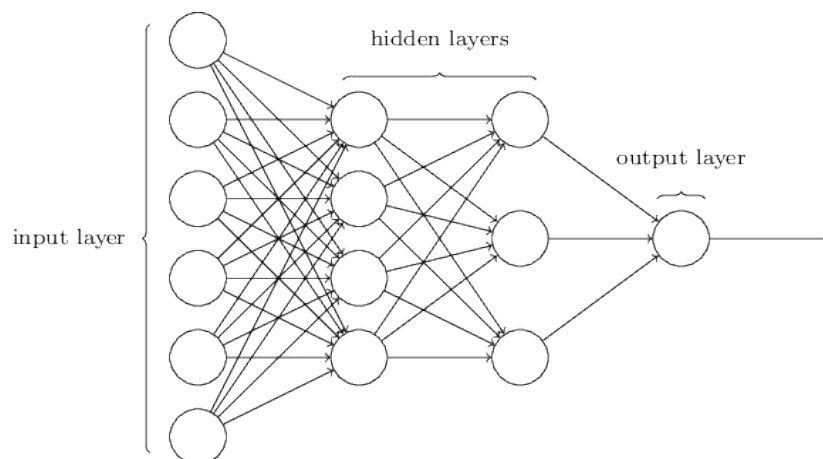


Figura 2 – Esquema de rede da Perceptron multicamadas Fonte: Retirada de (RODRÍGUEZ, 2020)

Depois da MLP outros modelos de rede foram desenvolvidos, cada um especializado em um tipo de entrada. O aprendizado profundo é uma categoria que contém modelos mais complexos (GOODFELLOW et al., 2016). Dentro do aprendizado profundo estão as redes recorrentes e as convolucionais.

A rede recorrente funciona bem para análise de texto e linguagens, onde existe uma sequência temporal entre os objetos e instâncias recentes dependem também das anteriores. A estrutura que garante esse comportamento são as conexões cíclicas formadas no modelo da rede, onde novos valores funcionam de entrada, enquanto os antigos continuam percorrendo as conexões. É considerada o estado da arte para problemas sequenciais (LIPTON; BERKOWITZ; ELKAN, 2015).

A rede convolucional é o estado da arte para classificação de imagens, e tem seu funcionamento baseado no córtex visual, o modelo aprende a partir de relações locais dos

objetos. A rede aplica filtros com uma espécie de janela em pedaços menores da imagem utilizando as relações locais entre os pontos (KRIZHEVSKY; SUTSKEVER; HINTON, 2012).

2.1.2 Aprendizado profundo

A partir da MLP outro conceito foi criado, o de aprendizado profundo (*deep learning*), que são redes com várias camadas ocultas que conseguem aprender padrões altamente complexos (GOODFELLOW et al., 2016), porém, com alto custo computacional e que normalmente necessitam de maiores quantidades de dados para resultados satisfatórios. Dentro do aprendizado profundo, foram também incluídos outros tipos de arquitetura de rede, dentre eles, as redes convolucionais, criadas baseando-se no córtex visual (LECUN; BENGIO; HINTON, 2015), as redes recorrentes que contém um “vetor de estado” e conseguem agregar ao processamento da entrada atual, os resultados dos objetos passados (LECUN; BENGIO; HINTON, 2015).

2.1.2.1 Redes *echo state*

Redes recorrentes são utilizadas quando existe uma representação sequencial nos dados e o resultado de um objeto, depende não só de seus atributos próprios mas também dos objetos anteriores (PEARLMUTTER, 1995). A forma encontrada para lidar com a recorrência nas Redes Neurais Recorrentes (do inglês, Recurrent Neural Networks - RNNs), foi a inclusão de ciclos em sua estrutura de rede, fazendo com que estados passados contribuam no estado atual.

A rede recorrente é um tipo de rede neural artificial bem conhecida, onde as informações são propagadas através de neurônios interconector, mas diferente do famoso modelo *feed forward*, (ou *Perceptron* Multicamada), a rede recorrente apresenta ciclos em suas topologias de rede, fazendo com que seja um sistema dinâmico capaz de aplicar transformações não lineares nos dados de entrada (HOCHREITER; SCHMIDHUBER, 1997; LUKOŠEVIČIUS; JAEGER, 2009; LECUN; BENGIO; HINTON, 2015).

Aplicações relacionadas as redes neurais artificiais tem características relacionadas ao domínio sequencial e temporal, como reconhecimento de fala, análise de séries temporais, traduções automáticas (GRAVES; MOHAMED; HINTON, 2013; ZAREMBA; SUTSKEVER; VINYALS, 2014; LI; HAN; WANG, 2012).

Apesar do número considerável de trabalhos demonstrando os sucessos notáveis do RNN em varias aplicações, existem diversas outras pesquisas que apontem suas limitações relacionadas a convergência, escalabilidade e parametrização (BENGIO; SIMARD; FRASCONI, 1994; PASCANU; MIKOLOV; BENGIO, 2013). Motivado por tamanhas desvantagens, novos paradigmas foram propostos.

O gradiente descendente utilizado para atualizar os pesos, tende a diminuir ou crescer a cada ciclo de treinamento, o problema é que pela repetição das conexões em poucas iterações o mesmo explode e os pesos crescem para valores altíssimos ou desaparecem com valores minúsculos (BENGIO; SIMARD; FRASCONI, 1994) que não influenciam a rede, tornando a convergência não garantida e quando acontece, pode ir para mínimos locais ruins (LIPTON; BERKOWITZ; ELKAN, 2015).

Para resolver este problema, diferentes estruturas foram propostas. Uma das mais bem sucedidas foi a Long Short-Term Memory (LSTM) (HOCHREITER; SCHMIDHUBER, 1997), que adicionou a estrutura de unidade de portão e memória. A memória grava as entradas importantes que devem ser salvas. Além dela, existe uma unidade de entrada que protege a memória de entradas irrelevantes, uma de saída, que protege as outras próximas unidades de conteúdos irrelevantes salvos na memória, e um portão de esquecimento responsável por apagar informações não importantes da memória. Dessa forma, nem todas as ligações são ativadas sempre, evitando o problema do gradiente descendente (LIPTON; BERKOWITZ; ELKAN, 2015).

Outro modelo proposto chamado de computação de reservatório (LUKOŠEVIČIUS; JAEGER, 2009) mantém a estrutura da rede recorrente inicial, porém, torna os pesos fixos, eliminando o treinamento e utiliza um vetor de estados que contém informações dos objetos passados. Quando os dados passam pela rede, sofrem transformações espaciais que incorporam o aspecto temporal quando cada objeto que serve de entrada se mistura ao vetor.

Em relação ao LSTM, o *reservoir* tem a vantagem de não ter fase de treinamento, ideal para entradas e processamentos de base de dados maiores. A LSTM também é integrada ao classificador, tendo uma MLP conectada ao final dela, o problema disso é que a estrutura fica mais travada. Como no EEG existem diferentes eletrodos, a forma de lidar com eles podem variar, o que é facilitado pelas redes echo state que realizam uma transformação da entrada e precisam de outro método para o papel de classificação, entre a transformação e a classificação, os dados conseguem ser manipulados e agrupados.

Um desses paradigmas é chamado de computação em reservatório, e foi originalmente idealizado em dois trabalhos independentes quase ao mesmo tempo, e que apresentam características em comum, sendo eles a rede de estado echo (do inglês, Echo State Network - ESN) (JAEGER, 2001) e a rede de estado líquido (do inglês, Liquid State Network - LSN) uma operação de convolução com um (NATSCHLÄGER; MAASS; MARKRAM, 2002). O modelo LSN tende a representar e replicar o neurônio de modo mais sofisticado e biologicamente plausível, enquanto o ESN que estamos interessados, é mais focado em teorias de aprendizado de máquina (LUKOŠEVIČIUS; JAEGER, 2009).

O modelo ESN padrão, assegura-se que a propriedade algébrica conhecida como propriedade echo state, seja satisfeita pela estrutura do *reservoir* (JAEGER, 2001). A rede echo state é usualmente composta por dois componentes principais: a estrutura do *reser-*

voir e a camada readout.

O conjunto é responsável por mapear os dados de entrada para um espaço de alta-dimensional e em seguida aprender os padrões deste novo espaço. No que diz respeito a estrutura do *reservoir*, ela contém ciclos como a rede recorrente original, porém, sua diferença principal é que os pesos internos das redes são sempre fixos, ou seja, não é necessário treinamento ou esquema para atualização de pesos.

Por outro lado, a camada de readout é responsável por todo o aprendizado, e criar as associações necessárias entre os dados transformados e a saída desejada, na qual pode requerer um processo de treinamento. Diversos trabalhos na literatura mostram que um simples algoritmo linear na camada de readout é o bastante para atingir bons resultados para muitos problemas, enquanto outras pesquisas apontam suas desvantagens (JAEGER, 2005).

Diversos métodos e variantes do modelo original ESN (JAEGER, 2001) foram propostos na literatura, a maioria deles cobrindo mudanças nas propriedades e comportamentos dos neurônios, que resultam em diferentes tipos de transformações não lineares (INUBUSHI; YOSHIMURA, 2017; XIA; JAHANCHAH; MANDIC, 2014).

Existem também métodos ESN que trabalham com múltiplos *reservoir* no lugar de somente um (GALLICCHIO; MICHELI, 2017; MALIK; HUSSAIN; WU, 2016) e outros trabalhos que focam em melhorar a estrutura randômica do *reservoir* usando heurísticas (ISHU et al., 2004; FERREIRA; LUDERMIR, 2009; SERGIO; LUDERMIR, 2012; RABIN et al., 2013; CHOUIKHI et al., 2017) ou modelos matemáticos (YIN; MENG; JIN, 2012). Por outro lado, modelos baseados em redes complexas foram tópicos pouco explorados no contexto de do ESN, com uma contribuição relacionada publicada recentemente em (KAWAI; PARK; ASADA, 2019).

Reservoirs são aplicados para problemas temporais supervisionados. Suas funções de atualização são dadas por:

$$\tilde{x}(n) = \tanh(W^{in}[1; u(n)] + Wx(n-1)) \quad (7)$$

$$x(n) = (1 - \alpha)x(n-1) + \alpha\tilde{x}(n) \quad (8)$$

onde $x(n) \in \mathbb{R}^{N_x}$ é o vetor de ativação dos neurônios do *reservoir*, $\tilde{x}(n) \in \mathbb{R}^{N_x}$ é a atualização do vetor estado inteiro no tempo n , $\tanh(\cdot)$ é aplicado a cada elemento, $[\cdot, \cdot]$ significa uma concatenação vertical em vetor(ou matriz), $W^{in} \in \mathbb{R}^{N_x \times (1+N_u)}$ e $W \in \mathbb{R}^{N_x \times N_x}$ são as matrizes de pesos de entrada e de recorrência respectivamente, e $\alpha \in (0, 1]$ é a taxa de vazamento usada para definir quanto o $x(n)$ vai trazer de $x(n-1)$ e quanto vai de $\tilde{x}(n)$. Uma função sigmoide pode ser usada na saída ao invés da tanh. O modelo pode ser feito sem a taxa de vazamento, o que é um caso especial onde $\alpha = 1$ e então $x(n) \equiv \tilde{x}(n)$ (LUKOŠEVIČIUS, 2012).

A saída linear pode ser definida por

$$y(n) = W^{out}[1; u(n); x(n)], \quad (9)$$

onde $y(n) \in \mathbb{R}^{N_y}$ é a saída da rede, $W^{out} \in \mathbb{R}^{N_y \times (1+N_u+N_x)}$ é a matriz de pesos de saída e $[\cdot; \cdot; \cdot]$ novamente significa concatenação vertical de vetor (ou matriz) (LUKOŠEVIČIUS, 2012).

A criação de um *reservoir* utilizando o ESN segue alguns princípios:

- ❑ Grande (centenas ou milhares de neurônios)
- ❑ Esparsos (de 20% a 30% de ligações)
- ❑ Randômico (pesos gerados aleatoriamente utilizando alguma distribuição com média zero)

Além desses princípios, existe a propriedade *echo state*, que serve para que todo objeto já passado pelo *reservoir* desapareça a medida que o tempo passa ($t \rightarrow \infty$). Dada a matriz de pesos W , a propriedade é observada quando $\rho(W) < 1$, onde $\rho(W)$ é o maior autovalor do espaço W (JAEGER, 2007). Esse valor teoricamente por si só não garante a propriedade *echo state*, porém, vários estudos já demonstraram, na prática que essa restrição é o bastante para garanti-la (YILDIZ; JAEGER; KIEBEL, 2012). Existem casos em que $\rho \geq 1$ ainda mantém a propriedade e momentos que $\rho < 1$ não a respeita. Vale notar que quanto maior o valor de ρ , maior memória o *reservoir* terá, ou seja, objetos passados a mais tempo ainda serão lembrados, e quanto menor mais importante os objetos recentes se tornam (LUKOŠEVIČIUS; JAEGER, 2009).

2.1.2.2 Redes Neurais Convolucionais

Redes convolucionais foram originalmente desenvolvidas para análise de imagens, pois, como o córtex visual, respondem ao estímulo apenas em regiões dos campos de visão, conhecidos como campos receptivos (O'SHEA; NASH, 2015). A camada convolucional funciona aplicando-se uma operação de convolução com uma janela de tamanho $z \times z$ em pixels da imagem de mesmo tamanho, deslizando-o por toda imagem.

A Figura 3 mostra um exemplo de um filtro 5×5 aplicado no canto superior esquerdo. Neste filtro existem valores que são multiplicados pelo valor contido na imagem na coordenada equivalente, e o resultado dos produtos são somados e aplicados na função de ativação, e adicionados a matriz de saída. Depois a janela desliza para o lado, e faz o mesmo cálculo com o ponto vizinho como central e seu resultado é também adicionado à matriz. Este processo é feito em toda a imagem, de forma que no final exista uma matriz chamada de mapa de características (ALBAWI; MOHAMMED; AL-ZAWI, 2017). Após as camadas convolucionais existe a rede densamente conectada, que nada mais é do que uma MLP. (KRIZHEVSKY; SUTSKEVER; HINTON, 2012)

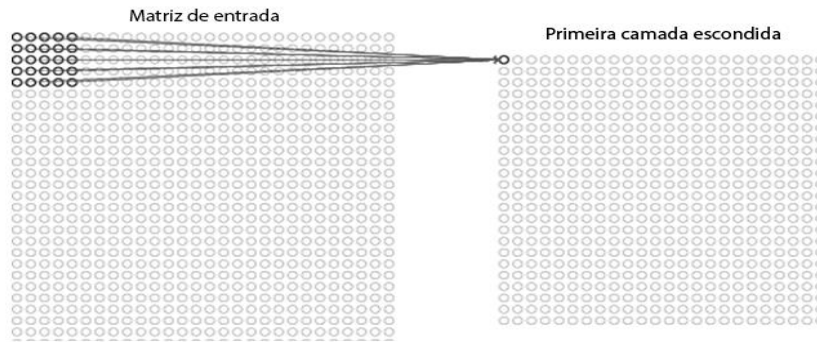


Figura 3 – Funcionamento do filtro na camada convolucional. Fonte: Retirada de (DESH-PANDE, 2020a)

A Figura 5 mostra um exemplo de um filtro 7×7 que representa como característica uma linha curva. A Figura 6 mostra o filtro sendo aplicado na parte selecionada da Figura 4 e resulta em um valor de 6600, o que demonstra que aquele pedaço da imagem contém a característica representada no filtro, enquanto a Figura 7 aplica em uma parte da imagem que não é caracterizada pelo filtro, então o valor resultante é 0, considerado baixo.

Uma mesma camada convolucional pode ter vários filtros de mesmo tamanho que são aninhados dando uma profundidade a camada. Várias camadas podem ser colocadas em sequência, de forma que as características aprendidas, sejam cada vez mais complexas de alto nível (KRIZHEVSKY; SUTSKEVER; HINTON, 2012; ALBAWI; MOHAMMED; AL-ZAWI, 2017; O'SHEA; NASH, 2015).

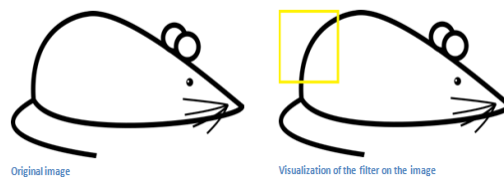


Figura 4 – Seleção de um pedaço da imagem para passagem do filtro. Fonte: Retirada de (DESHPANDE, 2020a)

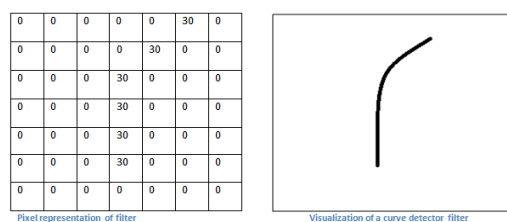


Figura 5 – Transformação da imagem em uma matriz numérica. Fonte: Retirada de (DESHPANDE, 2020a)

Stride representa quanto o filtro vai se deslocar entre uma aplicação do filtro e outra. A Figura 8 mostra um *stride* 1, logo o filtro será aplicado em todos os pontos da imagem,

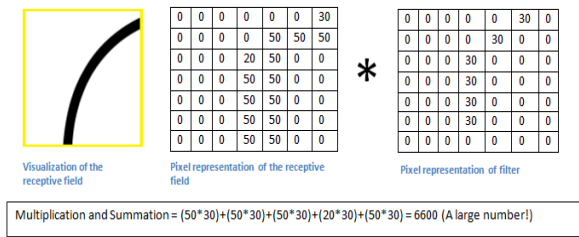


Figura 6 – Aplicação do filtro na matriz numérica que representa uma imagem semelhante a desejada pelo filtro. Fonte: Retirada de (DESHPANDE, 2020a)

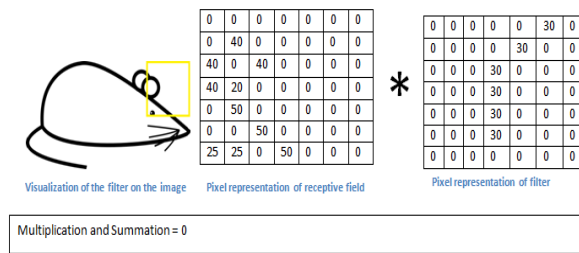


Figura 7 – Aplicação do filtro na matriz numérica que representa uma imagem diferente da desejada pelo filtro. Fonte: Retirada de (DESHPANDE, 2020a)

já a Figura 9 funciona com um *stride* 2, então valores vizinhos na matriz resultante vão ter um deslocamento de 2 valores da matriz de entrada (ALBAWI; MOHAMMED; AL-ZAWI, 2017).

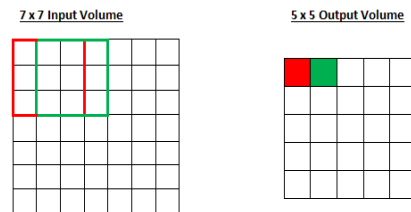


Figura 8 – Funcionamento do *stride* de tamanho 1, a seleção exibida na matriz da esquerda, se torna o ponto de mesma cor na matriz da direita. A distancia entre os filtros é o *stride*. Fonte: Retirada de (DESHPANDE, 2020b)

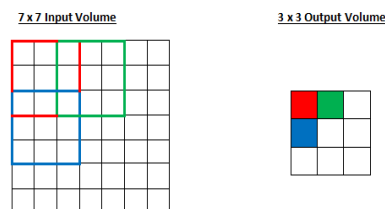


Figura 9 – Funcionamento do *stride* de tamanho 2, a seleção exibida na matriz da esquerda, se torna o ponto de mesma cor na matriz da direita. A distancia entre os filtros é o *stride*. Fonte: Retirada de (DESHPANDE, 2020b)

Ao analisar o funcionamento do filtro na imagem, é possível perceber que a matriz

resultante é menor que a matriz de entrada, porque os pontos extremos da imagem não conseguem ser pontos centrais da janela, pois, caso fossem o mesmo ultrapassaria o extremo da imagem. Quando se deseja que a matriz final tenha o mesmo tamanho da entrada, utiliza-se o *padding*, que são bordas com valor 0, para que todos os pontos na imagem possam ser pontos centrais do filtro em algum momento, considerando-se *stride* 1. A Figura 10 mostra como ficaria uma imagem 32×32 com filtro de tamanho 5, o *padding* necessário seria de tamanho 2 (ALBAWI; MOHAMMED; AL-ZAWI, 2017).

0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0								0	0
0	0								0	0
0	0								0	0
0	0								0	0
0	0								0	0
0	0								0	0
0	0								0	0
0	0								0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

Figura 10 – Demonstração do padding, para alcançar todos os pontos nos extremos, a matriz é rodeada de valores 0. Fonte: Retirada de (DESHPANDE, 2020b)

Os seguintes cálculos são empregados para se definir o tamanho do *padding* ideal e da matriz de saída dada os parâmetros da camada convolucional:

$Zeropadding = \frac{k-1}{2}$ $Saida = \frac{w-k+2p}{s} + 1$ onde k, w, p, s são o tamanho do filtro, do dado de entrada, do *padding* e do tamanho do *stride* respectivamente.

As formas mais comuns de modelar a saída f dos neurônios são as funções com entrada x como as equações 10 e 11. As Funções não lineares que saturam são mais lentas do que a não-saturante como a equação 12 em termo de tempo de treinamento. Segundo Nair e Hinton (NAIR; HINTON, 2010) estes neurônios não-saturantes são chamados de Rectified Linear Units (ReLUs).

$$f(x) = (1 + e^{-x})^{-1} \quad (10)$$

$$f(x) = \tanh(x) \quad (11)$$

$$f(x) = \max(0, x) \quad (12)$$

Redes convolucionais que utilizam as ReLUs treinam significativamente mais rápido que uma rede equivalente usando as funções saturantes. Em Krizhevsky (KRIZHEVSKY; SUTSKEVER; HINTON, 2012) foi feita uma comparação entre tanh e ReLU, analisando a taxa de erro em função do número de épocas de treino utilizando uma rede convolucional de quatro camadas. Testes mostraram que a ReLU é seis vezes mais rápida de treinar que a tanh. Essa função é um grande diferencial nas redes convolucionais, pois, seu treinamento é caro computacionalmente (SHARMA, 2017).

A camada de *Pooling* tem a função de evitar o overfitting, além de diminuir o tamanho da rede, fazendo com que seja mais rápido treiná-las. A camada *pooling* recebe-se um mapa de características da camada convolucional e as condensa de acordo com um tamanho $z \times z$ e um *stride* s . A Figura 11 mostra um exemplo de max-pooling de tamanho 2×2 e *stride* 2.

Na matriz resultante da camada convolucional, se criar uma matriz 2×2 , e dos valores internos, somente o maior deles é passado para frente (o maior por ser max-pooling, avg-pooling, por exemplo pega a média dos valores) (ALBAWI; MOHAMMED; AL-ZAWI, 2017). O *stride* 2, significa que a cada ponto analisado na matriz, o próximo será 2 casas de distância deste, diminuindo assim, o tamanho do mapa por 2.

Note que quanto maior o *stride*, menor o tamanho do mapa de saída. Como a camada convolucional pode ter vários mapas de características, então o *pooling* é executado para cada um deles, o que, na prática significa que a saída da camada de *pooling* vai ter a mesma profundidade da saída da camada convolucional (RAWAT; WANG, 2017).

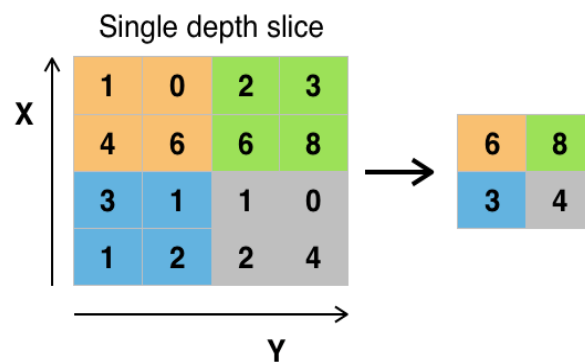


Figura 11 – Camada *Pooling*, a imagem mostra um max *pooling*, na matriz da esquerda, de cada 4 pontos de mesma cor, somente o maior é utilizado, resultando na matriz da direita. Fonte: Retirada de (DESHPANDE, 2020b)

Os modelos convolucionais mais conhecidos surgiram da competição de classificação de imagem ILSVRC, onde diferentes entidades competem na tarefa de classificação de imagens em uma base de dados gigante chamada ImageNet, com o intuito de acertar a classificação das imagens de validação utilizando a parte de treino. Os modelos são:

- ❑ **AlexNet:** A primeira rede convolucional a ganhar um grande destaque foi chamada de AlexNet. Com oito camadas, cinco convolucionais e três densamente conectadas. Utilizada na competição no ano de 2012, conseguiu diminuir de 26% para 16% a taxa de erro em relação ao melhor resultado no ano anterior (KRIZHEVSKY; SUTSKEVER; HINTON, 2012).
- ❑ **GoogLeNet:** A empresa google surgiu como a campeã do ano de 2014. A rede contém 13 camadas, formadas por duas camadas convolucionais, nove de uma nova estrutura chamada inception, é como estrutura que pode ser vista como uma rede

convolucional interna com suas próprias conexões e camadas, e por fim, duas camadas densamente conectadas (SZEGEDY et al., 2015).

Apesar de ter erro menor que os métodos posteriores, a versão que atingiu o melhor erro, foi executado 7 vezes em vez de somente uma, e o resultado foi decidido a partir de todas as execuções, gerando um custo computacional alto. Quando executado somente uma vez atingiu 7.89% de erro que é o considerado para comparação.

- **VGGNet:** A VGGNet (SIMONYAN; ZISSERMAN, 2014) foi a rede que serviu de base para a ResNet. A VGGNet ficou em segundo lugar na ILSVRC de 2014 e é formada por 16 camadas convolucionais, 3 camadas densamente conectadas, e 5 *pooling* entre elas.
- **ResNet Plain:** A ResNet Plain (HE et al., 2016) retirou da VGGNet as *poolings* e adicionou mais camadas convolucionais ficando com 33 ao todo. Das camadas densamente conectadas restou somente uma das três. O acréscimo de camadas a deixou com um erro maior que a VGGNet.
- **ResNet Residual:** Modelo vencedor no ano de 2016, a ResNet Residual (HE et al., 2016) resolveu o problema adicionando ao modelo da ResNet Plain uma nova estrutura de conexão entre camadas, a conexão residual. Ela conecta camadas distantes, onde a cada N conexões comuns entre duas camadas convolucionais (a saída de uma camada convolucional é a entrada da próxima camada), acontece a conexão residual, que traz o resultado da camada a N passos atrás e concatenada com a saída da camada convolucional anterior. Na competição a ResNet usou N igual a 2.

2.2 Análise de Sinais de Eletroencefalograma (EEG)

Com o avanço das tecnologias aplicadas na área da saúde, diversos tipos de exames médicos por imagens foram desenvolvidos e utilizados amplamente no diagnóstico de patologias neurológicas, como, por exemplo tomografia computadorizada e ressonância magnética (JUNG; PARK; HWANG, 2017; URSPRUNG et al., 2020; FU et al., 2019). Contudo, no contexto do Sistema Único de Saúde a utilização desse tipo de tecnologia em grande escala ainda não é a realidade (KAWANO, 2013). Sinais medidos a partir de um amplificador e alguns eletrodos, além de mais baratos, são mais acessíveis principalmente em ambientes de Unidades de Terapia Intensiva (UTIs). O eletroencefalograma (EEG) mede a atividade elétrica do córtex cerebral em diversas regiões do cérebro, utilizando para tal eletrodos dispostos no couro cabeludo abordando das regiões centrais até regiões occipitais (FREEMAN; QUIROGA, 2013; HENRY, 2006). Um exemplo do traçado EEG é mostrado na Figura 12, cada linha representa os sinais de um eletrodo ao longo do tempo.

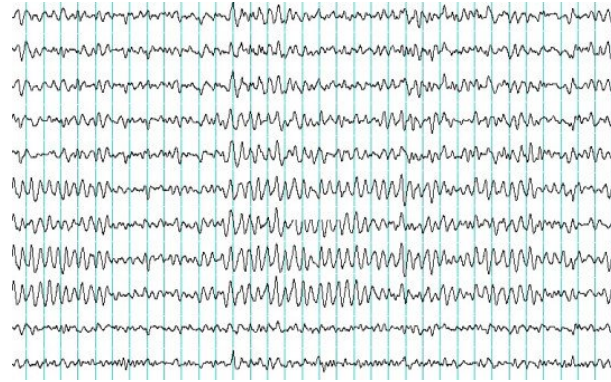


Figura 12 – Exemplo do traçado EEG Fonte: Retirada de (MAESTROVIRTUALE, 2021)

Esse sinal neurológico é analisado tanto a partir de seu comportamento quantitativo na análise temporal ou ainda no domínio da frequência (SÖRNMO; LAGUNA, 2005) quanto no morfológico, tendo alguns padrões de normalidade e anormalidade amplamente difundidos na neurologia diagnóstica (NIEDERMEYER; SILVA, 2005). O EEG tem como principal objetivo a monitorização da atividade cortical, seja de pacientes que se encontram em supervisão médica em UTIs, seja em tratamentos ambulatoriais de rotina (NUNEZ; SRINIVASAN et al., 2006). Uma das características mais relevantes do traçado EEG é sua irregularidade com relação à morfologia das ondas mensuradas (RUBIN; DAUBE, 2009), em consequência de sua não-estacionariedade, devendo inclusive ser analisado a partir de ferramentas não lineares (BABILONI et al., 2020). Uma das desvantagens da utilização desse sinal é elevada presença de artefatos (VIGÁRIO, 1997), que são todos aqueles sinais não originários do córtex cerebral, além da baixa resolução espacial (LAKSHMI; PRASAD; PRAKASH, 2014; BENBADIS, 2015).

2.2.1 Definições e conceitos

Os pioneiros na utilização da eletroencefalografia para registrar as correntes cerebrais foram Richard Caton e Hans Berger, aplicando em animais e humanos respectivamente (HAAS, 2003). Na década de 50, durante cirurgias cerebrais, Wilder Penfield e Hebert Jasper realizaram estimulações elétricas feitas de modo a localizar diferentes processos cerebrais em diferentes áreas do córtex (GUYTON, 2006).

Em 1958 foi elaborado um padrão de colocação de eletrodos chamado sistema 10-20 (HOMAN; HERMAN; PURDY, 1987) o qual posteriormente foi adotado como referência para captação do traçado EEG. O sistema 10-20 mostrado na Figura 13, fornece a posição específica em que os eletrodos devem ser posicionados, respeitando o espaçamento de 10 a 20% de distância entre as marcações (HERWIG; SATRAPI; SCHÖNFELDT-LECUONA, 2003).

Os eletrodos são representados por letras e números. As letras fazem referência ao osso em que o eletrodo deverá ser posicionado, podendo ser pré-frontal (Fp), frontal (F),

temporal (T), parietal (P), occipital (O), e central (C). Já os números pares referem-se ao hemisfério direito, os ímpares ao hemisfério esquerdo e aqueles detonados pela letra Z aos eletrodos localizados sobre a linha de separação dos hemisférios (EBERSOLE; PEDLEY, 2003).

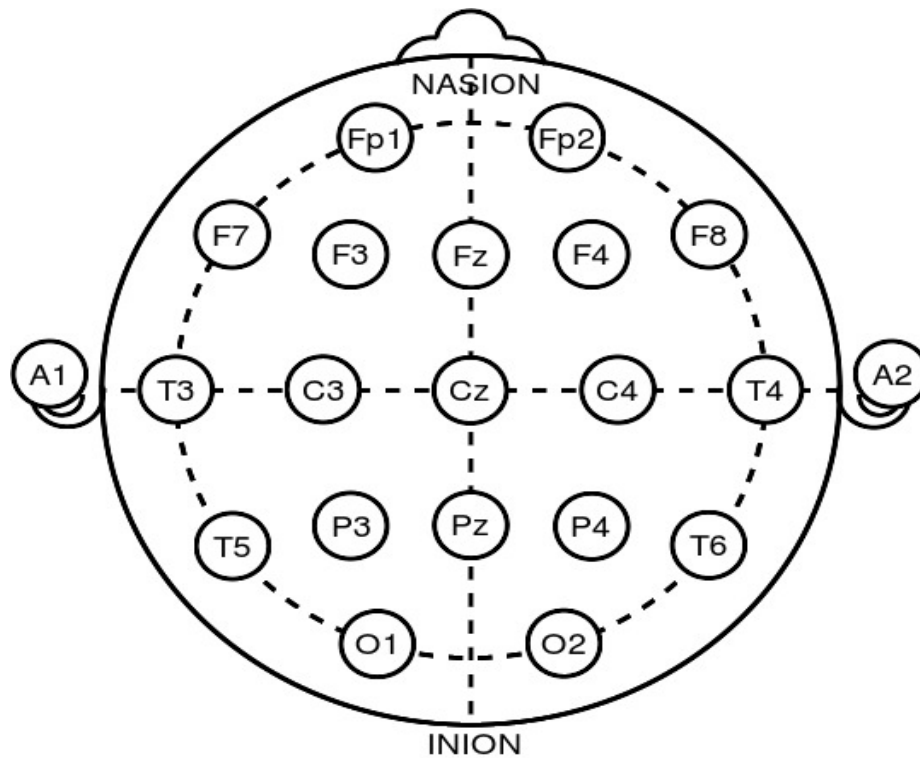


Figura 13 – Localização de eletrodos segundo o sistema 10-20. Fonte: Retirada de Retirada de (21..., 2021).

As oscilações do traçado EEG são bastante conhecidas na literatura e as principais ondas cerebrais são chamadas de ritmos, cujas faixas de frequências variam segundo autores. Esses ritmos cerebrais auxiliam os neurologistas no diagnóstico de pacientes com possíveis patologias neurológicas. A Tabela 1 nomeia e caracteriza as principais ondas cerebrais (RAMOS, 2017).

Estudos recentes avaliam ainda ritmos associados às frequências maiores, como ritmo Gama (30 a 80 Hz) e ritmo Supergama (frequências maiores que 80 Hz). Esses ritmos são relacionados às tarefas cognitivas que envolvem percepção, atenção e memória (KUCWICZ et al., 2017).

2.2.2 Abordagens convencionais de análise

Da literatura clássica sobre EEG (POSNER et al., 2007; NUNEZ; SRINIVASAN et al., 2006; SÖRNMO; LAGUNA, 2005; NIEDERMEYER; SILVA, 2005) sabe-se que esse exame é bastante utilizado para diagnóstico e acompanhamento de diversas patologias neurológicas como Epilepsia ou Parkinson, por exemplo. Por se tratar de um exame

Tabela 1 – Nomes e casos das diferentes frequências capturadas pelo exame EEG.

Delta	Observadas na faixa de 0 a 4 Hz, estão relacionadas ao sono profundo e bastante presentes em pacientes com distúrbios de consciência (RAMOS, 2017).
Teta	São indicadores de patologia neurológica quando presentes em adultos neurologicamente saudáveis, esse ritmo é observado nas frequências de 4 a 7 Hz (NI-EDERMEYER; SILVA, 2005).
Alfa	Observadas em frequências de 7 e 13 HZ indicam um estado de baixa ansiedade e estado de relaxamento. São comuns em adultos saudáveis em estado de vigília, ou seja, com a pessoa acordada. (FREEMAN; QUIROGA, 2013)
Beta	Comportando as faixas de frequência entre 13 a 30 Hz, são observadas com o paciente acordado e alerta ou concentrado.

simples, é muito utilizado no acompanhamento de pacientes neurologicamente críticos, como em estado de coma.

Em Amzica (2009) viu-se que estados de coma causado por etiologias como a hipóxia (baixo teor de O₂) ou ainda intoxicação por drogas, levam a um fenômeno denominado "surto-supressão", que pode ser claramente visualizado no EEG, apresentando um período relacionado as ondas lentas e de alta amplitude (*burst*), que está associado à fase de despolarização sináptica em um potencial de ação nos neurônios corticais, e um período relacionado a um plano contínuo, quase sem atividade elétrica (*suppression*), que acontece devido à ausência da atividade sináptica nos neurônios corticais.

Em Gollwitzer et al. (2015) constatou-se que a partir do sinal EEG é possível prever a isquemia cerebral, e em (LEMAIRE et al., 2014; RICARDO et al., 2012) notou-se que a redução da consciência devido às lesões cerebrais é visível em sinais EEG.

O exame já é amplamente utilizado para diagnosticar diferentes doenças como tumores cerebrais (SHARANREDDY; KULKARNI, 2013), doenças metabólicas (SAUNDERS, 1968), epilepsias (SHOEB; GUTTAG, 2010), controle de sedação (DEYNE et al., 1998), encefalites (SCHMITT et al., 2012), Parkinson (VANEGAS et al., 2018), Alzheimer (TRAMBAIOLLI et al., 2011), esquizofrenia (SABETI et al., 2011), epilepsia (SHOEB; GUTTAG, 2010), etc.

2.2.3 Prognóstico de pacientes em coma

Antes de definir o que é coma, é necessário definir o que é consciência. Consciência pode ser considerada como o perfeito conhecimento de si próprio e do ambiente, dotado da percepção de estímulos externos e internos (POSNER et al., 2007). Nível de consciência

tenta definir quão consciente uma pessoa é/está. Distúrbios de consciência diz respeito a problemas que podem diminuir severamente o nível de consciência de alguém (EEG... , 2021). Por fim, o coma pode ser definido como o nível mínimo ou mesmo a falta de consciência (WHAT... , 2021), podendo ser o resultado de lesões diretas ou indiretas da formação reticular ascendente, ou de lesões extensas em ambos os hemisférios cerebrais, ou mesmo de lesões diencefálicas bilaterais (HAUPT et al., 2015; TRAUB; WIJDICKS, 2016).

Em (SONG et al., 2020), uma revisão sistemática foi elaborada de modo a avaliar modelos prognósticos aplicados para identificar distúrbios de consciência. Esses modelos exploram dados demográficos dos pacientes, diagnóstico clínico, neurofisiologia, imagens cerebrais e ainda sinais EEG. Outra revisão sistemática que aborda aprendizado de máquina para prever resultados pós lesão cerebral foi a elaborada por (NOOR; IBRAHIM, 2020). Por meio dessa revisão, o sinal EEG em conjunto com outras informações, como nível de consciência, foram abordados em estudos anteriores para o desenvolvimento de modelos de prognóstico.

Os trabalhos relacionados ao coma não costumam utilizar o exame diretamente. De fato, a maioria dos trabalhos tenta resolver o problema com base no exame clínico de avaliação comportamental por meio de escalas como a Coma Recovery Scale Revised (CRS-R) e as escala de resultados de Glasgow (GOS) (ZIER et al., 2017), que são inevitavelmente subjetivas e vulneráveis a uma variedade de interferências (SONG et al., 2018).

Alguns trabalhos relacionados consideraram a análise de dados de equipamentos tecnológicos como ressonância magnética funcional (fMRI) (SONG et al., 2018) ou mesmo fMRI com o EEG (GOLKOWSKI et al., 2017), mas a maioria deles é estritamente com base na análise estatística padrão, ou seja, não há processo de aprendizado de máquina. Algumas exceções incluem os trabalhos (PUGIN et al., 2018), que adotou Support Vector Machine (SVM) e em (SONG et al., 2018) que utilizou regressão logística e Support Vector Machine em dados coletados utilizando ressonância magnética funcional (fMRI).

2.2.4 Aprendizado de sinais EEG de pacientes em coma

Geralmente os trabalhos que utilizam o EEG com relação ao coma extraem artefatos e atributos do exame para utilizar na classificação, e não utilizam o próprio exame. Qualquer potencial elétrico vindo de outra fonte que não o cérebro é definido como artefato, enquanto atributos são valores obtidos através dos sinais elétricos do exame e tem como fonte o cérebro (ANGHINAH et al., 2006).

O trabalho (BAGNATO et al., 2010) utiliza o EEG para medir a melhora do grau de consciência em pacientes que tiveram severo distúrbio de consciência seguido de coma, comparando 2 grupos diferentes, os traumáticos (que tinham ferimento na cabeça) e não traumáticos, demonstrando a utilização do EEG para análise de pacientes em coma.

Já a pesquisa (SCHORR et al., 2016), tenta classificar um grau para recuperação da síndrome da vigília sem resposta. Os pacientes foram observados por 12 meses, e a base de dados contém o EEG como objeto e um status de melhora ou piora após a observação como classe.

Ruijter (RUIJTER et al., 2018) utiliza os artefatos Musculares que são obtidos com a média da faixa de alta frequência (25-40Hz) do exame dividida pela média da baixa frequência (4-12Hz). Os picos de alta amplitude (Absoluto) que capturam a maior amplitude em um intervalo de 5 minutos. Os picos de alta amplitude (Relativos) que é a proporção máxima entre a amplitude e a média das amplitudes dos outros canais. E os canais Planos que são segmentos de 1 segundo do exame e tem desvio padrão menor que $0.1 \mu V$

Já Tolonen (TOLONEN et al., 2018) captura uma série de parâmetros que podem ser divididos em 5 categorias. Força Absoluta: Força delta, força theta, força alfa, força beta, força theta lenta, força theta rápida, etc; Força relativa: Força delta relativa, força theta relativa, força alfa relativa, força beta relativa, força theta lenta relativa, força theta rápida relativa, etc; Assimetria: Assimetria delta, assimetria theta, ..., índice de simetria cerebral revisado(r-BSI); Variabilidade: Variação de das diferentes faixas do exame, sendo absolutas e relativas; Outras: Média da frequência dominante, frequência mediana, frequência da borda espectral em 95%, etc;

Ramos (RAMOS, 2017) faz uma análise comparando os diferentes eletrodos, utilizando quantificadores como:

- Porcentagem de contribuição de potência (PCP): para uma determinada faixa de frequência, este quantificador informa a quantidade de energia acumulada. O mesmo é calculado sob as diferentes faixas de frequência do exame: Delta(1-3,5Hz), Teta(3,5-7,5Hz), Alfa(7,5-12,5Hz), Beta (12,5-30Hz;30-58Hz;62-80Hz), Gama(80-100Hz) e Supergama (58-62Hz).
- Variação da Potência Cerebral (VPC): é definido como o desvio padrão dos valores do PCP para cada ritmo.
- Frequência mediana (FM): meio do caminho entre as potências das frequências. Ponto onde metade da potência calculada estão nas frequências mais baixas e a outra metade nas frequências mais altas.
- Coerência: calculada pela magnitude quadrada de densidade espectral entre dois sinais, pode indicar a semelhança entre a base de dois sinais, valores próximos a 0 podem indicar baixa correlação enquanto próximos a 1 indicam alta correlação (SÖRNMO; LAGUNA, 2005).

Os impulsos gerados têm padrões não lineares e comportamentos altamente complexos (SUBHA et al., 2010), então utilizar artefatos e atributos diminuem a dimensão facilitando o processo de treinamento, uma vez que existem diversos algoritmos estado-da-arte para

vetor de atributos, sem ter que lidar com os problemas relacionados as séries temporais (SARAVANAN; SUJATHA, 2018).

Simplex extrações podem não retirar as informações necessárias para modelar o problema, já foi mostrado que modificar a representação das séries temporais podem levar ao declínio da função de erro (WANG; OATES, 2015), melhorando os resultados. Modelos de aprendizado profundo conseguem se adaptar a várias dessas representações.

Até onde sabemos, este projeto representa uma das primeiras tentativas de prever o resultado de pacientes em coma utilizando sinais do EEG considerando técnicas de aprendizado profundo.

2.2.5 Aprendizado profundo em sinais EEG

Em(CRAIK; HE; CONTRERAS-VIDAL, 2019), são discutidas diferentes técnicas de aprendizado profundo em uma ampla gama de tarefas de classificação de EEG, como reconhecimento de emoções (HUANG; XU; ZHANG, 2017; JIRAYUCHAROENSAK; PANGUM; ISRASENA, 2014), carga mental do paciente (BASHIVAN et al., 2015; LI et al., 2016), como motor de imagem (gerar imagens a partir das ondas)(STURM et al., 2016; LU et al., 2016), potencial de um evento (PEREIRA et al., 2018; SHAMWELL et al., 2016), detecção de convulsão (VIDYARATNE et al., 2016; ANTONIADES et al., 2017) e marcador de estágio de sono (ANSARI et al., 2018; SUPRATAK et al., 2017). O trabalho (CRAIK; HE; CONTRERAS-VIDAL, 2019) também mostra que de 54 artigos analisados, 20% trabalha com imagens, 41% calcula atributos a partir das ondas e 39% utiliza os próprios sinais do EEG.

Um trabalho utilizou o EEG para classificar emoções humanas (LI et al., 2017), criando uma imagem a partir dos valores dos eletrodos, montando uma espécie de imagem térmica baseada nas posições dos eletrodos na cabeça. Depois as imagens foram utilizadas numa rede recorrente, explicada no capítulo 2.1.2.1 para classificação.

Outro trabalho fez a classificação da carga mental dos indivíduos (JIAO et al., 2018). Ele também propôs uma imagem mental montada a partir do EEG, porém, separada por frequência de acordo com as faixas. Também propõe uma nova estrutura entre as redes convolucionais e as densamente conectadas.

A seguir serão apresentados pesquisas que serviram de motivação para o desenvolvimento de uma representação espacial do EEG bem como para os modelos de CNN investigados.

2.2.5.1 Representação espacial do EEG usando atributos multi-dimensionais

O trabalho de Li et al. (2017) visa reconhecer emoções humanas utilizando sinais de eletroencefalograma (EEG). Emoção é um importante símbolo da inteligência humana,

então a capacidade das máquinas reconhecerem emoções humanas também é naturalmente muito significativo para a inteligência artificial. No trabalho dois aspectos são observados:

O primeiro aspecto é a extração de características do EEG: geralmente quando se trata do EEG usa-se somente em dimensões temporais e de frequência e raramente na dimensão espacial. O segundo aspecto é como construir um classificador que consiga capturar essas mudanças e aprender com elas a categorizar as diferentes emoções.

Com o EEG usando o sistema 10-20, a ideia é utilizar a organização espacial vinda da coleta para a representação dos dados. Assim, a Figura 14 mostra que cada eletrodo terá uma posição específica na matriz. Como pode ser visto na Figura 15 o exame passa a ser representado por uma matriz de 3 dimensões, de tamanho $m \times m \times N$, sendo N o tamanho do exame e m a largura e altura de um mapa construído a partir de cada instante do EEG e simula o posicionamento dos eletrodos. Além da organização, o trabalho utiliza uma função para preencher os espaços vazios do mapa e aplica uma função de interpolação para aumentar a dimensão da matriz, a Figura 15 mostra a representação desta etapa.

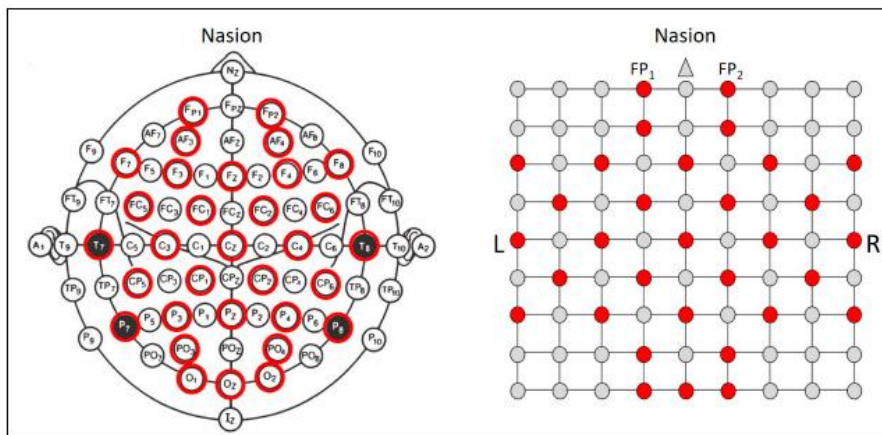
Primeiro cada ponto cinza na Figura 14 é preenchido seguindo a formula:

$$V_{m,n} = \frac{V'_{m+1,n} + V'_{m-1,n} + V'_{m,n+1} + V'_{m,n-1}}{K}, \quad (0 \leq m, n \leq 8, m, n \in N), \quad (13)$$

onde, $V_{m,n}$ é o valor na areá cinza nas coordenadas (m, n) , V' são os vizinhos de V , e K é o número de vizinhos diferente de zero.

Depois uma função de interpolação é utilizada transformando o tamanho da imagem para 200x200. A interpolação também é executada no quesito temporal, e com diferentes tempos entre as janelas utilizadas.

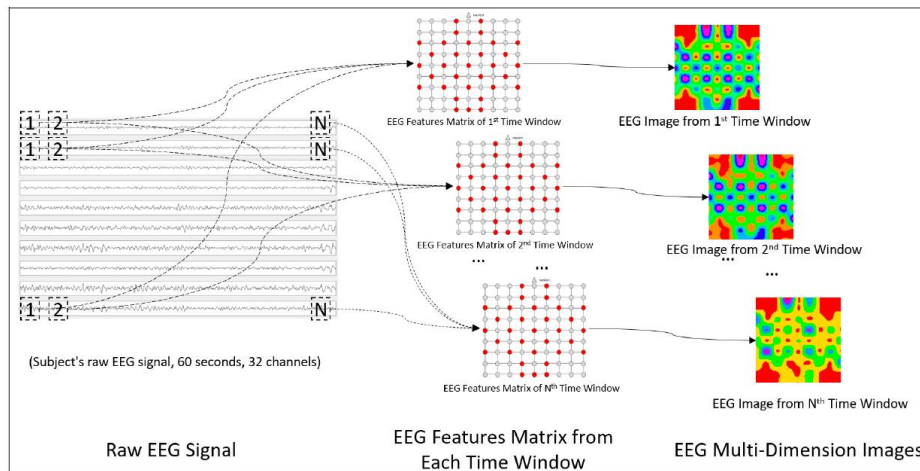
Figura 14 – Estrutura espacial para modelagem da matriz a partir do sistema de organização 10-20.



Fonte: Li, Youjun and Huang, Jiajin and Zhou, Haiyan and Zhong, Ning (LI et al., 2017)

O modelo escolhido para a classificação em Li et al. (2017) foi a rede neural artificial, porém, utilizando nas camadas finais LSTM no lugar da camada densamente conectada.

Figura 15 – Passos desde o captura do exame até a geração da imagem final utilizada pela rede.



Fonte: Li, Youjun and Huang, Jiajin and Zhou, Haiyan and Zhong, Ning (LI et al., 2017)

Para criar os rótulos, foi utilizado um dataset que incluem valores de medidas emocionais (Valência, Dominação, Excitação, Gosto e Familiaridade), onde duas foram utilizadas: valência e excitação. Um plano bidimensional é formado, onde em uma dimensão tem o nível de valência e em outra o nível de excitação. O plano é dividido em 4 quadrantes cada objeto recebe como rótulo o quadrante que caiu.

O método foi comparado com diferentes técnicas de classificação e analisado usando 13 diferentes tamanhos de janelas. O método proposto é melhor em 8/13 janelas. O dois algoritmos que obtém os melhores resultados fazem com a janela de tamanho "2 segundos". O melhor resultado (CLRNN) supera o segundo melhor (SVM) em 9.02%.

2.2.5.2 Representação multi-escala para classificação de séries temporais

Existem alguns fatores chaves quando se tem a intenção de classificar séries temporais. Diferentes séries temporais podem precisar de representações em diferentes escalas de tempos, e os métodos existentes raramente se adaptam às escalas mais adequadas. Além disso, séries temporais são frequentemente distorcidas por ruídos aleatórios e perturbações de altas frequências. É necessário um processo de tratamento de ruído e suavização automática para limpar a série.

O framework proposto em Cui, Chen e Chen (2016) possui três estágios sequenciais: transformação, rede convolucional local e rede convolucional completa. Na **transformação**, a série temporal sofre várias mudanças, tais como redução da dimensão dos dados, transformação na frequência e na própria série original. Na **rede convolucional local** é utilizada diversas camadas convolucionais (com *max pooling*) para extrair características de cada entrada/transformação. Na **rede convolucional completa**, todas as saídas das redes regionais são agrupadas, e então são aplicadas outras redes convolucionais (também

seguidas por camadas *max pooling*), então uma rede totalmente conectada e por final uma camada softmax que dita o resultado.

Em relação às transformações, destacam-se dois tipos: multi-escala e multi-frequência. Na transformação multi-escala, diferentes séries temporais são consideradas a partir de diferentes escalas de tempo. Supondo uma série $T = \{t_1, t_2, \dots, t_n\}$ e uma taxa de redução k , então da rede só é mantido o k -ésimo ponto da série, formando uma nova série T^k . Usando este método com vários valores de k múltiplas novas séries são geradas.

$$T^k = t_1 + k * i, i = 0, 1, 2, \dots, \lfloor \frac{n-1}{k} \rfloor. \quad (14)$$

Nas transformações multi-frequência são adotados múltiplos graus de suavização gerando várias sequências a partir da série original. Com filtro de baixa frequência, é reduzida a variação da série. Supondo a série original $T = \{t_1, t_2, \dots, t_n\}$, a nova série temporal após a aplicação de um filtro de média é dada por:

$$T^i = \frac{t_i + t_{i+1} + \dots + t_{i+l-1}}{l}, i = 0, 1, 2, \dots, n-l+1, \quad (15)$$

onde l é o tamanho da janela. Assim, cada ponto T^i vai ser substituído pelo intervalo de pontos que vai do índice i até $i+l$.

Uma transformada de furrier é utilizada na matriz final, para realizar uma interpolação, transformando em uma matriz final de tamanho 254x254.

A imagem formada é utilizada na rede convolucional para classificação, além disso, é utilizada no lugar da cada densamente conecta uma rede LSTM, uma rede recorrente.

Resultados

Os resultados demonstram uma melhora na acurácia de quase de 10% em relação ao segundo melhor método, o SVM. Diferentes pedaços de janelas tem diferentes resultados, os menores pedaços são os melhores. O maior resultado é para o pedaço de 2 segundos, uma acurácia de 75.21%, contra 65.11% do SVM.

Aprendizado Tradicional para o Prognóstico de Pacientes em Coma (PPC)

Neste capítulo são apresentados os dados considerados no estudo. O ambiente experimental e as medidas de desempenho adotadas também são descritos. Por fim, o capítulo também traz os resultados obtidos através de uma abordagem benchmark baseada em quantificadores, os quais servirão de referência para análise dos resultados obtidos nos capítulos seguintes.

3.1 Descrição da Base de Dados

Esse é um estudo retrospectivo, pois analisa dados de pacientes comatosos, internados na UTI adulta do HCU-UFU entre os anos de 2010 e 2013, e os anos de 2018 e 2019. Foram utilizadas, portanto duas bases de dados, produzidas sob orientação do professor da Faculdade de Engenharia Elétrica da UFU, João Batista Destro Filho.

A primeira base de dados utilizada, composta por 74 registros EEG, foi produzida em (PAIVA et al., 2018) e os dados avaliados no presente estudo foram selecionados previamente em (RAMOS, 2017), totalizando 74 registros EEG. A coleta e utilização desses dados foram autorizados pelo Comitê de Ética em Pesquisa (CEP) da UFU com protocolo nº369/11 de 02 de fevereiro de 2012.

A segunda base de dados utilizada, composta por 46 registros EEG, foi feita uma aluna de mestrado do professor João Batista Destro Filho, Patrícia Costa, cujas coletas tiveram início em 2018 (parecer de liberação emitido pelo CEP nº 2.570.022).

Os registros de EEG em pacientes comatosos no setor de UTI do HCU-UFU foram realizados utilizando o amplificador de sinais biomédicos disponível no SEM/HCU-UFU, seguindo o padrão 10-20 de colocação dos eletrodos, com banda de passagem de 1 Hz e de 40 Hz, para a primeira base de dados, e de 1 Hz a 100 Hz para a segunda base de dados.

Os valores das frequências de amostragens variaram entre 200, 300 e 400Hz. Os médicos neurologistas do HCU-UFU, Marcos Campos e Paulo Beterro foram responsáveis por avaliar os dados coletados, bem como escolher trechos (épocas) para análises quantitativas.

Dos trechos selecionados, cada paciente teve 9 trechos selecionados, cada um foi observado e avaliado como um exame independente, o resultado final para cada paciente foi definido pela moda de todos os resultados. Posteriormente uma heurística foi proposta para escolher o resultado final com base no conjunto de resultados destes trechos.

O total de 120 registros de EEG foram considerados para análise. Nesse cenário, a quantidade de registros EEG masculinos foi de 86. A idade média dos pacientes analisados foi de $51,9 \pm 18,2$ anos de idade. Já as etiologias dos comas foram diversas, sendo que as principais foram: 26 pacientes com Acidente Vascular Encefálico (AVE); 41 pacientes com Trauma Crânio Encefálico (TCE); 33 pacientes com encefalopatia pós anóxica e/ou coma metabólico. Os demais pacientes apresentaram diversas outras causas como etiologia do coma, por exemplo, tumor, ferimento por arma de fogo, entre outras.

Cada um desses registros EEG avaliados apresentou um possível desfecho. No presente estudo foram considerados três desfechos a saber:

- ❑ Favorável: Paciente recebeu alta médica da UTI adulta, sob condições favoráveis de vida;
- ❑ Desfavorável, Óbito Clínico: Paciente veio a óbito enquanto estava internado na UTI ou pouco tempo após ser transferido de setor, por causas clínicas variadas, como por exemplo, parada cardiorrespiratória;
- ❑ Desfavorável, Óbito por Morte Encefálica: Paciente apresentou morte cerebral enquanto internado na UTI, tendo todos os procedimentos para detecção de morte encefálica sido realizados.

Na Tabela 2, então as informações quantitativas a respeito da descrição desses pacientes utilizados no estudo.

Tabela 2 – Demografia dos pacientes comatosos analisados.

	Desfecho Favorável	Desfecho Óbito Clínico	Desfecho Óbito Encefálico
AVE	5	10	2
TCE	16	14	3
Encefalopatia / Coma Metabólico	5	15	3
Outras Etiologias	2	10	5
Desconhecida	13	15	2
Σ	41	64	15

Como exibido na tabela 2, a maioria dos pacientes (64) morrem, 41 sobrevivem e 15 tem morte cerebral. Steppacher (STEPPACHER; KAPS; KISSLER, 2016) mostra que as chances de sobrevivência sem nenhuma sequela é de 15%, enquanto sobrevivência, em geral é de 35% o que condiz com os dados.

Quanto às causas do coma, 26 foram por Derrame Cerebral (AVE), 40 por Trauma Cranioencefálico (TCE), 22 provável coma metabólico, 11 por Encefalopatia e 21 por outros motivos.

A maior parte dos pacientes observados são do sexo masculino (86 Homens contra 34 Mulheres). A idade média gira em torno de 52, o paciente mais novo tem 16 anos e o mais velho 92.

3.2 Ambiente Experimental e Métricas de Desempenho Preditivo

É importante notar que devido à falta de biblioteca relacionada ao *reservoir*, todo o código foi desenvolvido pelo autor, a linguagem escolhida foi o Python, pela praticidade e pelas inúmeras bibliotecas de aprendizado de máquina que podem ser utilizadas. Para os algoritmos de classificação e as métricas de desempenho é utilizada a biblioteca da Sklearn.

Cada simulação é conduzida através de uma validação cruzada estratificada, onde toda a base de dados é dividida em k partes disjuntas, e são feitas k execuções, onde cada iteração um conjunto é utilizado para teste enquanto os outros para treino. O desempenho preditivo exibido na equação é a média do resultado de 3 execuções da validação cruzada, pegando as "folders" randomicamente a cada rodada.

As métricas utilizadas são, acurácia, precisão, recall e F1-Macro. O cálculo de acurácia exibido na equação 16, pode ser interpretado como sendo a quantidade de rótulos classificados corretamente divididos pelo total de objetos.

Nas equações 17 e 18, TP quer dizer verdadeiro positivo, ou seja, indica a quantidade de objetos que o modelo classificou como pertencente a uma classe sendo verdadeiramente dessa classe. FP, por outro lado, indica falso positivo, sendo aqueles respondidos pelo método como positivo, porém, pertencem a outra classe. Diante disso, a precisão refere-se a quantidade de objetos que o método classificou e acertou pertencente a classe perante todos que foram classificados como pertencentes à tal classe.

FN diz respeito aos falsos negativos, sendo os objetos pertencentes a essa classe, mas classificados como sendo de outra. Portanto, recall diz respeito a quantidade de objetos classificados corretamente em relação a todos os objetos pertencentes a classe.

O F1-Macro de cada classe é exibido na equação 19, enquanto o F1-Macro final que é a média do F1-Macro de todas as classes é mostrada na equação 21. A média dos F1-Macros pode ser ponderada de acordo com o número de elementos em cada base de

dados, valorizando o grupo com maior quantidade de elementos. Contudo, no cenário do presente estudo, onde a base de dados é desbalanceado e objetiva-se um resultado balanceado entre as classes (C), foram dados pesos iguais a cada classe. Em 20, portanto, $P(k) = 1$ para todas as classes e o F score utilizado é apresentado em 22.

$$\text{acurácia} = \frac{TP + TN}{TP + FP + TN + FN} \quad (16)$$

$$\text{Precisão} = \frac{TP}{TP + FP} \quad (17)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (18)$$

$$F_{score}(k) = 2 \times \frac{\text{Precisão} \times \text{Recall}}{\text{Precisão} + \text{Recall}} \quad (19)$$

$$M = \sum_{k=1}^C P(k) \quad (20)$$

$$F_{score} = \sum_{k=1}^C \frac{P(k) \times F_{score}(k)}{M} \quad (21)$$

$$F_{score} = \sum_{k=1}^C \frac{F_{score}(k)}{C} \quad (22)$$

Pelo fato da ESN ser CNN métrica ponderada foi utilizada para que os resultados avaliem ambas as classes de forma equivalente. Um reflexo desse fato é que a acurácia não será relevante nas análises dos resultados uma vez que o problema trabalhado é consideravelmente difícil de mapear devido a inerências contidas nos exames (alta dimensionalidade, presença de artefatos, sensibilidade do exame, complexidade dos sinais cerebrais). Em casos como esse, é comum, altos valores de acurácia para classificações em que uma classe tem maiores representantes que outra, induzindo à conclusão errônea. Sendo assim, o foco dos resultados será na métrica do F1-Macro.

Duas possíveis análises foram abordadas nesse estudo. A primeira considera dois grupos de análises, sendo baseada apenas em desfecho favorável (N=41) versus desfecho desfavorável (N=79). A segunda abordagem considera os três possíveis desfechos, como mostrado na Tabela 2. Então testes com as duas formas de abordagem foram realizados. Os métodos investigados serão propostos utilizando a ESN e a CNN.

3.3 Uma Abordagem Benchmark para o PPC

O modelo gerado nesse capítulo leva em consideração os quatro quantificadores descritos no capítulo anterior: Porcentagem de Contribuição de Potência (PCP), variação da

potência cerebral (VPC), Frequência Mediana (FM) e Coerência. Esses quantificadores foram adotados em trabalhos relacionados, tais como (RAMOS; CARNEIRO; DESTRO-FILHO, 2021), e os atributos extraídos a partir deles são submetidos a oito diferentes classificadores cujos parâmetros avaliados são apresentados na Tabela 3.

Tabela 3 – Lista de parâmetros testados para cada técnica no readout e o parâmetro selecionado com melhor resultado.

Algoritmos	Lista de parâmetros	Selecionado
kNN	$k \in (1, 3, 5, 7, 9)$	$k = 1$
LSVM	$C \in (2^{-4}, 2^0, 2^4, 2^8)$	$C = 2^8$
LR	$C \in (2^{-4}, 2^0, 2^4, 2^8)$	$C = 2^8$
	$pen \in (l1, l2)$	$p = l1$
GSVM	$C \in (2^{-4}, 2^0, 2^4, 2^8)$	$C = 2^8$
	$\gamma \in (2^{-4}, 2^{-2}, 2^0, 2^2)$	$\gamma = 1$
RF	$t \in (2^6, 2^8, 2^{10})$	$t = 2^{10}$
MLP	$\alpha \in (0.0005, 0.002)$	$\alpha = 0.0005$
	$m \in (2n, 8n)$	$m = 2n$

3.3.1 Resultados PPC para Dois Desfechos

A Tabela 4 apresenta os resultados da abordagem baseada em quantificadores para a formulação do problema considerando dois desfechos. O melhor resultado utiliza o algoritmo MLP, alcançando 58.9% de F1-Macro. A floresta aleatória e a GSVM, ambos algoritmos estado-da-arte para várias tarefas de classificação, estão entre os piores resultados.

Tabela 4 – Resultados utilizando os métodos considerados estado-da-arte com o EEG utilizando 2 desfechos na base de dados

Classificador	Acc.	Prec.	Rec.	F1-macro
KNN	0.561	0.517	0.520	0.513 ± 0.15
LSVM	0.592	0.549	0.554	0.541 ± 0.19
LR	0.589	0.539	0.543	0.533 ± 0.16
GSVM	0.650	0.523	0.521	0.471 ± 0.13
DT	0.592	0.555	0.553	0.546 ± 0.18
RF	0.617	0.521	0.517	0.498 ± 0.21
MLP	0.636	0.604	0.596	0.589 ± 0.21
NB	0.494	0.563	0.557	0.490 ± 0.20

3.3.2 Resultados PPC para Três Desfechos

A Tabela 5 apresenta os resultados da abordagem baseada em quantificadores para a formulação do problema considerando três desfechos. Novamente, a MLP alcança o melhor

resultado com F1-Macro de 35%. A MLP pode se adaptar e criar divisões no espaço de busca da solução mais especializada nos dados, tendo mais liberdade e gerando separações dos espaços que outros algoritmos têm dificuldade em fazer. Por outro lado, o espaço não linear gerado pela GSVM não permitiu as transformações necessárias nos atributos de modo a facilitar a separação dos grupos para classificação dos dados. Os resultados na tabela representam um exemplo perfeito de como a acurácia pode ser uma medida falha para o PPC. Enquanto GSVM obteve o pior F1-Macro entre todos os classificadores tanto para dois quanto três desfechos, ele também obteve a melhor acurácia entre todos. O que acontece é que tal método rotula todos os objetos como a classe majoritária. Uma vez que a base de dados é desbalanceada, isso pode prover uma interpretação equivocada em termo de acerto.

Tabela 5 – Resultados utilizando os métodos considerados estado da arte com o EEG utilizando 3 desfechos na base de dados

Classificador	Acc.	Prec.	Rec.	F1-macro
KNN	0.431	0.283	0.323	0.298 ± 0.15
LSVM	0.403	0.335	0.352	0.332 ± 0.17
LR	0.419	0.349	0.374	0.346 ± 0.16
GSVM	0.497	0.280	0.329	0.277 ± 0.10
DT	0.422	0.335	0.342	0.333 ± 0.18
RF	0.453	0.285	0.304	0.265 ± 0.10
MLP	0.436	0.361	0.361	0.350 ± 0.20
NB	0.411	0.377	0.396	0.344 ± 0.15

Redes *echo state* regulares para o PPC

Este capítulo apresenta um estudo comparativo realizado com ESN. A pesquisa visa avaliar o reservóir com diferentes estruturas além da aleatória bem como com diferentes tipos de classificadores na camada readout (BISSARO; JIN; CARNEIRO, 2021). Os modelos com melhor desempenho preditivo são adotados para aplicação no PPC considerando as informações capturadas individualmente pelos eletrodos do EEG e no capítulo seguinte para o projeto e desenvolvimento de novas arquiteturas ESN.

4.1 Análise comparativa de ESNs em termos de reservóir e readout

Neste trabalho, as estruturas de rede, regular, pequeno mundo e randômica são investigadas como estruturas do reservóir, para avaliar suas habilidades de processar informações a um contexto específico em que foi inspirada pela conectividade do cortical neural humana. Estas mesmas estruturas de rede são investigados em outros trabalhos, apesar de serem como estratégias baseadas em propriedades de livre escala (KAWAI et al., 2017) e sistemas hierárquicos distribuídos (DENG; ZHANG, 2006).

Diferente da maioria dos trabalhos relacionados a ESN, os quais são focados no desenvolvimento de uma característica adicional ao modelo, ou inspirados por modelos biológicos, nós avaliamos aqui uma alternativa aos componentes relacionados à estrutura do reservóir, ao aprendizado da camada readout e as configurações de parâmetros com foco especial no aspecto do aprendizado de máquina.

Do ponto de vista estrutural, nós investigamos os modelos de rede regular e pequeno mundo contra o randômico, que é o adotado usualmente. Na camada de readout, nós analisamos um total de oito diferentes técnicas de classificação na tarefa de mapear os dados transformados para a saída correspondente desejada. Em relação aos parâmetros, nós sistematicamente consideramos uma quantidade considerável de configurações para ambos reservóir e camada readout. De modo a avaliar consistentemente os componentes

alternativos para o ESN, nós conduzimos experimentos com 5 aplicações do mundo real, que englobam características muito distintas em termos de domínio, tamanho de exemplos e números de características e classes.

4.1.1 Modelos de rede de reservatório

No estudo consideramos a estrutura de três diferentes modelos de rede: regular, pequeno mundo e aleatória. A Figura 16 apresenta em uma ilustração um exemplo de cada um dos três modelos. No contexto do reservatório cada vértice denota um neurônio e cada aresta uma ligação entre dois neurônios. Na figura, é possível perceber a mudança de uma estrutura bem definida para uma estrutura desorganizada: o modelo regular provê uma estrutura de rede fixa com conexões organizadas; a rede pequeno mundo adiciona a um pouco de aleatoriedade em algumas conexões, mas que apresenta um alto grau de organização; e a randômica exibe uma rede caótica. Discutiremos a seguir um pouco de cada rede:

- No modelo Regular (Reg), mostrado na Figura 16(a), cada neurônio na rede tem o mesmo número k de conexões, i.e, o mesmo grau.
- No modelo pequeno mundo (SW) as conexões de uma rede k -regular é modificada considerando uma taxa aleatória ρ no qual os valores usualmente vão de $0 < \rho < 0.1$ (WATTS; STROGATZ, 1998). A Figura 16(b) exibe uma rede pequeno mundo com $\rho = 0.1$
- Na rede aleatória (Rand) as conexões são definidas randomicamente em torno de um parâmetro de densidade d , que se refere a porcentagem de arestas existentes na rede em relação a todas as possibilidades. Esta é a principal estrutura utilizada pelo reservatório altamente utilizada na literatura. A Figura 16(c) exibe uma rede aleatória.

4.1.2 Classificadores readout

As informações processadas pela estrutura do reservatório são analisadas por técnicas de classificação que com os novos dados, vão conter as informações temporais intrínsecas à cada objeto. A saída pode ser usualmente separada por métodos lineares, mas isso não quer dizer que outros métodos não podem ter melhor desempenho (JAEGER, 2005).

Neste trabalho oito algoritmos foram avaliados para a camada de Readout: K vizinhos mais próximos (kNN), máquina de vetor de suporte linear (LSVM), regressão logística (LR), máquina de vetor de suporte RBF (GSVM), Árvore de decisão (DT), Floresta Aleatória (RF), Perceptron Multicamadas (MLP) e Naive Bayes(NB). Nós consideramos no trabalho a influência tanto das estruturas, quanto das técnicas em consideração para a performance do ESN.

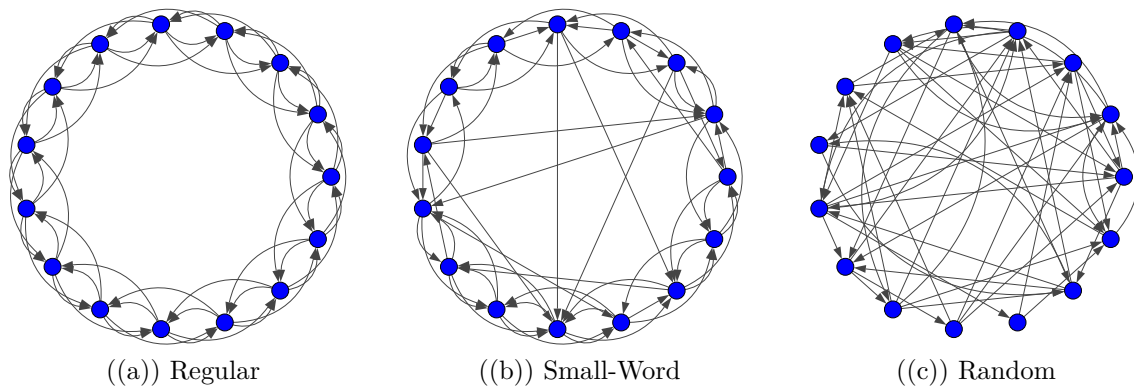


Figura 16 – Exemplo das estruturas do reservoir geradas com modelos de rede distintos. a) Na regular, os neurônios são conectados com seus k -vizinhos mais próximos. b) Na pequeno mundo, cada neurônio tem uma probabilidade $p = 0.1$ de trocar seus links depois de terem sido conectadas pelos seus k -vizinhos mais próximos; e c) Na aleatória, as conexões são totalmente aleatórias.

4.1.3 Parâmetros considerados

No estudo avaliamos rigorosamente dois grupos de parâmetros relacionados tanto a estrutura do reservoir quanto a camada de classificadores. com relação à estrutura do reservoir temos três parâmetros: o número de neurônios N , o número de conexões dado por $k = d * 8$, onde d é referente a porcentagem de conexões existentes na rede, que também deve ser selecionada de forma a manter a propriedade Echo State, e o parâmetro σ , representando o desvio padrão dos valores contidos nas matrizes de peso ($W^{(in)}$ e W) geradas randomicamente a partir de uma distribuição normal com média 0. Com relação à camada readout, cada técnica tem seus próprios parâmetros, que foram cuidadosamente selecionados pelo método “grid search” sobre um conjunto representativo de valores.

4.2 Resultados em Bases de Dados Reais

Em seguida descrevemos os experimentos conduzidos nas estruturas do reservoir apresentadas anteriormente. A Tabela 6 apresenta as informações a respeito das 5 bases de dados selecionadas para o estudo (DUA; GRAFF, 2017; CHEN et al., 2015). Como é possível perceber, tal seleção visa englobar diversidade nos dados, com diferentes números de classes, atributos, e tamanhos (variam de 3 a 387, 8 a 91 e 360 a 27619, respectivamente). Apresentamos a seguir uma breve introdução sobre cada base de dados:

- Dataset acelerômetro: cada objeto representa informações de um sensor acelerômetro no tempo T , preso em um paciente. A classe representa onde o paciente está, podendo ser, deitado, sentado na cama, e em pé. O dataset foi proposto para com objetivo de conseguir identificar a localização do paciente no quarto, e prever

quando ele está se levantando, de formar a dar algum auxílio neste momento e evitar uma queda.

- ❑ Dataset Libras: Libras é a língua brasileira de sinais, largamente usada por pessoas surdas nos centros urbanos Brasileiros. Cada palavra é convertida em um sinal. Um sinal pode ser visto como uma sequência de movimentos usando as mãos e braços. Note que uma sequência de movimentos também pode formar uma frase, mas não é o foco do dataset. O dataset foi adquirido mapeando movimentos reais em vídeos e para cada frame foi gerado 90 atributos representando as coordenadas do movimento.
- ❑ Dataset RSSI: os objetos são escritos por 13 atributos, cada um representa a distância do objeto até um sensor específico. Os sensores foram posicionados no primeiro andar da biblioteca da universidade Western Michigan. O andar foi mapeado em quadrantes, e o objetivo é descobrir em qual quadrante está.
- ❑ Dataset Wall-Following: um robô tentando prever para onde se mover utilizando 24 sensores de sensor ultrassônico. Os dados são coletados através de um robô navegando em uma sala seguindo a parede dando 4 voltas na direção horária.
- ❑ Dataset Ozônio: os objetos são caracterizados por 73 características relacionadas ao dia, como temperatura, velocidade do vento, radiação solar e estatísticas de emissões, recomendadas pela Comissão de qualidade do ambiente do Texas para monitorar picos de ozônio. O modelo deve aprender a prever os picos de ozônio local em dois grupos: dia de Ozônio ou dia normal.

Tabela 6 – Metadados das bases de dados consideradas no estudo.

Data set	#Amostras	#Atributos	#Classes
Acelerômetro	27619	8	3
LIBRAS	360	91	48
RSSI	1420	15	378
Wall-Following	5456	24	4
Ozônio	2536	73	2

Nos experimentos consideramos as configurações de parâmetros relacionadas a ambos reservatório e classificadores. No que diz respeito aos modelos de rede, consideramos os números de neurônios $n \in \{100, 200, 400\}$, o desvio padrão $\sigma \in \{0.08, 0.15, 0.22\}$ e a densidade $d \in 0.1$. A estrutura SW foi gerada com $\rho = 0.1$.

Com relação aos algoritmos avaliados na camada readout: kNN tem o número de vizinhos k ; LR tem a regularização C e a penalidade normal pen ; LSVM(linear) tem o parâmetro de regularização C ; GSVM (kernel rbf) tem a regularização C e o coeficiente

do kernel γ ; RF tem o número de árvores t ; MLP tem a taxa de aprendizado α ; DT e NB não tem parâmetros para selecionar. A Tabela 7 lista os possíveis valores dos parâmetros de cada técnica bem como aqueles selecionados pela nossa avaliação experimental.

Tabela 7 – Lista de parâmetros testados para cada técnica no readout e o parâmetro selecionado com melhor resultado.

Algoritmos	Lista de parâmetros	Selecionado
kNN	$k \in (1, 3, 5, 7, 9)$	$k = 1$
LSVM	$C \in (2^{-4}, 2^0, 2^4, 2^8)$	$C = 2^8$
LR	$C \in (2^{-4}, 2^0, 2^4, 2^8)$	$C = 2^8$
	$pen \in (l1, l2)$	$p = l1$
GSVM	$C \in (2^{-4}, 2^0, 2^4, 2^8)$	$C = 2^8$
	$\gamma \in (2^{-4}, 2^{-2}, 2^0, 2^2)$	$\gamma = 1$
RF	$t \in (2^6, 2^8, 2^{10})$	$t = 2^{10}$
MLP	$\alpha \in (0.0005, 0.002)$	$\alpha = 0.0005$
	$m \in (2n, 8n)$	$m = 2n$

Análise de parâmetros do reservóir: Para analisar milhares de resultados, nós adotamos o teste de *Friendman*, que permite comparar múltiplas técnicas com múltiplos datasets (DEMŠAR, 2006). Primeiramente, nós avaliamos as configurações de $n; \sigma$ na estrutura do reservóir. A hipótese nula diz que diferentes configurações de $n; \sigma$ são equivalentes. Utilizando um nível de significância de 0.05, a hipótese nula é rejeitada. O teste de Nemenyi foi utilizado para mostrar quais configurações são não equivalentes. A Figura 17 apresenta o diagrama de diferença crítica obtido comparando todas as configurações uma contra a outra. O teste indica que as configurações $n = 200/\sigma = 0.22$ e $n = 400/\sigma = 0.15$; possuem melhor desempenho que as outras configurações, com exceção da configuração $n = 200/\sigma = 0.15$, que é estatisticamente equivalente as outras duas. Entretanto, pode notar-se que altos valores desses parâmetros atingem melhores resultados.

4.2.1 Análise dos classificadores readout

Foram avaliadas as performances de 8 diferentes classificadores e suas configurações de parâmetros. Os resultados das três diferentes arquiteturas do ESN, considerando somente o melhor (ou equivalente) conjunto de parâmetros para cada algoritmo de acordo com os resultados estatísticos, são apresentados na Tabela 8.

A Tabela 8 exhibe os resultados em função das estruturas de reservóir e dos algoritmos da camada readout. Na tabela, “Reg”, “SW” e “Rand”, dizem respeito às estruturas, regular, pequeno mundo e aleatória respectivamente. Para cada base de dados, o melhor resultado entre as três configurações está sublinhado e o melhor resultado está em negrito.

Pela tabela, pode-se notar que o GSVM tem o melhor resultado para a maioria dos datasets, mesmo considerando as diferentes estruturas, é interessante que o kNN também

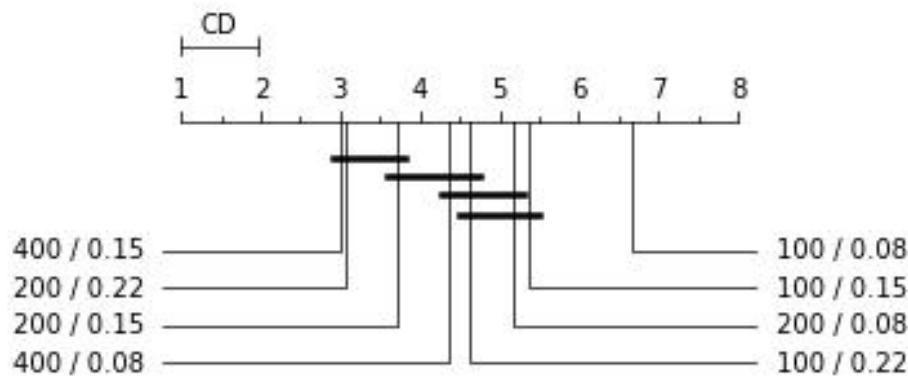


Figura 17 – Diagrama de diferença crítica obtida pelo teste post-hoc de Nemenyi, considerando as diferentes configurações de parâmetros na estrutura do reservóir.

apresenta bons resultados em comparação com outros modelos altamente adotados como classificadores do reservóir, como regressão linear e o SVM linear.

Além disso, o kNN parece altamente indicado com o modelo regular, já que o mesmo obtém 4 dos 5 melhores resultados gerais com essa configuração. A LR consegue todos os melhores resultados com a estrutura regular, enquanto o LSVM tem 3 dos 5 melhores resultados com a estrutura randômica.

No que diz respeito às análises estatísticas, a hipótese nula diz que os resultados de alguns dos classificadores são equivalentes. Com um nível de significância de 0.05, a hipótese nula é rejeitada. Como é mostrado no diagrama apresentado na Figura 18, o teste post-hoc de Nemenyi revela que o GSVM supera todos os classificadores com exceção do kNN e do LSVM, onde ambos tem os resultados equivalentes ao do GSVM. Em poucas palavras, o GSVM demonstra uma performance consistente em todo o dataset.

Para o Acelerômetro, o GSVM com a estrutura SW provê o melhor resultado, seguida de perto pelo GSVM com a estrutura randômica. Para o dataset de LIBRAS, a estrutura Regular com o GSVM tem o melhor resultado, seguida pela GSVM com a SW. Para o dataset RSSI, o melhor resultado foi o GSVM com a regular. Para o dataset Wall-Following existe um empate entre os algoritmos LR e GSVM usando as estruturas Reg e SW respectivamente. Por último, a estrutura SW com o classificador LSVM alcançou o melhor resultado para o dataset Ozônio, seguido de perto pelos algoritmos kNN e GSVM.

Uma técnica que merece uma atenção é o kNN que atinge ótima performance com um custo computacional baixíssimo $O(N)$ contra o GSVM que é $O(N^3)$. Diferente do GSVM, kNN e LSVM, outros classificadores tiveram problemas para aprender os padrões diretamente do reservóir, especialmente o NB, que apresenta os piores resultados para os datasets Acelerômetro, RSSI, Wall-Following e Ozônio.

Tabela 8 – F1 Macro para os oito classificadores com as cinco bases de dados reais. Em negrito estão os melhores resultados para cada base, e sublinhado estão os melhores resultados para configuração de rede em cada método.

Alg.	Accelerometer			LIBRAS			RSSI			Wall-Following Robot			Ozone		
	Rand.	Reg.	SW	Rand.	Reg.	SW	Rand.	Reg.	SW	Rand.	Reg.	SW	Rand.	Reg.	SW
KNN	94.27	94.66	94.46	93.31	94.93	94.53	73.45	74.01	74.88	90.16	90.62	90.47	93.32	93.64	93.05
LSVM	<u>87.29</u>	86.41	87.22	<u>95.84</u>	<u>95.56</u>	95.35	<u>69.26</u>	68.52	67.32	89.73	<u>90.81</u>	90.53	93.36	93.51	93.79
LR	78.83	79.62	78.43	70.43	70.92	70.54	23.8	24.78	24.16	90.27	90.83	90.69	89.04	89.23	89.23
GSVM	96.83	96.65	96.96	95.9	97.19	97.11	74.95	77.01	76.51	89.97	90.72	90.83	<u>93.24</u>	93.14	92.91
DT	86.55	87.63	<u>87.65</u>	54.37	54.35	59.44	<u>16.32</u>	15.79	15.57	90.45	<u>90.56</u>	90.41	78.89	80.46	<u>80.88</u>
RF	<u>73.48</u>	73.46	73.27	82.89	82.54	<u>83.44</u>	<u>44.81</u>	43.19	44.04	90.0	<u>90.34</u>	90.29	78.21	<u>78.37</u>	78.29
MLP	<u>77.37</u>	<u>80.33</u>	78.31	76.01	75.8	<u>77.06</u>	<u>47.52</u>	39.74	42.52	89.99	90.06	<u>90.29</u>	89.22	88.89	<u>89.27</u>
NB	<u>69.9</u>	69.72	69.21	39.4	<u>44.84</u>	41.54	28.55	<u>29.42</u>	28.85	44.18	<u>44.85</u>	<u>44.85</u>	62.66	<u>64.88</u>	62.28

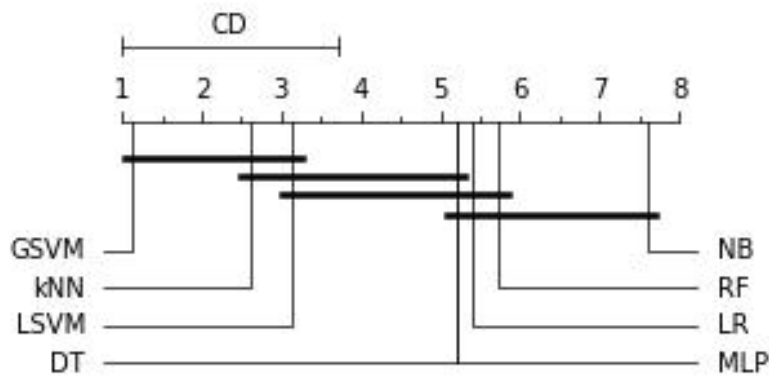


Figura 18 – Diagrama de diferença crítica obtida pelo teste post-hoc de Nemenyi, considerando as diferentes técnicas de classificação para a camada readout.

4.2.2 Análises do modelo de rede

Analisando estatisticamente os modelos de rede investigados neste estudo, onde a hipótese nula do teste de Friedman diz que não existe diferença significativa entre os modelos, Regular, SW e randômico. Sob um nível de significância 0.05, a hipótese nula é rejeitada.

Depois de aplicar um teste post-hoc de Nemenyi, o diagrama de diferença crítica mostrada na Figura 19 é obtido. Na imagem, é possível ver que a rede regular supera estatisticamente a rede randômica, apesar de ambas serem estatisticamente equivalentes à rede pequeno mundo.

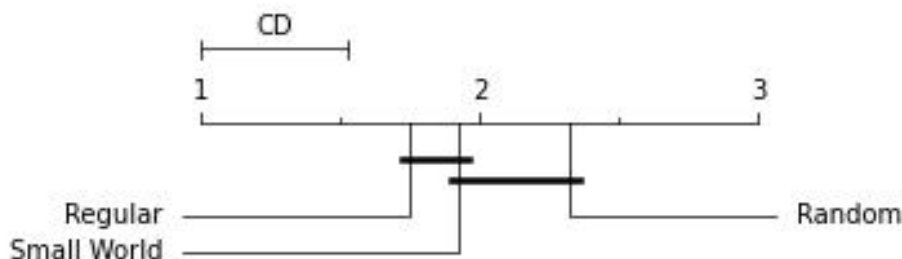


Figura 19 – Diagrama de diferença crítica obtida a partir do teste post-hoc de Nemenyi considerando a randômica, a pequeno mundo e a regular.

Analisando estatisticamente a estrutura em função dos três melhores classificadores: GSVM, kNN e LSVM, tem-se a intenção de comparar os modelos em pares usando todos datasets, o teste de Wilcoxon foi adotado. Basicamente, o teste calcula a diferença dos resultados dos dois métodos para cada dataset e compara o rank com as diferenças positivas e negativas. A hipótese nula diz que os ranks obtidos por cada dois modelos de rede (Rand vs Reg, Reg vs SW, SW vs Rand) são similares. O p-valor encontrado pelo teste de Wilcoxon é exibido na Tabela 9.

O teste falhou sob um nível de significância de 0.05, porém sob significância de 0.1, o modelo regular é estatisticamente superior ao randômico. Além disso, analisando os resultados apresentados na Tabela 8, é possível perceber que os melhores resultados são obtidos pelos modelos regular e pequeno mundo, especialmente nos classificadores não lineares kNN e GSVM. Isso indica que o alto nível de organização na estrutura do reservoir fornecido pela rede pequeno mundo e regular pode beneficiar muitas aplicações reais e permitir uma criação mais simples e organizada do reservoir.

Tabela 9 – P-valores encontrados para o teste de Wilcoxon, quando considera os resultados dos três melhores métodos na readout: GSVM, kNN, LSVM.

#	Rand.	Reg.
Reg.	0.069	-
SW	0.139	0.410

4.2.3 Análise da contribuição do reservoir

Agora avaliamos a relevância da estrutura do reservoir em termos de performance comparando os resultados anteriores com os obtidos pela execução direta dos algoritmos da camada readout, ou seja, sem utilizar o reservoir. A Tabela 9 apresenta o Fscore médio ponderado com os resultados dos algoritmos GSVM, kNN e LSVM com e sem o reservoir. Como é possível notar, com exceção do dataset de Ozônio, o reservoir causa uma melhora significativa nos resultados.

Na base de dados RSSI o F1-macro obtido com reservoir é pelo menos o dobro maior que o obtido sem o reservoir. Esse resultado que exibe a qualidade da ESN visto que ambos os modelos (com e sem reservoir) tiveram os parâmetros otimizados. Mais ainda, o teste de Wilcoxon atesta que o modelo com o reservoir é estatisticamente melhor que sem o reservoir (p-valor < 0.01).

Tabela 10 – F1 Macro para os 3 melhores classificadores na camada readout para as cinco bases de dados reais, considerando dois modelos: Com (Res.) e sem (No-Res.) a estrutura do reservoir. Em negrito estão os melhores resultados para cada base, e sublinhado denota qual modelo prove o melhor resultado.

Data sets	KNN		LSVM		GSVM	
	Res.	No-Res.	Res.	No-Res.	Res.	No-Res.
Accelerometer	<u>94.66</u>	90.66	<u>86.41</u>	77.32	96.65	91.63
LIBRAS	<u>94.93</u>	91.86	<u>95.56</u>	91.50	97.19	93.43
RSSI	<u>74.01</u>	32.74	<u>68.52</u>	26.38	77.01	32.98
Wall-Following Robot	<u>90.62</u>	88.30	90.81	73.23	<u>90.72</u>	87.25
Ozone	93.64	<u>95.55</u>	93.51	<u>94.10</u>	93.16	95.84

4.3 Aplicação para o PPC usando sinais de EEG

Baseado nos resultados apresentados na seção anterior, escolhemos às três melhores configurações de redes *echo state* para aplicação no problema de prognóstico de pacientes em coma. Especificamente, o reservóir regular é utilizado para processar os sinais EEG capturados a partir de cada eletrodo separadamente. Apesar do exame EEG compreender um total de 20 eletrodos, os neurologistas identificaram que alguns deles continham ruídos severos ou tiveram seus valores zerados para alguns pacientes. Nesse sentido, foram selecionados apenas os eletrodos P3, P4, O1, O2, CZ e OZ para essa análise individual.

4.3.1 Descrição do Modelo

Processando os eletrodos separadamente, temos uma série temporal única associada a uma classe do problema. Formalmente, essa série é definida pelo vetor $T' = \{T'_1, T'_2, T'_3, \dots, T'_n\}$, sendo n o número de passos de tempo considerado para o exame. Diferente das bases reais avaliadas anteriormente, nas quais cada amostra era denotada pelas suas características registradas em um passo de tempo específico, o desafio aqui é projetar um modelo de representação capaz de usufruir do processamento sequencial realizado pelas redes *echo state*.

Nesse sentido, desenvolvemos uma função de adaptação do sinal EEG, definida por $fa(T') \rightarrow T$, a qual é responsável por dividir uma série $T'_{1 \times n}$ em p séries de tamanho m , com $m = n/p$. Dessa forma, o resultado dessa função é uma matriz de séries temporais $T_{p \times m}$, na qual cada linha representa um trecho da atividade cerebral registrada pelo eletrodo em um dado intervalo de tempo m , tal como o exemplo a seguir:

$$T = \begin{bmatrix} T_{11} & T_{12} & T_{13} & \dots & T_{1m} \\ T_{21} & T_{22} & T_{23} & \dots & T_{2m} \\ T_{31} & T_{32} & T_{33} & \dots & T_{3m} \\ \dots & \dots & \dots & \dots & \dots \\ T_{p1} & T_{p2} & T_{p3} & \dots & T_{pm} \end{bmatrix}$$

O processamento dessa matriz T' pelo reservóir ocorre então da seguinte forma: a cada iteração $i = \{1, 2, \dots, p\}$, uma sequência T'_i é processada pelo reservóir, resultando em um vetor de saída de tamanho r . Em seguida, todos os vetores de saída obtidos, conforme a matriz $R_{p \times r}$ abaixo, são achatados em um único vetor e utilizados como entrada para os algoritmos de classificação na camada de readout.

$$R = \begin{bmatrix} R_{11} & R_{12} & R_{13} & \dots & R_{1r} \\ R_{21} & R_{22} & R_{23} & \dots & R_{2r} \\ R_{31} & R_{32} & R_{33} & \dots & R_{3r} \\ \dots & \dots & \dots & \dots & \dots \\ R_{p1} & R_{p2} & R_{p3} & \dots & R_{pr} \end{bmatrix} .$$

4.3.2 Resultados PPC (por eletrodo) para Dois Desfechos

Nossa primeira análise é realizada para o PPC considerando dois desfechos. A Tabela 11 apresenta os resultados das redes *echo state* regular para os sinais dos eletrodos analisados considerando os classificadores KNN, LSVM e GSVM. Nas simulações, o número de divisões (Div.) de cada exame é selecionado entre $\{10, 20, 40\}$. Apesar de apresentar acurácia maior que outros classificadores, GSVM possui o pior desempenho preditivo, conforme aponta a medida F1-macro, uma vez que classifica todos os exames de teste no mesmo desfecho. O oposto acontece com LSVM, que alcança o melhor desempenho preditivo em termos de F1-macro para todos os eletrodos considerados. Os melhores resultados foram obtidos a partir dos dois eletrodos na linha média, CZ e OZ. Este último inclusive alcançou 0.503 de F1-macro, o que é um resultado interessante considerando que apenas as informações capturas por um único eletrodo foram analisadas e também que não houve nenhum procedimento específico de extração de características.

Tabela 11 – Resultados do reservóir utilizando um único eletrodo e com a base de dados dividida em dois desfechos.

Eletrodo	Readout	Div.	Acc.	Prec.	Rec.	F1-macro
P3	KNN	10	0.6528	0.5067	0.4753	0.4323
	LSVM	40	0.5917	0.4790	0.4371	<u>0.4416</u>
	GSVM	10	0.6583	0.5000	0.3292	0.3969
P4	KNN	10	0.6444	0.4933	0.3721	0.4037
	LSVM	10	0.6250	0.5056	0.4846	<u>0.4644</u>
	GSVM	10	0.6583	0.5000	0.3292	0.3969
O1	KNN	20	0.6556	0.5058	0.4472	0.4227
	LSVM	10	0.6167	0.5131	0.5122	<u>0.4845</u>
	GSVM	10	0.6583	0.5000	0.3292	0.3969
O2	KNN	40	0.6417	0.4978	0.4285	0.4228
	LSVM	40	0.6250	0.5095	0.4997	<u>0.4695</u>
	GSVM	10	0.6583	0.5000	0.3292	0.3969
CZ	KNN	20	0.6667	0.5167	0.4831	0.4346
	LSVM	20	0.6444	0.5275	0.5455	<u>0.4967</u>
	GSVM	10	0.6583	0.5000	0.3292	0.3969
OZ	KNN	40	0.6667	0.5241	0.5738	0.4586
	LSVM	10	0.6389	0.5268	0.6022	<u>0.5026</u>
	GSVM	10	0.6583	0.5000	0.3292	0.3969

A Tabela 12 compara o melhor resultado nos algoritmos puros com o melhor resultado utilizando só um eletrodo (OZ) que utiliza o algoritmo LSVM. O resultado do reservóir não conseguiu alcançar 0.5886 de F1-Macro do baseline, apesar da acurácia parecida.

Tabela 12 – Resultado comparando o melhor método de aprendizado de máquina sem o uso de aprendizado profundo, e a melhor execução da rede recorrente utilizando somente um eletrodo para dois desfechos.

Modelo (Readout)	Eletrodo	Div.	Acc.	Prec.	Rec.	F1-macro
MLP	Todos	-	0.6361	0.6041	0.5957	0.5886
ESN (LSVM)	OZ	10	0.6389	0.5268	0.6022	0.5026

4.3.3 Resultados PPC (por eletrodo) para Três Desfechos

A Tabela 13 apresenta os resultados das ESN para o PPC com três desfechos. Os parâmetros foram os mesmos considerados na simulação anterior. GSVM apresentou a mesma limitação de antes, classificando mais uma vez todos os exames de teste para um único desfecho. Por outro lado, KNN e LSVM alternaram os melhores resultados por eletrodo, com destaque para os eletrodos localizados à esquerda: P3 e O1, nos quais a ESN regular com readout KNN alcançou respectivamente 0.348 e 0.351 de F1-macro. Esses resultados são bastante relevantes quando comparados com aqueles obtidos por métodos de classificação tradicionais. Por exemplo, é possível ver na Tabela 14 que o resultado utilizando um único eletrodo (O1) com o algoritmo LSVM obtém o mesmo resultado do baseline em termos de F1-macro.

Tabela 13 – Resultados do reservatório utilizando um único eletrodo e com a base de dados dividida em três desfechos.

Eletrodo	Readout	#Div	Acc.	Prec.	Rec.	F1-macro
P3	KNN	40	0.3889	0.4525	0.4247	<u>0.3483</u>
	LSVM	10	0.4611	0.3396	0.3277	0.3239
	GSVM	10	0.5333	0.3333	0.1778	0.2318
P4	KNN	40	0.3278	0.4167	0.3163	0.2879
	LSVM	20	0.4417	0.3165	0.2938	<u>0.2971</u>
	GSVM	10	0.5333	0.3333	0.1778	0.2318
O1	KNN	20	0.4667	0.4003	0.3769	0.3505
	LSVM	40	0.4556	0.3361	0.3109	0.3198
	GSVM	10	0.5333	0.3333	0.1778	0.2318
O2	KNN	40	0.3139	0.4349	0.3793	0.3031
	LSVM	10	0.4389	0.3233	0.3090	<u>0.3074</u>
	GSVM	10	0.5333	0.3333	0.1778	0.2318
CZ	KNN	40	0.3694	0.4757	0.4340	<u>0.3399</u>
	LSVM	40	0.4333	0.3439	0.3492	0.3369
	GSVM	10	0.5333	0.3333	0.1778	0.2318
OZ	KNN	40	0.3278	0.4087	0.3917	0.3098
	LSVM	40	0.4333	0.3346	0.3258	<u>0.3229</u>
	GSVM	10	0.5333	0.3333	0.1778	0.2318

Tabela 14 – Resultado comparando o melhor método de aprendizado de máquina sem o uso de aprendizado profundo, e a melhor execução da rede recorrente utilizando somente um eletrodo para três desfechos.

Modelo (Readout)	Eletrodo	Div.	Acc.	Prec.	Rec.	F1-macro
MLP	Todos	-	0.4361	0.3607	0.3611	0.3505
ESN (LSVM)	O1	20	0.4667	0.4003	0.3769	0.3505

Redes echo state multi-sequências para o PPC

Neste capítulo são apresentadas novas arquiteturas de ESN especialmente projetadas para o PPC usando sinais de EEG. Diferente do capítulo anterior, no qual foi apresentado um modelo de ESN capaz de processar os registros dos eletrodos separadamente, este capítulo desenvolve modelos baseados em *reservoir* capazes de considerar todos os eletrodos, alterando as formas de processamento dos dados bem como das estruturas da rede *echo state*.

5.1 Visão Geral

Sendo uma rede recorrente, a proposta do *reservoir* é se aproveitar das características temporais do exame. Existem diferentes abordagens de como trabalhar o *reservoir* com o EEG, já que o exame não se trata de apenas uma série temporal e sim de um conjunto de séries capturadas por diferentes eletrodos.

O que considerar como um instante i no *reservoir* pode variar dependendo da forma que a entrada é inserida. Pode-se utilizar um pedaço do exame, ou seja, um intervalo de tempo como feito no capítulo anterior ou cada instante i pode ser o agrupamento de todos eletrodos em um tempo específico t . O padrão é o exame ser dividido em partes e inserido em sequência no *reservoir*. O que difere as arquiteturas são a forma de divisão dos dados e como lidar com os diferentes eletrodos.

A computação em reservatório por si só não rotula os objetos, o que a torna dependente de outra técnica. Os classificadores kNN, LSVM e GSVM mais se adaptaram ao *reservoir* e foram estatisticamente superiores como mostrado no capítulo anterior, então utilizaremos esses métodos. O *reservoir* deste capítulo também utilizou a estrutura do reservatório, proposta no capítulo 4. Ele foi projeto na forma de uma rede regular, a qual se mostrou estatisticamente superior à estrutura randômica normalmente empre-

gada. Outra vantagem dessa estrutura é a maior facilidade para ser implementado em sistemas embarcados.

Em todas estruturas onde é necessária uma ordem, os eletrodos seguem a sequência de captura do exame, pegando os eletrodos posicionados mais ao extremo exterior da cabeça seguindo para o centro, primeiro do lado esquerdo depois do lado direito. A sequência é: F7, T3, T5, Fp1, F3, C3, P3, O1, F8, T4, T6, Fp2, F4, C4, P4, O2, FZ, CZ, PZ, OZ. Ademais, no capítulo 4 foi mostrado testes analisando diferentes parâmetros, seleção de técnica e estrutura. Utilizamos as configurações com os melhores resultados, tanto no *reservoir* quanto nos algoritmos selecionados. Assim, o *reservoir* é criado com tamanho $r = 200$, e contém 10% das conexões ativas, o que significa que cada eletrodo tem 20 ligações, com os 10 vizinhos mais próximos de cada lado. Os valores do *reservoir* foram preenchidos aleatoriamente, mas seguindo uma distribuição normal, com desvio = 0.15 e média 0. Para os algoritmos, no KNN foi utilizado $k = 1$, no LSVM $C = 2^8$, e no GSVM $C = 2^8$ e $\gamma = 2^0$.

5.2 Descrição do Modelo

O *reservoir* utiliza uma matriz de entrada que é inserida por linha. Uma matriz T de tamanho (p, n) , faz com que o *reservoir* tenha um tamanho de entrada n , já que cada valor do vetor linha é a entrada em um ponto da ESN. Os dados são inseridos sequencialmente, por p passos. O *reservoir* tem um vetor de estado do tamanho do modelo $r = 200$ iniciado em 0 para cada paciente.

A saída do *reservoir* para cada instante de tempo i é o vetor de estado após a inserção de cada entrada. A saída do final é a matriz formada depois que as p linhas são inseridas no *reservoir*, formando a saída de tamanho (p, r) . A saída do *reservoir* é usada para alimentar os algoritmos de classificação.

5.2.1 MS-ESN-1

A arquitetura multi-sequência MS-ESN-1 pode ser definida pelo processamento conjunto dos registros de todos os eletrodos em um único *reservoir*. Considerando os registros elétricos capturados pelos eletrodos em um período de tempo n , dado por $T_{20 \times n}$, onde cada linha representa um eletrodo $e \in \{1, 2, \dots, 20\}$:

$$T = \begin{bmatrix} T_{11} & T_{12} & T_{13} & \dots & T_{1n} \\ T_{21} & T_{22} & T_{23} & \dots & T_{2n} \\ T_{31} & T_{32} & T_{33} & \dots & T_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ T_{20,1} & T_{20,2} & T_{20,3} & \dots & T_{20,n} \end{bmatrix} .$$

Os dados em T são processados pelo *reservoir* da seguinte forma. Em cada instante de tempo $i \in \{1, 2, \dots, n\}$, uma sequência T_i^\top , a qual representa o registro dos vinte eletrodos em um dado instante de tempo, é processada pelo *reservoir* resultando em um vetor de saída de tamanho r . Ao término do processo, a matriz $R_{r \times n}$ abaixo é obtida. Essa matriz é então achatada para um vetor unidimensional de tamanho $r * n$, o qual é então utilizado como entrada para os algoritmos de classificação na camada de readout.

$$R = \begin{bmatrix} R_{11} & R_{12} & R_{13} & \dots & R_{1n} \\ R_{21} & R_{22} & R_{23} & \dots & R_{2n} \\ R_{31} & R_{32} & R_{33} & \dots & R_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ R_{r1} & R_{r2} & R_{r3} & \dots & R_{rn} \end{bmatrix}$$

A motivação por trás da arquitetura MS-ESN-1 é permitir uma transformação não linear dos registros capturados por todos os eletrodos em cada instante de tempo através de um *reservoir*, o qual atua como um mapa único para levar esses dados de entrada para um novo espaço. A Figura 20 exibe os passos de MS-ESN-1. Onde o exame é inserido com todos os eletrodos de uma vez no *reservoir*, uma saída é gerada e usada como entrada para os algoritmos de classificação, que realizam a predição final do exame.

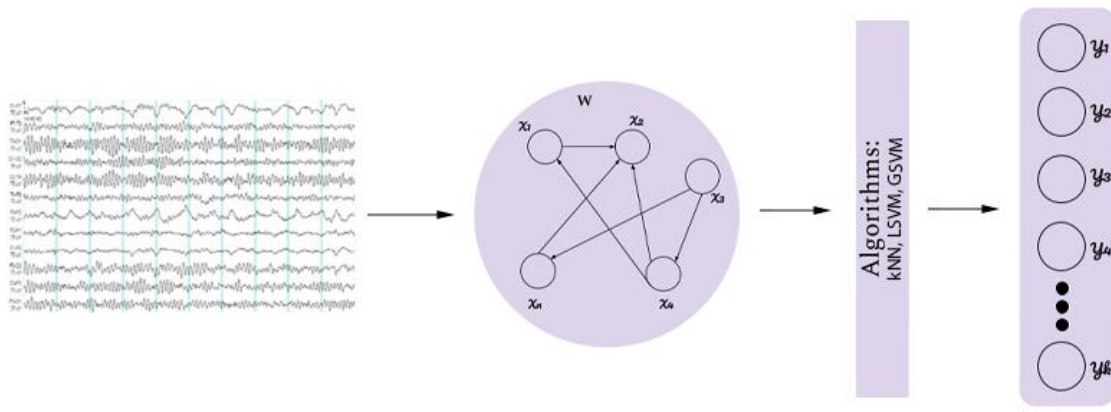


Figura 20 – Estrutura da rede MS-ESN-1, onde os dados são colocados em sequência no *Reservoir*, e utilizado nos algoritmos de classificação.

5.2.2 MS-ESN-2

A arquitetura multi-sequência MS-ESN-2 pode ser definida pelo processamento individual dos registros de todos os eletrodos em um único *reservoir*. Considerando os registros elétricos capturados por um eletrodo $e \in \{1, 2, \dots, 20\}$ em um período de tempo n , dado

por $T'_{n \times 1}^{(e)}$, uma função de adaptação $fa(T'^{(e)}) \rightarrow T^{(e)}$ é responsável por dividir uma série $T'^{(e)}$ em p partes de tamanho $m = n/p$, resultando na matriz $T_{p \times m}^{(e)}$ abaixo.

$$T^{(e)} = \begin{bmatrix} T_{11}^{(e)} & T_{12}^{(e)} & T_{13}^{(e)} & \dots & T_{1m}^{(e)} \\ T_{21}^{(e)} & T_{22}^{(e)} & T_{23}^{(e)} & \dots & T_{2m}^{(e)} \\ T_{31}^{(e)} & T_{32}^{(e)} & T_{33}^{(e)} & \dots & T_{3m}^{(e)} \\ \dots & \dots & \dots & \dots & \dots \\ T_{p1}^{(e)} & T_{p2}^{(e)} & T_{p3}^{(e)} & \dots & T_{pm}^{(e)} \end{bmatrix}$$

Para cada eletrodo e , uma sequência $T_i^{(e)}$, com $i = \{1, 2, \dots, p\}$, é processada pelo *reservoir* resultando em uma nova matriz $R_{p \times r}^{(e)}$, conforme exibido abaixo. A motivação por trás dessa arquitetura é permitir uma transformação não linear de períodos de sequência do exame através de um mesmo *reservoir*, o qual atua como um mapa único para levar os dados de entrada processados separadamente para cada eletrodo para um novo espaço.

$$R^{(e)} = \begin{bmatrix} R_{11}^{(e)} & R_{12}^{(e)} & R_{13}^{(e)} & \dots & R_{1r}^{(e)} \\ R_{21}^{(e)} & R_{22}^{(e)} & R_{23}^{(e)} & \dots & R_{2r}^{(e)} \\ R_{31}^{(e)} & R_{32}^{(e)} & R_{33}^{(e)} & \dots & R_{3r}^{(e)} \\ \dots & \dots & \dots & \dots & \dots \\ R_{p1}^{(e)} & R_{p2}^{(e)} & R_{p3}^{(e)} & \dots & R_{pr}^{(e)} \end{bmatrix}$$

Depois de processar todos os eletrodos, cada matriz $R^{(e)}$ é achatada para um vetor unidimensional $R'^{(e)}$ de tamanho $p * m$, os quais são em seguida, concatenados, i.e., $[R'^{(1)} R'^{(2)} \dots R'^{(20)}]$, e utilizados como entrada para os algoritmos de classificação na camada de readout. A Figura 21 traz uma visão geral por trás da arquitetura MS-ESN-2, onde o exame é separado por eletrodo, cada eletrodo é dividido em pedaços e inserido no mesmo *reservoir*. Cada um tem uma saída gerada que é concatenada e inserida nos algoritmos de classificação, que definem o rótulo para o exame.

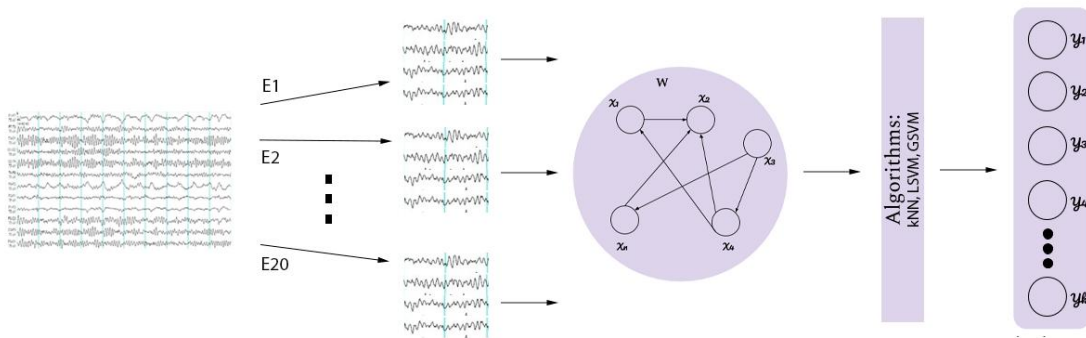


Figura 21 – Estrutura da rede MS-ESN-2, o sinal de cada eletrodo é quebrado em pedaços separadamente e utilizado no *Reservoir*, depois é agrupado e usado pelos algoritmos de classificação.

5.2.3 MS-ESN-3

A arquitetura multi-sequência MS-ESN-3 pode ser definida pelo processamento conjunto de períodos de tempo dos registros de todos os eletrodos em um único *reservoir*. Considerando os registros elétricos capturados por um eletrodo $e \in \{1, 2, \dots, 20\}$ em um período de tempo n , dado por $T_{n \times 1}^{(e)}$, uma função de adaptação $fa(T^{(e)}) \rightarrow T^{(e)}$ é responsável por dividir uma série $T^{(e)}$ em p partes de tamanho $m = n/p$, resultando na matriz $T_{p \times m}^{(e)}$ abaixo.

$$T^{(e)} = \begin{bmatrix} T_{11}^{(e)} & T_{12}^{(e)} & T_{13}^{(e)} & \dots & T_{1m}^{(e)} \\ T_{21}^{(e)} & T_{22}^{(e)} & T_{23}^{(e)} & \dots & T_{2m}^{(e)} \\ T_{31}^{(e)} & T_{32}^{(e)} & T_{33}^{(e)} & \dots & T_{3m}^{(e)} \\ \dots & \dots & \dots & \dots & \dots \\ T_{p1}^{(e)} & T_{p2}^{(e)} & T_{p3}^{(e)} & \dots & T_{pm}^{(e)} \end{bmatrix}$$

Antes de serem processados pelo *reservoir*, os dados registrados em cada eletrodo, denotados por $T^{(e)}$, são concatenados por linha (partes), de modo que $T^{(1:20)} = [T^{(1)}T^{(2)} \dots T^{(20)}]$. Em seguida, para cada parte do exame $i \in \{1, 2, \dots, p\}$, um conjunto de sequências $[T_{i1}^{(1)}, T_{i2}^{(1)}, \dots, T_{im}^{(1)}, T_{i1}^{(2)}, T_{i2}^{(2)}, \dots, T_{im}^{(2)}, \dots, T_{i1}^{(20)}, T_{i2}^{(20)}, \dots, T_{im}^{(20)}]$, o qual representa a concatenação dos registros dos vinte eletrodos em um dado período de tempo m , é processada pelo *reservoir* resultando em um vetor de saída de tamanho r . Ao término do processo, a matriz $R_{p \times r}$ abaixo é obtida. Essa matriz é então achatada para um vetor unidimensional de tamanho $p*r$, o qual é então utilizado como entrada para os algoritmos de classificação na camada de readout.

$$R = \begin{bmatrix} R_{11} & R_{12} & R_{13} & \dots & R_{1r} \\ R_{21} & R_{22} & R_{23} & \dots & R_{2r} \\ R_{31} & R_{32} & R_{33} & \dots & R_{3r} \\ \dots & \dots & \dots & \dots & \dots \\ R_{p1} & R_{p2} & R_{p3} & \dots & R_{pr} \end{bmatrix}$$

A motivação por trás da arquitetura MS-ESN-3 é permitir uma transformação não linear dos registros capturados por todos os eletrodos considerando cada parte (divisão) do exame através de um *reservoir*, o qual atua como um mapa único para levar essa concatenação das divisões dos dados de entrada para um novo espaço.

A Figura 22 exibe os passos da MS-ESN-3, os exames são divididos por eletrodo, cada série temporal de cada eletrodo é separado em pedaços formando uma matriz. Os pedaços são concatenados por linha, formando uma nova matriz, onde cada linha contém os dados da linha equivalente de todos os eletrodos. A matriz gerada é passada pelo *reservoir*, que tem sua saída transformada em um vetor e inserida nos algoritmos de classificação.

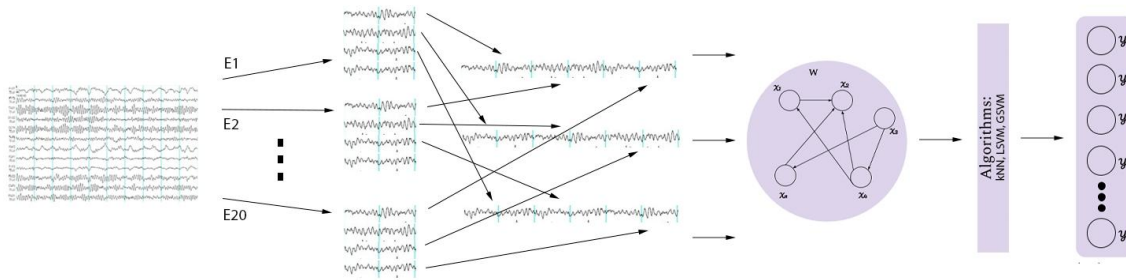


Figura 22 – Estrutura da rede MS-ESN-3, a serie de cada eletrodo é dividida em pedaços, os pedaços são agrupados de acordo com seu índice, na ordem dos eletrodos, em seguida passam pelo *Reservoir* e são classificados pelos algoritmos.

5.2.4 MS-ESN-4

A arquitetura multi-sequência MS-ESN-4 pode ser definida pelo processamento individual dos registros de todos os eletrodos em seus respectivos *reservoirs*. MS-ESN-4 é baseado na arquitetura MS-ESN-2, exceto porque aqui são considerados múltiplos *reservoirs*, de modo que os registros de cada eletrodo são processados no seu *reservoir* correspondente. A motivação por trás dessa arquitetura é permitir uma transformação não linear de períodos de sequência do exame através de múltiplos *reservoirs*, os quais atuam como um mapa específico para o seu eletrodo correspondente, de modo a levar os dados de entrada processados separadamente para cada eletrodo para novos espaços.

Depois de processar todos os eletrodos em seus *reservoirs* correspondentes, cada matriz $R^{(e)}$ é achatada para um vetor unidimensional $R^{(e)}$ de tamanho $p * m$, e em seguida, concatenados com aquelas obtidas pelos outros *reservoirs*, i.e., $[R^{(1)} R^{(2)} \dots R^{(20)}]$.

A Figura 23 traz uma visão geral por trás da arquitetura MS-ESN-4, que assim como a MS-ESN-2, tem seu exame separado pelos eletrodos, a serie de cada um é quebrada em pedaços, mas cada um tem seu próprio *reservoir* e sofre sua própria transformação. As saídas são agrupadas e transformadas em um grande vetor, que é inserido nos algoritmos de classificação gerando a saída y .

5.2.5 MS-DeepESN

Outra estrutura vista na literatura, é o deep echo state network (deepESN) (GALLICCHIO; MICHELI, 2017), onde várias redes da computação em reservatórios são utilizadas em sequência, e cada uma delas recebe de entrada a saída da rede anterior.

Com um comportamento semelhante ao modelo deepESN, a MS-DeepESN, pode ser definida pelo processamento dos registros dos eletrodos em sequência por diferentes *reservoirs*.

Considerando os registros elétricos capturados por um eletrodo $e \in \{1, 2, \dots, 20\}$ em um período de tempo n , dado por $T^{(e)n \times 1}$, uma função de adaptação $fa(T^{(e)} \rightarrow T^{(e)})$ é

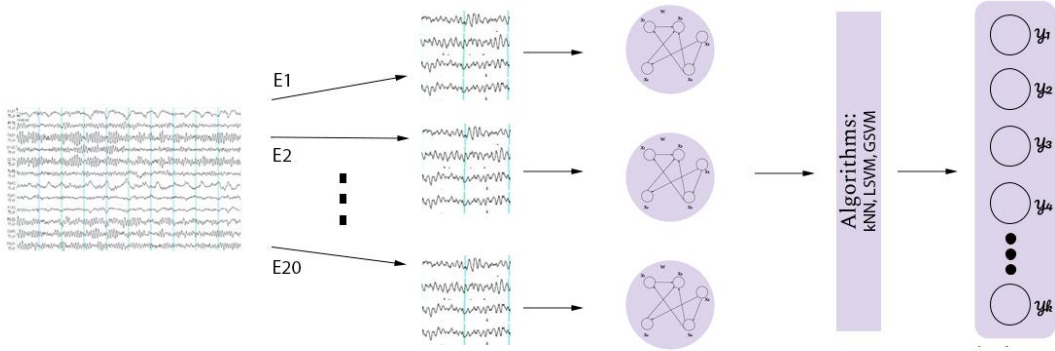


Figura 23 – Estrutura da rede MS-ESN-4, cada eletrodo é dividido em pedaços que em sequência é transformado cada um pelo seu próprio *Reservoir*, as saídas são agrupadas e utilizadas nos classificadores.

responsável por dividir uma serie $T^{(e)}$ em p partes de tamanho $m = n/p$, resultando em uma matriz $T^{(e)p \times m}$ abaixo.

$$T^{(1)} = \begin{bmatrix} T_{11}^{(1)} & T_{12}^{(1)} & T_{13}^{(1)} & \dots & T_{1m}^{(1)} \\ T_{21}^{(1)} & T_{22}^{(1)} & T_{23}^{(1)} & \dots & T_{2m}^{(1)} \\ T_{31}^{(1)} & T_{32}^{(1)} & T_{33}^{(1)} & \dots & T_{3m}^{(1)} \\ \dots & \dots & \dots & \dots & \dots \\ T_{p1}^{(1)} & T_{p2}^{(1)} & T_{p3}^{(1)} & \dots & T_{pm}^{(1)} \end{bmatrix}$$

A sequência é definida por $S = \{F7, T3, T5, Fp1, F3, C3, P3, O1, F8, T4, T6, Fp2, F4, C4, P4, O2, FZ, CZ, PZ, OZ\}$. Para o primeiro eletrodo, definido por uma sequência $T^{(1)i}$, com $i = 1, 2, \dots, p$ é processada pelo primeiro *reservoir* resultando em uma nova matriz $R^{(1)p \times r}$, conforme exibido abaixo.

$$R^{(1)} = \begin{bmatrix} R_{11}^{(1)} & R_{12}^{(1)} & R_{13}^{(1)} & \dots & R_{1r}^{(1)} \\ R_{21}^{(1)} & R_{22}^{(1)} & R_{23}^{(1)} & \dots & R_{2r}^{(1)} \\ R_{31}^{(1)} & R_{32}^{(1)} & R_{33}^{(1)} & \dots & R_{3r}^{(1)} \\ \dots & \dots & \dots & \dots & \dots \\ R_{p1}^{(1)} & R_{p2}^{(1)} & R_{p3}^{(1)} & \dots & R_{pr}^{(1)} \end{bmatrix}$$

Uma nova matriz $F^{(x)}$ é criada com $R^{(1)}$ sendo concatenado por linha (partes) com a matriz $T^{(e)}$ do segundo eletrodo e , i.e. $F^{(1)} = [R^{(1)}, T^{(2)}]$ é definida abaixo.

$$F^{(1)} = \begin{bmatrix} F_{11}^{(1)} & F_{12}^{(1)} & F_{13}^{(1)} & \dots & F_{1r}^{(1)} & T_{11}^{(1)} & T_{12}^{(1)} & T_{13}^{(1)} & \dots & T_{1m}^{(1)} \\ F_{21}^{(1)} & F_{22}^{(1)} & F_{23}^{(1)} & \dots & F_{2r}^{(1)} & T_{21}^{(1)} & T_{22}^{(1)} & T_{23}^{(1)} & \dots & T_{2m}^{(1)} \\ F_{31}^{(1)} & F_{32}^{(1)} & F_{33}^{(1)} & \dots & F_{3r}^{(1)} & T_{31}^{(1)} & T_{32}^{(1)} & T_{33}^{(1)} & \dots & T_{3m}^{(1)} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ F_{p1}^{(1)} & F_{p2}^{(1)} & F_{p3}^{(1)} & \dots & F_{pr}^{(1)} & T_{p1}^{(1)} & T_{p2}^{(1)} & T_{p3}^{(1)} & \dots & T_{pm}^{(1)} \end{bmatrix}$$

A matriz $F^{(1)}$ alimenta um novo *reservoir*, gerando uma nova matriz $R^{(2)}$. Esse passo de se concatenar a saída do *reservoir* com o eletrodo, acontece por todos os eletrodos da sequência S. De forma geral a equação de se concatenar é definida por $F^{(i)} = [R^{(i)}, T^{(i+1)}]$ para $i \in 1, 2, \dots, 19$. A saída do último *reservoir* $R^{(20)}$ é achatada para um vetor unidimensional $p * r$, o qual é então utilizado como entrada para os algoritmos de classificação na camada de readout.

A Figura 24 mostra como o exame é separado por eletrodo e cada série temporal de cada eletrodo é dividida em p pedaços. As matrizes são inseridas na sequência definida dos eletrodos de forma que a saída do *reservoir* utilizado pelo eletrodo anterior é concatenada com a matriz do eletrodo atual, e inserido em um novo *reservoir*. Esse passo é repetido até que todos eletrodos sejam utilizados. O *reservoir* final tem sua saída transformada em um vetor unidimensional e inserido nos algoritmos de classificação.

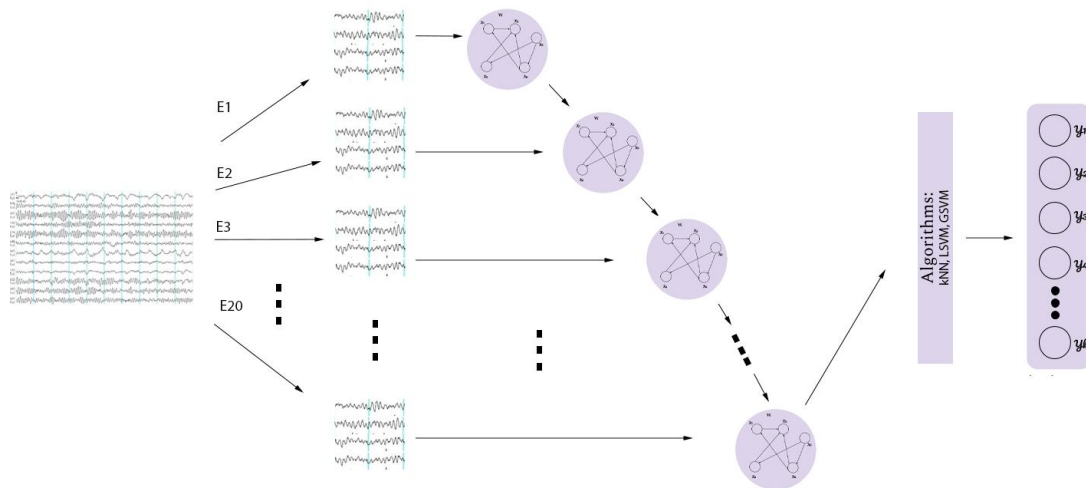


Figura 24 – Estrutura da rede MS-DeepESN, os eletrodos divididos em pedaços são transformados em cascata por diferentes reservatórios, cada reservatório recebe a saída do reservatório anterior agrupada com o novo eletrodo, a saída do último reservatório é utilizada pelos classificadores.

5.3 Resultados PPC para Dois Desfechos

Para cada arquitetura MS-ESN, a Tabela 15 apresenta o melhor resultado para cada algoritmo considerando os classificadores KNN, LSVM e GSVM, para o PPC com dois desfechos. Nas simulações o número de divisões (Div.) de cada exame é selecionado entre $\{10, 20, 40\}$, e os valores $\{10, 40\}$ são os parâmetros utilizados no melhor resultado de cada estrutura.

O GSVM novamente atingiu melhor acurácia com pior F1-macro, mas continua classificando todos os pacientes em um único desfecho, mostrando que transformação não-linear pode não ser ideal para os dados do *reservoir*, o LSVM que utiliza kernel linear consegue

três dos cinco melhores resultados, mostrando que o *Reservoir* cumpre com seu papel de transformar o espaço inicial em um espaço linearmente separável. As outras duas arquiteturas tem como melhor resultado a execução com o kNN, o que significa que os dados transformados de mesmo desfecho ficaram próximos no espaço euclidiano.

As arquiteturas MS-ESN-1 e MS-ESN-3 tiveram os piores resultados, elas também empregaram diferentes representações para serem usadas no *reservoir*. A MS-ESN-1 utiliza cada instante i do exame como uma entrada no *reservoir*, fazendo com que o exame necessite de muita memória temporal, uma vez que o exame tem 400 pontos, diferente das outras arquiteturas onde a dimensão temporal é o número de pedaços m que as séries temporais são divididas. A MS-ESN-1 com *reservoir* de tamanho r tem como saída uma matriz que é achatada em um vetor de tamanho $400 * r$. Assim, a alta dimensionalidade dos dados transformados pode incorrer em problemas já conhecidos pela maldição da dimensionalidade (KUO; SLOAN, 2005).

Já as outras arquiteturas, utilizam cada série temporal proveniente de um eletrodo individualmente em cada *reservoir*, com exceção da MS-DeepESN que junta as saídas do *reservoir* com os dados do eletrodo seguinte. Essa abordagem parece fazer mais sentido, uma vez que a divisão dos m pedaços mantém a sequência temporal entre eles, ao mesmo tempo que cada pedaço contendo um intervalo da série é transformada pelo *reservoir* em um espaço que pode ser separado com mais facilidade.

A MS-DeepESN obteve 0.5631 de F1-macro, com o melhor resultado entre as arquiteturas. Isso mostra que a deepESN oferece benefícios em relação a utilizar somente um *reservoir* e também que simplesmente concatenar os dados para diferentes eletrodos pode gerar uma representação muito grande para os classificadores enquanto passar em cascata pelo eletrodos gera uma saída única que agrupa todo o espaço gerado por todos os eletrodos. O segundo melhor resultado pertence ao MS-ESN-2 com F1-macro 0.5192. Sua representação é uma das mais simples, em que cada eletrodo é transformado pelo *reservoir* individualmente e seus resultados são agrupados para classificação. Nenhum novo resultado obtido foi melhor que o benchmark usando quantificadores, o qual obteve F1-macro 0.5886.

As Tabelas 16 e 17 mostram o teste estatístico de Wilcoxon entre os algoritmos e entre o parâmetro número de divisões. As tabelas foram organizadas na forma “variável coluna símbolo variável linha”, em que símbolo $>$ indica que a variável coluna é estatisticamente superior, $<$ que a variável linha é estatisticamente superior, e \approx que o desempenho de ambas variáveis são equivalentes (falha para rejeitar hipótese nula).

No que diz respeito aos algoritmos, o LSVM é superior estatisticamente, considerando um nível de significância de 0.05, ao KNN nas arquiteturas MS-ESN-1, MS-ESN-3 e MS-ESN-4, e superior ao GSVM nas arquiteturas MS-ESN-1, MS-ESN-2, MS-ESN-3 e MS-ESN-4.

Para o número de divisões, apesar de os valores 10 e 40 estarem nos melhores resultados

Tabela 15 – Acurácia, precision, recall F1-macro com a melhor configuração de cada método para os testes sem *reservoir* e para as 5 estruturas analisadas, com a base de dados dividida em 2 desfechos.

Modelo	Readout	Div.	Acc.	Prec.	Rec.	F1-macro
MLP	-	-	0.6361	0.6041	0.5957	0.5886
S/ <i>Reservoir</i>	KNN	-	0.6500	0.5188	0.5389	0.4664
	LSVM	-	0.5611	0.4495	0.4194	0.4015
	GSVM	-	0.6583	0.5000	0.3292	0.3958
OZ (eletrodo)	LSVM	10	0.6389	0.5268	0.6022	0.5026
MS-ESN-1	KNN	10	0.6583	0.5000	0.3292	0.3969
	LSVM	40	0.6250	0.4966	0.5003	<u>0.4492</u>
	GSVM	10	0.6583	0.5000	0.3292	0.3969
MS-ESN-2	KNN	10	0.5889	0.5283	0.5269	<u>0.5192</u>
	LSVM	10	0.5806	0.4920	0.4818	0.4729
	GSVM	10	0.6583	0.5000	0.3292	0.3969
MS-ESN-3	KNN	20	0.6583	0.5021	0.3630	0.4040
	LSVM	40	0.6194	0.5004	0.5128	<u>0.4639</u>
	GSVM	10	0.6583	0.5000	0.3292	0.3969
MS-ESN-4	KNN	10	0.6583	0.5000	0.3292	0.3969
	LSVM	40	0.6417	0.5303	0.5425	<u>0.5027</u>
	GSVM	10	0.6583	0.5000	0.3292	0.3969
MS-DeepESN	KNN	10	0.6417	0.5691	0.5779	0.5631
	LSVM	10	0.6194	0.5423	0.5547	0.5374
	GSVM	10	0.6583	0.5000	0.3292	0.3969

para cada arquitetura, todos os valores(10, 20 e 40) são estatisticamente equivalentes considerando o nível de significância de 0.05.

5.4 Resultados PPC para Três Desfechos

A Tabela 18 apresenta os resultados das MS-ESN para o PPC com três desfechos. Os parâmetros foram os mesmos considerados na simulação anterior, e novamente só os valores {10, 40} aparecem como melhor resultado de cada estrutura. O comportamento das estruturas e dos algoritmos seguem o mesmo padrão que para dois desfechos. MS-ESN-2 e MS-DeepESN utilizaram o KNN e tiveram os dois melhores resultados entre as novas arquiteturas, com F1-macro 0.4123 e 0.4121, respectivamente. Dessa vez, quatro das cinco arquiteturas ultrapassaram o benchmark baseado em quantificadores para três desfechos.

As Tabelas 19 e 20 mostram o teste estatístico de Wilcoxon entre os algoritmos e entre o parâmetro número de divisões. O algoritmo LSVM é estatisticamente superior

Tabela 16 – P-values do teste Wilcoxon considerando os três classificadores readout avaliados para dois desfechos. Considerando um nível de confiança de 95% ($\alpha = 0.05$), os símbolos $>$, $<$ e \approx indicam que a variável da coluna é respectivamente melhor, pior ou equivalente à variável da linha.

Modelo	Readout	KNN	LSVM
MS-ESN-1	LSVM	0.0312 ($<$)	-
	GSVM	0.3173 (\approx)	0.0312 ($>$)
MS-ESN-2	LSVM	0.6875 (\approx)	-
	GSVM	0.1088 (\approx)	0.0312 ($>$)
MS-ESN-3	LSVM	0.0312 ($<$)	-
	GSVM	0.3173 (\approx)	0.0312 ($>$)
MS-ESN-4	LSVM	0.0312 ($<$)	-
	GSVM	1.0000 (\approx)	0.0312 ($>$)
MS-DeepESN	LSVM	0.2188 (\approx)	-
	GSVM	0.0625 (\approx)	0.1562 (\approx)

Tabela 17 – P-values do teste Wilcoxon considerando os diferentes valores de divisões do exame para dois desfechos. Considerando um nível de confiança de 95% ($\alpha = 0.05$), os símbolos $>$, $<$ e \approx indicam que a variável da coluna é respectivamente melhor, pior ou equivalente à variável da linha.

Modelo	Div.	10	20
MS-ESN-1	20	0.5930 (\approx)	-
	40	0.6547 (\approx)	0.5930 (\approx)
MS-ESN-2	20	0.1088 (\approx)	-
	40	0.1088 (\approx)	0.2850 (\approx)
MS-ESN-3	20	0.2850 (\approx)	-
	40	0.6547 (\approx)	0.2850 (\approx)
MS-ESN-4	20	0.1797 (\approx)	-
	40	0.6547 (\approx)	0.6547 (\approx)
MS-DeepESN	20	0.4652 (\approx)	-
	40	0.1441 (\approx)	0.7150 (\approx)

Tabela 18 – Acurácia, precision, recall F1-macro com a melhor configuração de cada método para os testes sem *reservoir* e para as 5 estruturas analisadas, com a base de dados dividida em 3 desfechos.

Modelo	Readout	Div.	Acc.	Prec.	Rec.	F1-macro
MLP	-	-	0.4361	0.3607	0.3611	0.3505
S/ <i>Reservoir</i>	KNN	-	0.3583	0.4751	0.3960	0.3392
	LSVM	-	0.3889	0.2929	0.2730	0.2797
	GSVM	-	0.5333	0.3333	0.1778	0.2318
O1 (eletrodo)	KNN	20	0.4667	0.4003	0.3769	0.3505
MS-ESN-1	KNN	10	0.1917	0.3751	0.1817	0.1434
	LSVM	10	0.4861	0.3308	0.2826	<u>0.2956</u>
	GSVM	10	0.5333	0.3333	0.1778	0.2318
MS-ESN-2	KNN	10	0.4972	0.4187	0.4284	0.4123
	LSVM	10	0.4833	0.3510	0.3113	0.3264
	GSVM	10	0.5333	0.3333	0.1778	0.2318
MS-ESN-3	KNN	40	0.1806	0.3684	0.2291	0.1353
	LSVM	40	0.5306	0.3903	0.3990	<u>0.3696</u>
	GSVM	10	0.5333	0.3333	0.1778	0.2318
MS-ESN-4	KNN	40	0.1694	0.3611	0.1818	0.1238
	LSVM	40	0.5111	0.3819	0.3646	<u>0.3632</u>
	GSVM	10	0.5333	0.3333	0.1778	0.2318
MS-DeepESN	KNN	40	0.4722	0.4283	0.4119	<u>0.4121</u>
	LSVM	20	0.5083	0.3688	0.3281	0.3430
	GSVM	10	0.5333	0.3333	0.1778	0.2318

ao GSVM em todas as execuções considerando um nível de significância de 0.05. Nas arquiteturas MS-ESN-1, MS-ESN-3, MS-ESN-4, onde o KNN não foi o melhor algoritmo da arquitetura, ele foi estatisticamente pior que os outros dois algoritmos, o que faz com que ele varie de pior a melhor de acordo com o modelo. Por outro lado, o desempenho da LSVM é mais robusto à variação das arquiteturas. Em relação ao número de divisões, o único caso em que um número de divisão é estatisticamente superior ao outro, é 40 em relação a 10 para a arquitetura MS-DeepESN.

O número de divisões tem todas amostras como estatisticamente equivalentes para um $\beta = 0,5$, reduziremos as opções de divisão para 10 e 20 pedaços quando for necessária nos próximos testes.

Tabela 19 – P-values do teste Wilcoxon considerando os três classificadores readout avaliados para três desfechos. Considerando um nível de confiança de 95% ($\alpha = 0.05$), os símbolos $>$, $<$ e \approx indicam que a variável da coluna é respectivamente melhor, pior ou equivalente à variável da linha.

Modelo	Readout	KNN	LSVM
MS-ESN-1	LSVM	0.0312 ($<$)	-
	GSVM	0.0312 ($<$)	0.0312 ($>$)
MS-ESN-2	LSVM	0.4375 (\approx)	-
	GSVM	0.8438 (\approx)	0.0312 ($>$)
MS-ESN-3	LSVM	0.0312 ($<$)	-
	GSVM	0.0312 ($<$)	0.0312 ($>$)
MS-ESN-4	LSVM	0.0312 ($<$)	-
	GSVM	0.0312 ($<$)	0.0312 ($>$)
MS-DeepESN	LSVM	0.2076 (\approx)	-
	GSVM	0.0079 ($>$)	0.0087 ($>$)

Tabela 20 – P-values do teste Wilcoxon considerando os diferentes valores de divisões do exame para três desfechos. Considerando um nível de confiança de 95% ($\alpha = 0.05$), os símbolos $>$, $<$ e \approx indicam que a variável da coluna é respectivamente melhor, pior ou equivalente à variável da linha.

Modelo	Div.	10	20
MS-ESN-1	20	0.6547 (\approx)	-
	40	0.1797 (\approx)	0.1797 (\approx)
MS-ESN-2	20	0.1088 (\approx)	-
	40	0.1088 (\approx)	0.1088 (\approx)
MS-ESN-3	20	0.0679 (\approx)	-
	40	0.2733 (\approx)	0.7150 (\approx)
MS-ESN-4	20	0.4652 (\approx)	-
	40	0.0679 (\approx)	0.0679 (\approx)
MS-DeepESN	20	0.6051 (\approx)	-
	40	0.0052 ($<$)	0.2553 (\approx)

Redes neurais convolucionais para o PPC

Neste capítulo, nós estendemos a literatura do PPC a partir do desenvolvimento de uma representação espacial para o exame EEG considerando a resolução temporal dos sinais bem como a disposição dos eletrodos no escalpo. O capítulo também apresenta três arquiteturas de aprendizado baseadas em redes neurais convolucionais desenvolvidas para tirar proveito de tal representação.

6.1 Resolução Espacial dos Sinais EEG

Os modelos de rede convolucional utilizam uma representação espacial do exame EEG com uma estratégia semelhante à utilizada em (LI et al., 2017). Utilizando da geometria do sistema 10-20 e a disposição do exame EEG, organizam-se as séries temporais em um mapa o qual possui posições correspondentes às posições de cada eletrodo no escalpo. A Figura 25 apresenta um mapa de tamanho 5 o qual consiste em uma matriz de 5×5 com a posição dos eletrodos devidamente representadas na imagem. Em uma analogia simples, esse mapa representa uma foto do EEG considerando a resolução temporal capturada em cada instante de tempo. Outro mapa, esse de tamanho 7, é apresentado na Figura 26.

Além da representação em si, também foi desenvolvida uma estratégia de preenchimento dos espaços vazios do mapa, baseada em (LI et al., 2017). Tal estratégia, denominada vizinhança, preenche as posições sem valores do mapa considerando a média das suas vizinhanças horizontais e verticais, tal como descrito na equação a seguir.

$$V_{m,n} = \frac{V'_{m+1,n} + V'_{m-1,n} + V'_{m,n+1} + V'_{m,n-1}}{K}, (0 \leq m, n \leq 4, m, n \in N), \quad (23)$$

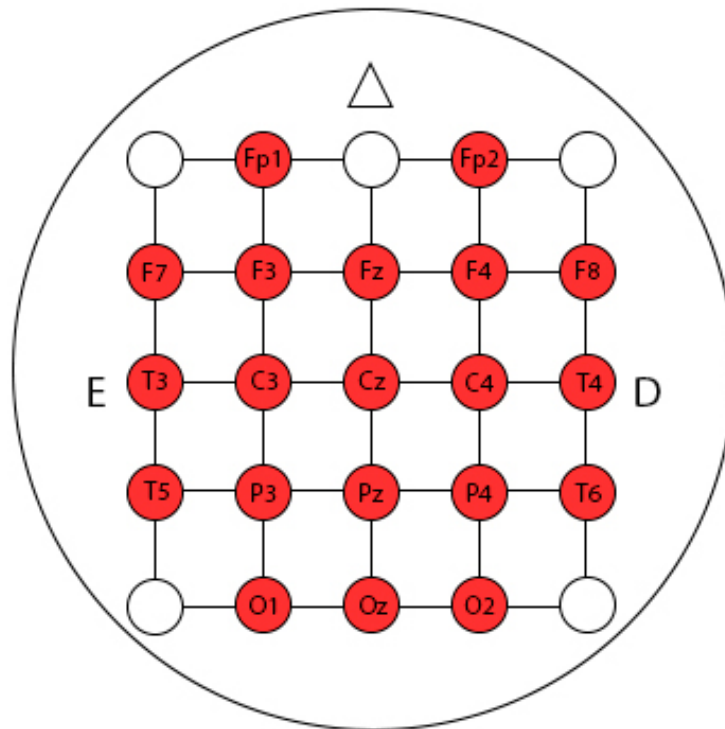


Figura 25 – Representação dos eletrodos na matriz 5x5

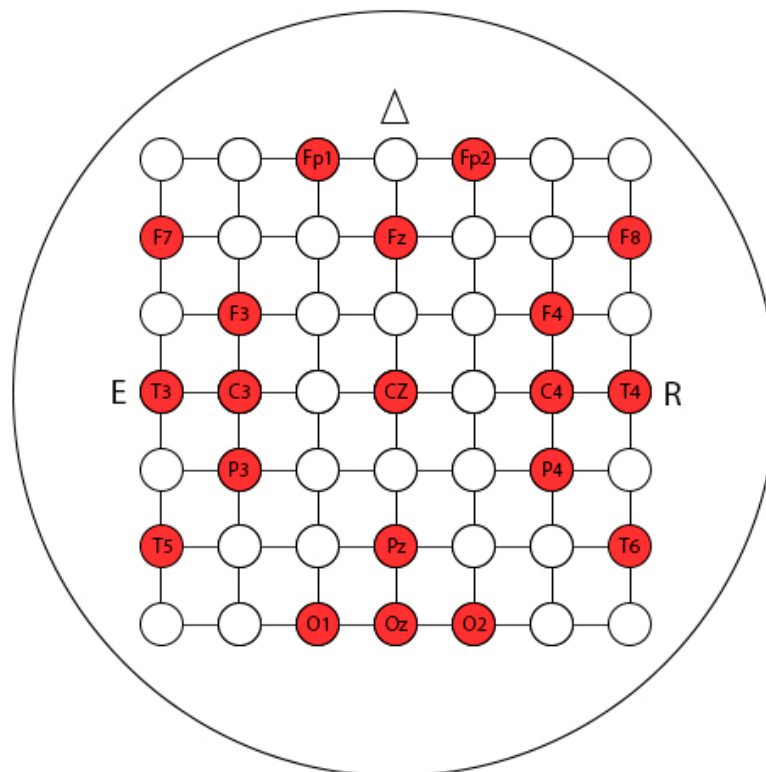


Figura 26 – Representação dos eletrodos na matriz 7x7

6.2 Arquiteturas de CNN para uma matriz de entrada

O modelo C1, exibido na Figura 27, tem três camadas convolucionais, a primeira contém 96 filtros de tamanho $= (3, 3, 10)$, com $\text{stride} = (1, 1, 5)$. As duas camadas seguintes tem 128 e 64 filtros respectivamente, ambos de tamanho $= (3, 3, 3)$. Após cada camada convolucional existe uma camada maxpooling com filtro de tamanho $(1, 1, 3)$ e $\text{stride} = (1, 1, 2)$. Todas as camadas só reduzem a dimensão no aspecto temporal, mantendo o tamanho m na lateral e altura da matriz, m varia entre os tamanhos 5 e 7. Além disso, existe uma sobreposição entre os filtros adjacentes, isso é consequência do tamanho do filtro no maxpooling ser maior que o tamanho do stride. Após a terceira camada convolucional, existem duas camadas densamente conectadas de tamanho 512 com dropout de 0.1, i.e. durante o treinamento 10% das conexões são desativadas aleatoriamente, a intenção é evitar overfitting. No momento de teste todas as conexões são ativadas e multiplicadas por 0.9. Após as camadas densas, existe a softmax com número de neurônios de acordo com o número de classes.

O modelo C2, exibido na Figura 28 também tem três camadas convolucionais, com 96, 128 e 32 filtros de tamanhos $= ((3, 3, 10), (3, 3, 3), (3, 3, 3))$ respectivamente. Existe duas camadas maxpooling após as duas primeiras camadas convolucionais, a primeira com tamanho $= (1, 1, 2)$ e $\text{stride} = (3, 3, 3)$ e a segunda com tamanho $= (3, 3, 3)$ e $\text{stride} = (2, 2, 2)$, após as camadas convolucionais existem 2 camadas densamente conectadas de tamanho 1024 e dropout também de 0.1 seguidas da camada Softmax com tamanho igual ao número de classes. Dessa vez a maxpooling causa uma redução do tamanho m do mapa na matriz C , causada pelo stride de tamanho $= (2, 2, 2)$ Em todas as camadas é utilizado a função de ativação reLU.

6.3 Arquitetura de CNN para múltiplas matrizes de entrada

A arquitetura mostrada na Figura 29 foi projetada especialmente para o modelo CNN-Multi, uma vez que ele apresenta três estruturas de entrada no lugar de uma: exame original, as transformações multi-escala e as transformações multi-frequência.

Para cada matriz C , duas variações são criadas: C^k representa a matriz criada utilizando a equação 14 aplicada em cada eletrodo representado em C ; e C^l representa a transformação feita pela equação 15 aplicada em cada eletrodo de C . Cada uma das entradas C , C^k e C^l para diferentes valores de k e l , são aplicadas em uma camada convolucional inicial de 96 filtros de tamanho $= (3, 3, 15)$ seguida de uma camada maxpooling. A quantidade de matrizes de entrada criadas é igual a E

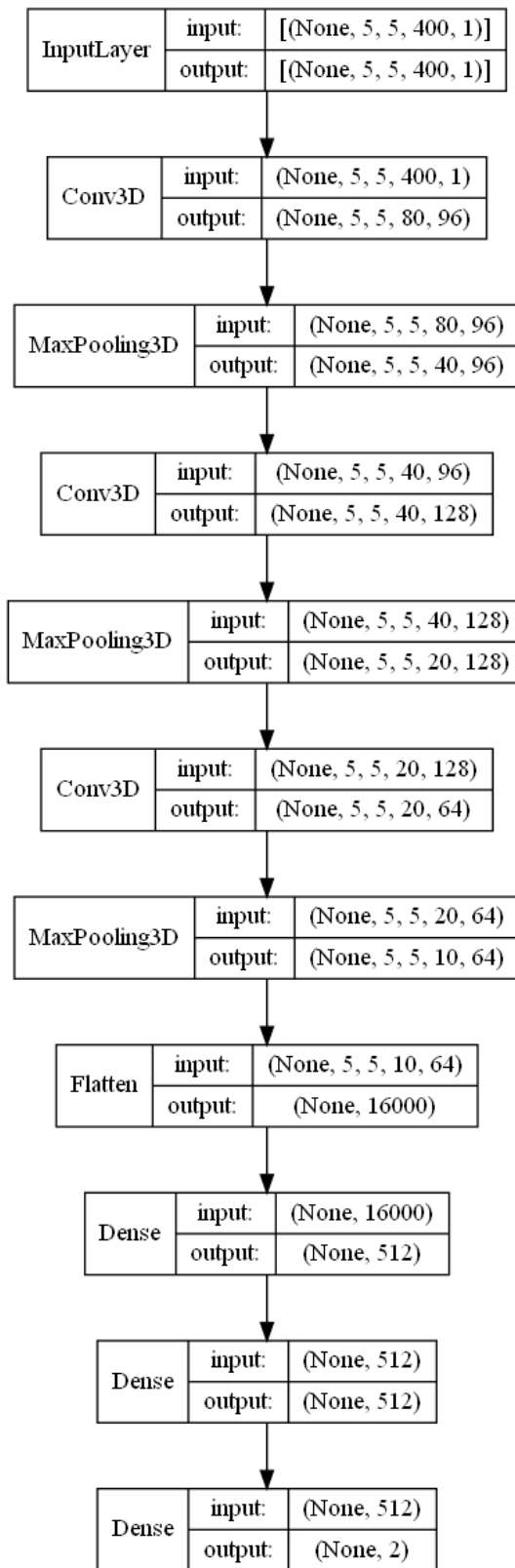


Figura 27 – Modelo de rede convolucional C1 utilizada pelos CNN-Padrão e CNN-Concat

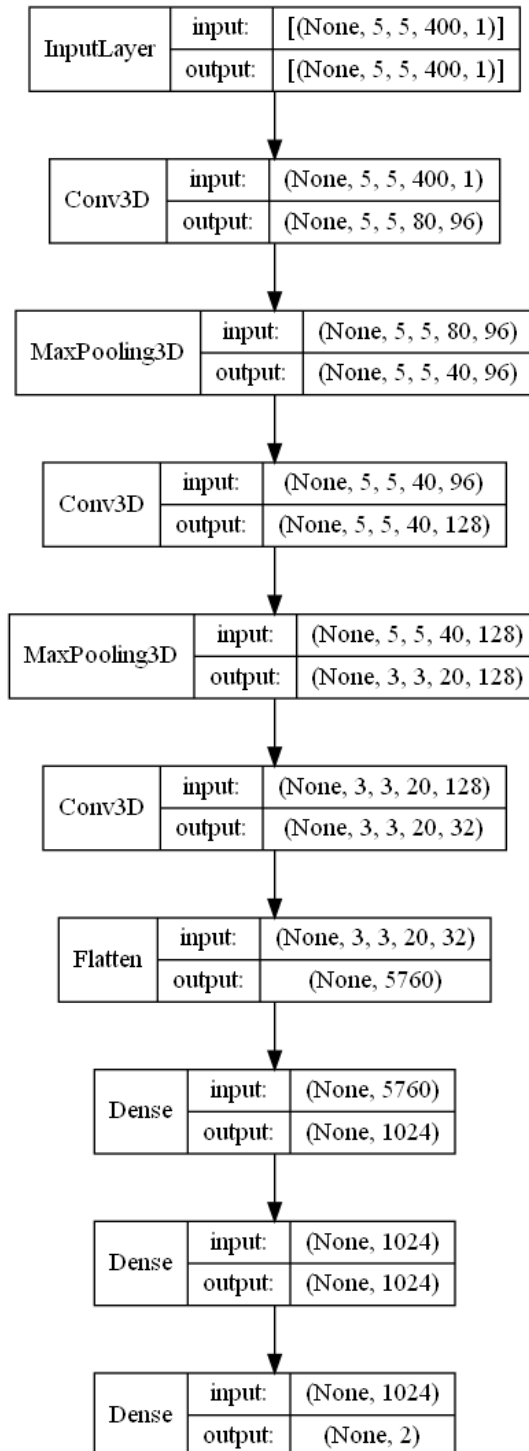


Figura 28 – Modelo de rede convolucional C2 utilizada pelos CNN-Padrão e CNN-Concat

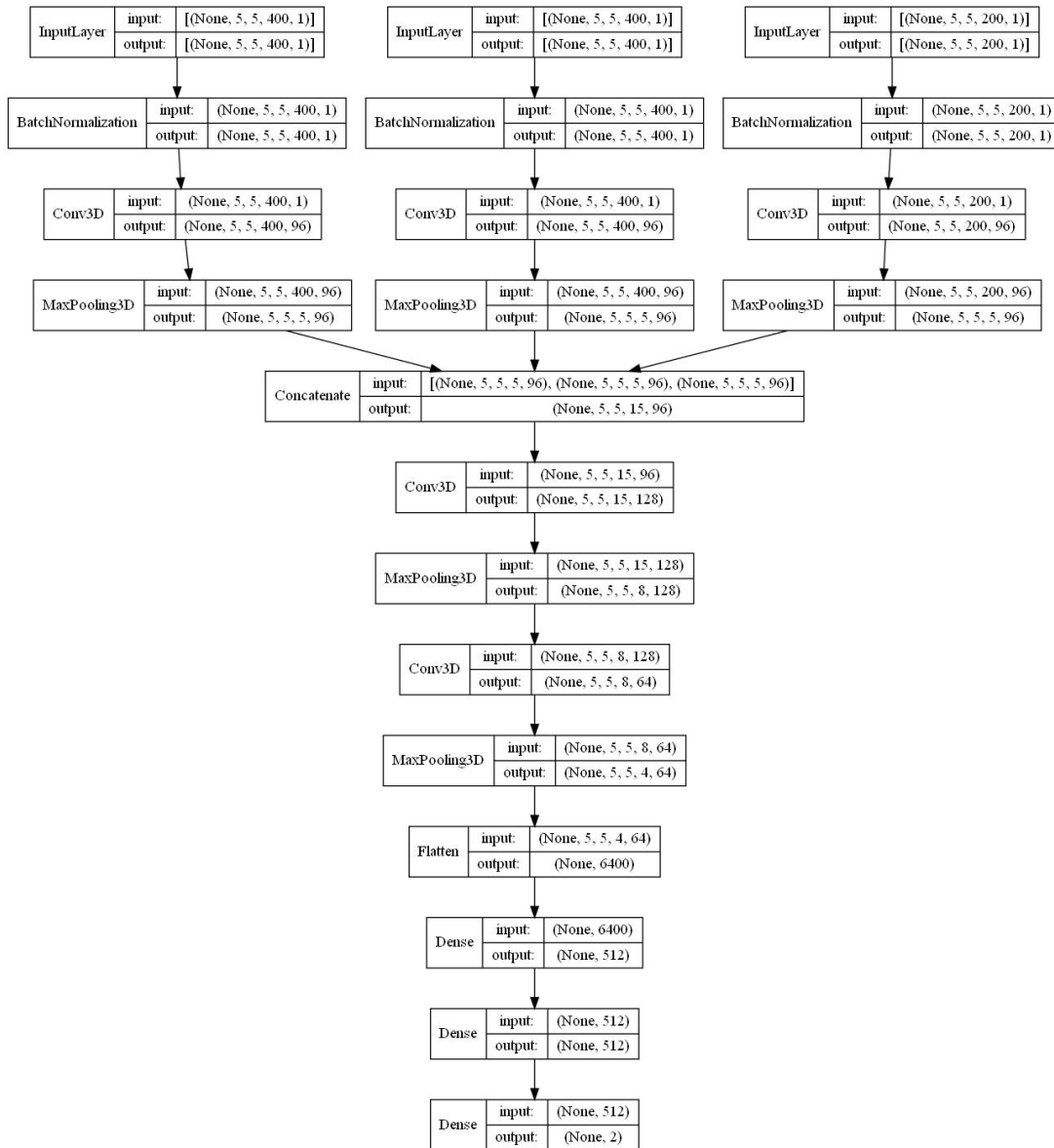


Figura 29 – Modelo de rede convolucional utilizada pelo CNN-Multi.

Pela matriz gerada pela transformação multi-escala sofrer redução de tamanho na dimensão temporal, e por serem geradas muitas matrizes de entrada, a camada de maxpooling tem o número de filtros e stride de tamanho variado para cada entrada, de forma que a saída de todas as camadas maxpooling tenham tamanhos iguais a 5 na terceira dimensão, o que diminui o número de parâmetros na rede e padroniza os tamanhos de saída após a primeira camada. Então se uma matriz tem o tamanho 400, a maxpooling terá tamanho= (1, 1, 80) e stride= (1, 1, 80), com tamanho 200, a maxpooling terá tamanho= (1, 1, 40) e stride= (1, 1, 40). No final teremos varias matrizes de dimensão $(m \times m \times 5)$ uma para cada matriz de entrada.

Todas saídas da maxpooling são concatenadas na terceira dimensão, em uma matriz Co de dimensão $(m \times m \times 5 * E)$. A continuação da rede utiliza Co e em mais duas camadas convolucionais com 128 e 64 filtros respectivamente as duas de tamanho (3,3,3)

ambas seguidas de uma maxpooling com tamanho = (1, 1, 2) e stride = (1, 1, 2). O último maxpooling é seguido de duas camadas densamente conectadas e uma softmax com número de neurônios iguais ao de classe.

6.4 Descrição dos modelos de CNN

Para ser utilizado pela CNN o exame EEG foi mapeado para uma matriz de 3 dimensões definidas como $C_{m \times m \times n}$, sendo m o tamanho da matriz criada, também chamada de mapa, e n o tamanho do exame. Conforme apresentado na seção anterior, a matriz pode ter tamanho 5 ou 7 e adota uma estratégia de vizinhança. Para cada instante i na matriz C temos uma imagem representando o estado do cérebro, definida pelos valores registrados pelos eletrodos.

Para a tarefa de classificação, três modelos de redes foram desenvolvidos. A diferença dos modelos está nas entradas das redes. O primeiro modelo é uma rede convolucional comum (CNN-Padrão). O CNN-Concat, divide a matriz de entrada em pedaços, onde cada pedaço é utilizado separadamente na convolucional com propósito de extração de atributos os quais são posteriormente utilizados por classificadores. O terceiro modelo é o CNN-Multi, ele utiliza um modelo de rede que considera múltiplas matrizes de entrada.

6.4.1 CNN-Padrão

A CNN-padrão, transforma as entradas na representação 3D do exame a partir das arquiteturas C1 ou C2 sumarizadas nas Figuras 27 e 28, respectivamente. O modelo exibido nas duas figuras dizem respeito ao utilizado para o PPC com dois desfechos. No caso do PPC com três desfechos, a última camada softmax terá três neurônios.

6.4.2 CNN-Concat

O CNN-Concat reparte a matriz $C_{m \times m \times n}$ de cada exame em p pedaços, de forma que passa a existir p matrizes de tamanho $C_{m \times m \times n/p}^{(i)}$. A CNN nesse modelo não funciona para classificação, mas sim para realizar uma transformação dos dados. É possível treinar os modelos de rede, e ao invés de pegar a saída da camada softmax, pega-se os valores da última camada convolucional, tirando proveito da extração automática de atributos providas por tais camadas. A ideia com este modelo é transformar as matrizes $C_{m \times m \times n/p}^{(i)}$ em vetores de atributos. Os valores resultantes são então utilizados em classificadores readout, em especial o KNN, LSVM e GSVM.

Todos os pedaços $C_{m \times m \times n/p}^{(i)}$ de todos os exames do conjunto de treino são utilizados para treinar a CNN. Cada pedaço recebe como rótulo o resultado do exame completo durante a fase de treinamento. E cada pedaço gera um vetor $T^{(i)} = t1, t2, \dots, tx$, x é a quantidade de atributos na última camada convolucional.

A transformação de todos os pedaços por exame são agrupadas em um único vetor T^{total} que é utilizado para treinar o modelo de classificação. O KNN, o LSVM e o GSVM continuam com a mesma seleção de parâmetros. As arquiteturas utilizadas para esse modelo de CNN são C1 ou C2.

6.4.3 CNN-Multi

A CNN-Multi utiliza a arquitetura de múltiplas matrizes de entrada apresentada na seção anterior. Em comparação com os outros dois modelos, este é o mais complexo do ponto de vista estrutural. Sua inspiração vem do trabalho apresentado em 2.2.5.2, embora diferente do modelo de 1D considerado naquele estudo, adaptamos uma convolucional 3D própria para receber a representação de mapa. Assim, além da matriz C , duas outras transformações aplicadas sobre ela serve como entrada na CNN, a saber:

- Matriz multi-escala: diferentes escalas de tempo são produzidas a partir da série original, com um espaço de tamanho k entre os índices escolhidos. Dessa forma, para a matriz C definiremos a nova matriz como C^k um corte de todo o mapa de tamanho $m \times m$ a cada k pontos. O mapa final fica com tamanho $m \times m \times fracn - 1k$. Os valores de k utilizados são $\{2,3,4,5,6,7\}$.
- Matriz multi-frequência: diferentes problemas obtém diferentes resultados para diferentes frequências, como não é possível saber qual a melhor, a ideia é gerar varias séries temporais com diferentes suavizações, de forma que se tenha o máximo de informações. A técnica utilizada para suavizar é a da janela de média.

6.4.4 Heurística de atenção

Em cada exame o médico fez a seleção de diferentes trechos do EEG e 9 diferentes são analisados para uma decisão sobre o prognóstico. Os trechos selecionados nem sempre são referentes ao comportamento característico do real prognóstico do paciente, podendo ser escolhido um pedaço de exame que pela nossa representação seria classificado como vida, mas pertencente a um paciente com desfecho de morte. Além disso, os conjuntos têm classes desbalanceadas, ou seja, possuem quantidades diferentes de exames em cada grupo de desfecho, o que pode acabar influenciando (favorecendo) ao grupo de desfecho mais representativo.

Considerando que os resultados definidos para os pedaços são representados pelo vetor $Res = \{r_i\}$, $1 \leq i \leq 9, r_i \in \{0, 1\}$ para dois desfechos e $r_i \in \{0, 1, 2\}$ para três desfechos, o prognóstico final do paciente em todas as abordagens desenvolvidas até o presente momento é dado pela moda do vetor Res , ou seja, o valor r_i que mais aparece em Res . Para três desfechos a quantidade mínima de respostas iguais para uma classe sair vitoriosa varia, por exemplo, no vetor $Res = \{0, 0, 0, 1, 1, 1, 2, 2, 2, 2\}$ tem como

resultado final 2, com 4 exemplos enquanto $Res = \{1, 1, 1, 1, 1, 1, 2, 2, 2, 2\}$ os mesmo 4 casos não garantiram ao resultado da classe 2 neste outro exemplo. Contudo, para dois desfechos existe um valor fixo, qualquer grupo de desfecho que alcance mais de 5 resultados iguais receberão tal prognóstico. Especificamente para essa situação de dois desfechos, desenvolvemos uma heurística de atenção ha , a qual é responsável por ponderar a predição final do prognóstico considerando padrões específicos e menos frequentes nos exames. Neste caso, a predição do prognóstico é definida através da função ha , a qual considera um limiar menor do que a moda das predições para classificar os sinais EEG em um grupo de desfecho minoritário, tal como segue:

$$ha(Res) = \begin{cases} 0 & \text{se } Cont(Res, 0) \geq x \\ 1 & \text{caso contrário,} \end{cases} \quad (24)$$

com $2 \leq x \leq 4$. A função $Cont(vetor, x)$ retorna quantos elementos iguais a x existe no vetor de saída Res : 0 representa desfecho positivo (grupo minoritário) e 1 desfecho negativo.

6.5 Resultados PPC para Dois Desfechos

A melhor arquitetura do reservior MS-DeepESN foi testada com a heurística de atenção, a qual melhorou o modelo em 0.01 de F1-macro, alcançando 0.5747. Apesar da melhoria, tal arquitetura continuou com desempenho menor que o benchmark de quantificadores (MLP) que obteve F1-macro de 0.5886. A CNN-Padrão obteve o melhor F1-macro, mostrando que a representação espacial e a CNN convencional foram capazes de modelar o espaço de forma mais eficiente. Apesar da CNN-Multi e da CNN-Concat obterem resultados um pouco menores, ainda assim são superiores ao benchmark quando considerados com a heurística de atenção. CNN-Multi se torna ineficiente, visto que as transformações dos dados são necessárias, e um input maior gera uma rede convolucional maior. A CNN-Concat divide os dados o que resulta em um menor tamanho da rede convolucional, contrapondo a necessidade de testar diferentes números de divisões e qual é mais eficiente (KNN e LSVM visto que GSVM não é eficiente em nenhum caso).

A heurística de atenção melhorou o resultado de todos os modelos de CNN em pelo menos 0.02 de F1-macro. Sem essa estratégia, apenas a CNN-Padrão superaria o resultado da MLP. A CNN-Concat obteve a melhoria mais considerável com a heurística: 0.05 de F1-macro. Interessante notar que para esse modelo os parâmetros selecionados para número de divisões e uso da vizinhança foram alterados na presença da heurística, enquanto na CNN-Multi e CNN-Padrão os parâmetros selecionados com e sem a presença da heurística permaneceram os mesmos.

Em síntese, os testes que não utilizam a heurística de atenção têm como melhor resultado a CNN-Padrão com um F1-macro de 0.61 utilizando o modelo C1, o mapa de

tamanho 5 e sem uso da vizinhança. Com a heurística a mesma rede obtém F1-Macro de 0.6391.

Tabela 21 – Acurácia, precision, recall F1-macro com a melhor configuração de cada arquitetura anterior e para a melhor configuração dos testes com os modelos convolucionais, sem a heurística e com(*), com a base de dados dividida em dois desfechos.

Modelo (Readout)	Mapa	Viz.	Arq.	Div.	Acc.	Prec.	Rec.	F1-macro
MLP	-	-	-	-	0.6361	0.6041	0.5957	0.5886
MS-DeepESN (KNN)	-	-	-	10	0.6417	0.5691	0.5779	0.5631
MS-DeepESN (KNN)*	-	-	-	40	0.5944	0.5961	0.5928	<u>0.5747</u>
CNN-Multi	5	T	-	-	0.6417	0.5877	0.5944	0.5722
CNN-Multi*	5	T	-	-	0.6222	0.6240	0.6186	<u>0.5935</u>
CNN-Padrão	5	F	C1	-	0.6750	0.6174	0.6155	0.6102
CNN-Padrão*	5	F	C1	-	0.6722	0.6524	0.6420	0.6391
CNN-Concat. (LSVM)	7	T	C2	20	0.6222	0.5555	0.5726	0.5504
CNN-Concat. (LSVM)*	7	F	C2	10	0.6194	0.6335	0.6222	<u>0.6062</u>

A Tabela 22 apresenta o resultado do teste de Wilcoxon comparando o uso ou não da equação de vizinhança. Para o CNN-Padrão, a hipótese nula é rejeitada para $\alpha < 0.01$. Para os outros dois modelos o teste estatístico considera os resultados equivalentes, mas ainda sim, todos os modelos apresentam um rank menor(melhor) para o não uso da vizinhança.

Tabela 22 – Resultado do teste Wilcoxon comparando o uso ou não da estratégia de vizinhança com a base de dados dividida em 2 desfechos.

		Sem vizinhança
CNN-Padrão	Com vizinhança	0.0034 (>)
CNN-Concat	Com vizinhança	0.1257 (\approx)
CNN-Multi	Com vizinhança	0.1970 (\approx)

A Tabela 23 mostra o resultado do teste estatístico em razão do tamanho do mapa espacial considerado. O teste rejeita a hipótese nula para $\alpha < 0.01$ nas arquiteturas CNN-Padrão e CNN-Multi, e falha para rejeitá-la para o modelo CNN-Concat.

A Tabela 24 compara os valores utilizados pela heurística de atenção. Nos modelos CNN-Concat e CNN-Multi, a heurística 3 e heurística 4 rejeitam a hipótese nula com um $\alpha = 0.05$ quando comparadas com o não uso da heurística (heurística 5), sendo as duas heurísticas consideradas estatisticamente superiores à moda. Na CNN-Padrão e CNN-Concat ambos valores também rejeitam a hipótese nula quando comparados à heurística 2 com $\alpha = 0.06$. As hipóteses 3 e 4 são estatisticamente equivalentes em todos os modelos.

Tabela 23 – Resultado do teste Wilcoxon comparando o tamanho do Mapa com a base de dados dividida em 2 desfechos.

		Mapa 5
CNN-Padrão	Mapa 7	0.0008 (>)
CNN-Concat	Mapa 7	0.0679 (\approx)
CNN-Multi	Mapa 7	0.0001 (>)

Tabela 24 – Resultado do teste Wilcoxon comparando o valor utilizado na heurística com a base de dados dividida em 2 desfechos.

		Heurística 2	Heurística 3	Heurística 4
CNN-Padrão	Heurística 3	0.0547 (\approx)	-	-
	Heurística 4	0.0547 (\approx)	0.2500 (\approx)	-
	Heurística 5	0.3828 (\approx)	0.7422 (\approx)	0.1484 (\approx)
CNN-Concat	Heurística 3	0.0001 (<)	-	-
	Heurística 4	0.0018 (<)	0.1207 (\approx)	-
	Heurística 5	0.3309 (\approx)	0.0001 (>)	0.0001 (>)
CNN-Multi	Heurística 3	0.0934 (\approx)	-	-
	Heurística 4	0.1591 (\approx)	1.0000 (\approx)	-
	Heurística 5	0.5282 (\approx)	0.0335 (>)	0.0092 (>)

Como visto na Tabela 25, tanto o modelo CNN-Padrão quanto CNN-Concat testam os modelos C1 e C2. O teste de Wilcoxon rejeita a hipótese nula em ambos os modelos considerando um nível de significância de 0.05, demonstrando que o modelo C1 é estatisticamente superior ao C2. Apesar disso, o melhor resultado para o CNN-Concat usando ou não a heurística é com o modelo C2.

Tabela 25 – Resultado do teste Wilcoxon comparando os modelos de rede C1 e C2 para a estrutura CNN-Padrão com base de dados dividida em 2 desfechos.

		Modelo C1
CNN-Padrão	Modelo C2	0.0063 (>)
CNN-Concat	Modelo C2	0.0297 (>)

A Tabela 26 exibe o resultado do teste estatístico de Wilcoxon. A hipótese nula é rejeitada entre todos os algoritmos, sendo a LSVM estatisticamente superior ao KNN e GSVM e o KNN estatisticamente superior ao GSVM. Por fim, o número de divisões é estatisticamente equivalente pelo teste de Wilcoxon, como apresentado na Tabela 27.

Tabela 26 – Resultado do teste Wilcoxon comparando os algoritmos para a estrutura CNN-Concat com base de dados dividida em 2 desfechos.

		KNN	LSVM
CNN-concat	LSVM	0.0001 (<)	-
	GSVM	0.00001 (>)	0.00001 (>)

Tabela 27 – Resultado do teste Wilcoxon comparando 10 e 20 divisões para a estrutura CNN-Concat com base de dados dividida em 2 desfechos.

10 Divisões		
CNN-Concat	20 Divisões	0.7994 (\approx)

6.6 Resultados PPC para Três Desfechos

A Tabela 28 apresenta os resultados dos modelos CNN considerando o PPC para três desfechos. Interessante notar que todos eles superaram tanto o benchmark baseado em quantificadores (MLP) quanto o melhor resultado do reservóir (MS-ESN-2). Dentre os três modelos de CNN, a CNN-Multi teve novamente pior resultado apesar de sua estratégia mais sofisticada para criar diferentes representações em frequências e escalas. A rede convolucional consegue criar representações em multifrequências e multiescalas de forma automática, então a representação de exame pode se adequar melhor a essa representação automática do que usar as transformações fixas como entrada. A CNN-Concat obteve melhor resultado que a CNN-Padrão, provavelmente pela divisão da sequência em trechos menores, os quais poderiam reduzir um pouco a complexidade inerente a processar trechos maiores do exame, especialmente considerando o cenário com três desfechos. A melhor configuração da CNN-Concat utilizou um mapa de tamanho 7, sem o uso de vizinhança, com o modelo C2 e utilizando o KNN, o qual parece ter se adaptado bem a representação criada. A CNN-Padrão utilizou a mesma estrutura e parâmetros que com dois desfechos. Com exceção do CNN-Concat, tal modelo superou bem os demais em termos de F1-macro.

Tabela 28 – Acurácia, precision, recall F1-macro com a melhor configuração de cada arquitetura anterior e para a melhor configuração dos testes com os modelos convolucionais, com a base de dados dividida em 3 desfechos.

Modelo (Readout)	Mapa	Viz.	Arq.	Div.	Acc.	Prec.	Rec.	F1-macro
MLP	-	-	-	-	0.4361	0.3607	0.3611	0.3505
MS-ESN-2 (KNN)	-	-	-	10	0.4972	0.4187	0.4284	0.4123
CNN-Multi	5	T	-	-	0.5111	0.4652	0.4897	0.4397
CNN-Padrão	5	T	C1	-	0.5389	0.4852	0.4970	0.4746
CNN-Concat. (KNN)	7	F	C2	20	0.5306	0.5039	0.5598	0.5165

A Tabela 29 mostra que o uso ou não da vizinhança é estatisticamente equivalente

quando se considera o PPC para três desfechos. Já a Tabela 30 mostra que o Mapa de tamanho 5 é estatisticamente superior do que o de tamanho 7 para o modelo CNN-Multi considerando um nível de significância de 0.05. Enquanto isso, para os outros dois modelos, o tamanho do mapa é estatisticamente equivalente.

Tabela 29 – Resultado do teste Wilcoxon comparando o uso ou não da estratégia de vizinhança com a base de dados dividida em 3 desfechos.

		Sem vizinhança
CNN-Padrão	Com vizinhança	0.6250 (\approx)
CNN-Concat	Com vizinhança	0.1961 (\approx)
CNN-Multi	Com vizinhança	0.2500 (\approx)

Tabela 30 – Resultado do teste Wilcoxon comparando o tamanho do Mapa com a base de dados dividida em 3 desfechos.

			Mapa 5
CNN-Padrão	Mapa 7	0.8750 (\approx)	
CNN-Concat	Mapa 7	0.7174 (\approx)	
CNN-Multi	Mapa 7	0.0391 ($>$)	

A Tabela 31 mostra o resultado do teste de Wilcoxon para os modelos C1 e C2 utilizados pelas arquiteturas CNN-Padrão e CNN-Concat. A hipótese nula falha em ser rejeitada, indicando que para três desfechos as arquiteturas são estatisticamente equivalentes.

Tabela 31 – Resultado do teste Wilcoxon comparando os modelos de rede C1 e C2 com a base de dados dividida em 3 desfechos.

			Modelo C1
CNN-Padrão	Modelo C2	0.1250 (\approx)	
CNN-Concat	Modelo C2	0.4691 (\approx)	

A Tabela 32 tem os resultados para o teste de Wilcoxon comparando os algoritmos de classificação. A hipótese nula é rejeitada para um nível de significância de 0.05 indicando que o KNN e o LSVM são estatisticamente superiores ao GSVM e equivalentes entre si. Por fim, a Tabela 33 exibe o resultado do teste para os dois valores de divisão, mostrando que ambos são estatisticamente equivalentes não rejeitando a hipótese nula para um $\alpha = 0.1$

Tabela 32 – Resultado do teste Wilcoxon comparando os algoritmos com a base de dados dividida em 3 desfechos.

		KNN	LSVM
CNN-Concat	LSVM	0.8603 (\approx)	-
	GSVM	0.0001 ($>$)	0.0001 ($>$)

Tabela 33 – Resultado do teste Wilcoxon comparando 10 e 20 divisões para a estrutura CNN-Concat com base de dados dividida em 3 desfechos.

		10 Divisões
CNN-Concat	20 Divisões	0.8361 (\approx)

Conclusão

Para a realização do trabalho, a utilização do EEG puro ocasiona em dificuldades, o exame contém diversas séries temporais, que juntar criam modelos de entradas com dimensionalidade muito alta. Representações para cada modelo foram propostas para cada estrutura de rede, para se utilizar das características do exame que beneficiam cada uma destas estruturas.

Os modelos de aprendizado recorrente foram desenvolvidos abordando diferentes formas de tratar as várias séries temporais provenientes dos eletrodos do EEG. A definição temporal do que é um instante de tempo para o reservóir também variou. Os testes foram realizados com dois e três desfechos.

Para os modelos de rede convolucionais, representações espaciais, utilizando as séries temporais puras do exame, podem atingir melhores resultados, a nova representação simula a geometria do sistema 10-20 do EEG. A representação foi utilizada nos 3 modelos convolucionais desenvolvidos e testados na base de dados com dois e três desfechos.

O modelo de aprendizado profundo recorrente foi desenvolvido em etapas, sendo diferentes processos necessários para a adaptação e modelagem da rede para o uso do EEG. A rede permitia liberdade em como os dados poderiam ser inseridos, então diferentes modelos de representação foram desenvolvidos para permitir fazer o uso de todos eletrodos na rede echo state. Foi investigada a melhor estrutura para criação da echo state e quais parâmetros utilizar, também foi definido um conjunto de algoritmos de classificação que fazem melhor proveito da transformação dos dados oferecida pela ESN.

Para começar, foi feito um estudo sobre a rede echo state, servindo para definir a estrutura de criação do reservóir, os parâmetros, e os melhores métodos para utilizar na camada readout. Três diferentes estruturas de conexão da rede foram observadas no reservóir, cinco bases de dados do mundo real foram aplicadas nas redes, pequeno mundo, regular e aleatória. Após a definição da melhor configuração do reservóir, foi necessário testar alguns parâmetros referentes a aplicação do reservóir no EEG, por exemplo, o número de pedaços em que o exame foi dividido.

Uma vez definida a estrutura de construção da rede echo state, quais parâmetros

e algoritmos de classificação utilizar, diferentes formas de usar as 20 séries temporais provenientes dos eletrodos no reservóir foram desenvolvidas. Os eletrodos com menos ruídos nos exames foram utilizados separadamente, para mensurar quanto podem atingir na tarefa de classificação isolado. Os testes por eletrodo tiveram resultados piores que o benchmark MLP para dois desfechos. para três desfechos o eletrodo O1, atingiu o mesmo resultado da MLP de F1-Macro 0.3505. Posteriormente com as multi sequências foram desenvolvidas, 5 arquiteturas recorrentes que combinaram todos eletrodos em diferentes modelos. Dos 5 modelos, o modelo MS-DeepESN que utilizou como base a deepESN, um reservóir desenvolvido em cacata, adaptado para inserir cada eletrodo em cada reservóir atingiu os melhores resultados. Outro modelo que teve destaque foi o MS-ESN-2, que tratou cada eletrodo separadamente pelo reservóir e juntou as transformações finais, tendo o melhor resultado para três desfechos, quase empatado com o MS-DeepESN. Analisando os algoritmos, das 3 estruturas, os métodos LSVM e o KNN revesaram entre os melhores resultados.

A arquitetura com maior F1-Macro foi a MS-DeepESN com 0.5631. Para três desfechos as melhores arquiteturas foram a MS-ESN-2 e MS-DeepESN, com F1-Macro de ambos atingindo aproximadamente 0.41. A rede echo state superou o uso dos exames em classificadores estados-da-arte sem o reservóir, mostrando que a rede recorrente provê uma melhoria preditiva, para 2 desfechos a extração de atributos ainda supera os resultados da ESN. Com 3 desfechos, o reservóir supera o baseline em 6%

No que diz respeito ao uso da rede convolucional no EEG, outras representações foram necessárias para adaptar o EEG para uso da rede. A representação transformou o exame, organizando as séries temporais em uma matriz 3D, que reproduz a geometria espacial do sistema de coleta do EEG (10-20) ao longo do tempo. Dessa forma, além dos benefícios da conexão sequencial da serie temporal, a representação passa a se beneficiar da relação entre os eletrodos vizinhos. O modelo monta uma imagem de cada instante do exame e é ideal para uso nas redes convolucionais.

A rede convolucional pode ser construída de diferentes formas, com diferentes números de inputs e camadas, os valores em suas camadas podem ser utilizados como transformações. Dos três modelos desenvolvidos, o primeiro utiliza a matriz de entrada inteira em uma CNN-Padrão, algumas camadas convolucionais em sequência, algumas reduções de espaço, camadas finais densamente conectadas. O segundo modelo, CNN-Concat, na dimensão temporal, a matriz formada pelo exame é dividida em vários segmentos, cada segmento passa por uma rede convolucional, porém para realizar uma extração de atributos. De cada pedaço a saída da rede será os valores calculados para a última camada convolucional, atributos extraídos automaticamente da rede. Os pedaços do mesmo exame tiveram seus atributos agrupados e foram testados em 3 classificadores, o KNN, o LSVM e o GSVM. O terceiro modelo foi baseado no artigo 2.2.5.2, a representação final do EEG sofre diversas transformações criando representações multi escala e multi frequência. To-

das as representações são utilizadas em simultâneo por uma rede convolucional de várias entradas, com camadas convolucionais de múltiplas entradas, que utiliza as camadas densamente conectadas com uma camada softmax final para o papel de classificação.

A base de dados é organizada de forma que cada paciente tem 9 trechos do exame utilizado para classificação, e o prognóstico final é moda das classificações encontradas. Com 2 desfechos, foi heurística foi proposta para trocar a utilização da moda. A heurística está apresentada na Equação 24 e é uma generalização da moda, e é equivalente à moda quando $x = 5$. A heurística testa valores diferentes para x definir o prognóstico final, os testes realizados na MS-DeepESN, apontam melhora do resultado em 0.01 de F1-Macro.

A rede CNN-Padrão conseguiu o melhor resultado atingindo 0.6391 de F1-Macro. Às três redes convolucionais tiveram F1-macro superiores ao baseline MLP, o que demonstra que a representação espacial do exame consegue performar melhor que uma extração de atributos e que o uso da representação recorrente feita para a rede echo state. Reproduzir a organização dos eletrodos na cabeça através do mapa, estabelece uma relação entre as séries, que são aproveitadas na rede neural convolucional e foram responsáveis pelo aumento do desempenho. Os resultados utilizando a CNN-Concat e CNN-Multi atingiram 0.6062 e 0.5935 de F1-Macro respectivamente. Para 3 desfechos o F1-Macro da rede CNN-Concat (KNN) chegou a 0.5165 superando em 0.15 o F1-Macro de 0.3505 do baseline. As redes CNN-padrão e CNN-Multi tiveram 0.4746 e 0.4397 de F1-Macro respectivamente. A heurística com $x = \{3, 4\}$ foram consideradas superiores estatisticamente que a de valor 5 e 2 segundo o teste de Wilcoxon para os resultados obtidos. Também foi definido o mapa de tamanho 5 é estatisticamente superior que ao mapa de tamanho 7.

7.1 Principais Contribuições

Após a realização dos testes, foi notada a dificuldade de se utilizar os exame EEG puro, devido ao objeto conter não somente uma e sim dezenas de séries temporais, que podem ser representadas e utilizadas nas redes de formas diferentes. Variando dos outros trabalhos que geralmente utilizam artefatos e atributos obtidos a partir do exame, foi utilizada diferentes representações utilizando as séries dos eletrodos ou geometrias espaciais. O resultado utilizando as redes convolucionais demonstraram o quanto alterar a representação pode ajudar na função de classificação. Além disso, o reservóir é geralmente utilizado em séries temporais onde cada pedaço no instante t contém um rótulo, o que é diferente da forma utilizada neste trabalho, onde o rótulo não é por pedaço e sim por um exame completo, e consegue desempenhar um papel de melhora nos resultados.

7.2 Trabalhos Futuros

Os seguintes pontos pretendem ser abordados em trabalhos futuros:

- ❑ Neste trabalho foram utilizadas trechos de exame EEG selecionados pelos médicos. Nossas investigações futuras contemplam a análise de trechos do exame automaticamente selecionados, isto é, sem a necessidade de seleção por parte do médico. Além de cada exame utilizado possuir 4000 pontos, o exame total pode ter mais de 100 mil pontos então como trabalho futuro é possível realizar testes utilizando pedaços maiores e aproveitando melhor o exame.
- ❑ Neste trabalho foram utilizadas redes *echo state* para o aprendizado de padrões sequenciais. Além desse tipo de arquitetura, também pretendemos avaliar modelos de rede recorrentes baseadas em LSTM e redes de atenção.
- ❑ Serão desenvolvidos modelos de autoML para encontrar configurações automáticas dos modelos capazes de melhorar os resultados.

7.3 Contribuições em Produção Bibliográfica

Este trabalho resultou na publicação de um artigo até o presente momento. É importante destacar que outros artigos permanecem em fase de revisão e serão publicados preferencialmente em veículos relevantes da área. É o caso, por exemplo das arquiteturas multi-sequenciais de redes *echo state* e das resoluções espaciais sobre as quais arquiteturas de CNNs multi-dimensionais foram desenvolvidas.

- ❑ Lucas Z. Bissaro, Yaochu Jin e Murillo G. Carneiro. Regular echo state networks: simple and accurate reservoir models to real-world applications. In: Proceedings of the 36th Annual ACM Symposium on Applied Computing, p. 1063–1069, 2021.
- ❑ Lucas Z. Bissaro, Camila D. Ramos, João B. Destro-Filho e Murillo G. Carneiro. Multi-sequential echo state networks of EEG signals for the prognosis of comatose patients (em revisão).
- ❑ Lucas Z. Bissaro, Camila D. Ramos, João B. Destro-Filho e Murillo G. Carneiro. Spatial representation of EEG signals for the prognosis of comatose patients using convolutional neural networks (em revisão).

Referências

- 21 electrodes of International 10-20 system for EEG. 2021. <[https://en.wikipedia.org/wiki/10%E2%80%9320_system_\(EEG\)](https://en.wikipedia.org/wiki/10%E2%80%9320_system_(EEG))>. Accessed: 2021-02-18.
- ABTO. **Associação Brasileira de Transplante de Órgãos**. 2021. <<https://site.abto.org.br/publicacoes/rbt/>>. Accessed: 2021-02-18.
- ALBAWI, S.; MOHAMMED, T. A.; AL-ZAWI, S. Understanding of a convolutional neural network. In: IEEE. **2017 International Conference on Engineering and Technology (ICET)**. [S.l.], 2017. p. 1–6.
- AMZICA, F. Basic physiology of burst-suppression. **Epilepsia**, Wiley Online Library, v. 50, p. 38–39, 2009.
- ANGHINAH, R. et al. Artefatos biológicos no eeg quantitativo. **Arquivos de Neuro-psiquiatria**, SciELO Brasil, v. 64, p. 264–268, 2006.
- ANSARI, A. H. et al. Quiet sleep detection in preterm infants using deep convolutional neural networks. **Journal of neural engineering**, IOP Publishing, v. 15, n. 6, p. 066006, 2018.
- ANTONIADES, A. et al. Detection of interictal discharges with convolutional neural networks using discrete ordered multichannel intracranial eeg. **IEEE Transactions on Neural Systems and Rehabilitation Engineering**, IEEE, v. 25, n. 12, p. 2285–2294, 2017.
- AWAD, M.; KHANNA, R. Support vector regression. In: **Efficient learning machines**. [S.l.]: Springer, 2015. p. 67–80.
- BABILONI, C. et al. International federation of clinical neurophysiology (ifcn)–eeg research workgroup: Recommendations on frequency and topographic analysis of resting state eeg rhythms. part 1: Applications in clinical research studies. **Clinical Neurophysiology**, Elsevier, v. 131, n. 1, p. 285–307, 2020.
- BAGNATO, S. et al. Prognostic value of standard eeg in traumatic and non-traumatic disorders of consciousness following coma. **Clinical Neurophysiology**, Elsevier, v. 121, n. 3, p. 274–280, 2010.
- BALOGLU, U. B. et al. Classification of myocardial infarction with multi-lead eeg signals and deep cnn. **Pattern Recognition Letters**, Elsevier, v. 122, p. 23–30, 2019.

- BANSAL, A. et al. Non traumatic coma. **The Indian Journal of Pediatrics**, Springer, v. 72, n. 6, p. 467–473, 2005.
- BASHIVAN, P. et al. Learning representations from eeg with deep recurrent-convolutional neural networks. **arXiv preprint arXiv:1511.06448**, 2015.
- BEGUM, S.; BARUA, S.; AHMED, M. U. Physiological sensor signals classification for healthcare using sensor data fusion and case-based reasoning. **Sensors**, Multidisciplinary Digital Publishing Institute, v. 14, n. 7, p. 11770–11785, 2014.
- BENBADIS, S. R. What type of eeg (or eeg-video) does your patient need? **Expert review of neurotherapeutics**, Taylor & Francis, v. 15, n. 5, p. 461–464, 2015.
- BENGIO, Y.; SIMARD, P.; FRASCONI, P. Learning long-term dependencies with gradient descent is difficult. **IEEE transactions on neural networks**, IEEE, v. 5, n. 2, p. 157–166, 1994.
- BISSARO, L. Z.; JIN, Y.; CARNEIRO, M. G. Regular echo state networks: simple and accurate reservoir models to real-world applications. In: **Proceedings of the 36th Annual ACM Symposium on Applied Computing**. [S.l.: s.n.], 2021. p. 1063–1069.
- BLACK, C. L. **Method and system for training an artificial neural network**. [S.l.]: Google Patents, 2001. US Patent 6,269,351.
- BURGES, C. J. A tutorial on support vector machines for pattern recognition. **Data mining and knowledge discovery**, Springer, v. 2, n. 2, p. 121–167, 1998.
- CAPRON, A. M.; KASS, L. R. A statutory definition of the standards for determining human death: An appraisal and a proposal. **University of Pennsylvania Law Review**, JSTOR, v. 121, n. 1, p. 87–118, 1972.
- CASSON, A. J. et al. Electroencephalogram. In: **Seamless Healthcare Monitoring**. [S.l.]: Springer, 2018. p. 45–81.
- CHEN, Y. et al. **The UCR Time Series Classification Archive**. 2015. <www.cs.ucr.edu/~eamonn/time_series_data/>.
- CHEN, Z. et al. An empirical eeg analysis in brain death diagnosis for adults. **Cognitive Neurodynamics**, Springer, v. 2, n. 3, p. 257–271, 2008.
- CHOUIKHI, N. et al. Pso-based analysis of echo state network parameters for time series forecasting. **Applied Soft Computing**, Elsevier, v. 55, p. 211–225, 2017.
- COVER, T.; HART, P. Nearest neighbor pattern classification. **IEEE transactions on information theory**, IEEE, v. 13, n. 1, p. 21–27, 1967.
- CRAIK, A.; HE, Y.; CONTRERAS-VIDAL, J. L. Deep learning for electroencephalogram (eeg) classification tasks: a review. **Journal of neural engineering**, IOP Publishing, v. 16, n. 3, p. 031001, 2019.
- CUI, Z.; CHEN, W.; CHEN, Y. Multi-scale convolutional neural networks for time series classification. **arXiv preprint arXiv:1603.06995**, 2016.
- DANIELSSON, P.-E. Euclidean distance mapping. **Computer Graphics and image processing**, Elsevier, v. 14, n. 3, p. 227–248, 1980.

- DEMŠAR, J. Statistical comparisons of classifiers over multiple data sets. **Journal of Machine learning research**, v. 7, n. Jan, p. 1–30, 2006.
- DENG, Z.; ZHANG, Y. Complex systems modeling using scale-free highly-clustered echo state network. In: IEEE. **The 2006 IEEE International Joint Conference on Neural Network Proceedings**. [S.l.], 2006. p. 3128–3135.
- DESHPANDE, A. **A Beginner's Guide To Understanding Convolutional Neural Networks**. 2020. <<https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/>>. Accessed: 2020-08-18.
- _____. **A Beginner's Guide To Understanding Convolutional Neural Networks Part 2**. 2020. <<https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks-Part-2/>>. Accessed: 2020-08-18.
- DEYNE, C. D. et al. Use of continuous bispectral eeg monitoring to assess depth of sedation in icu patients. **Intensive care medicine**, Springer, v. 24, n. 12, p. 1294–1298, 1998.
- DUA, D.; GRAFF, C. **UCI Machine Learning Repository**. 2017. Disponível em: <<http://archive.ics.uci.edu/ml>>.
- EBERSOLE, J. S.; PEDLEY, T. A. **Current practice of clinical electroencephalography**. [S.l.]: Lippincott Williams & Wilkins, 2003.
- EEG vs. MRI vs. fMRI What are the Differences? 2021. <<https://imotions.com/blog/eeg-vs-mri-vs-fmri-differences/>>. Accessed: 2021-07-25.
- FERREIRA, A. A.; LUDERMIR, T. B. Genetic algorithm for reservoir computing optimization. In: IEEE. **2009 International Joint Conference on Neural Networks**. [S.l.], 2009. p. 811–815.
- FREEMAN, W. J.; QUIROGA, R. Q. Electroencephalography. In: **Imaging Brain Function With EEG**. [S.l.]: Springer, 2013. p. 1–19.
- FU, J. et al. Deep learning approaches using 2d and 3d convolutional neural networks for generating male pelvic synthetic computed tomography from magnetic resonance imaging. **Medical physics**, Wiley Online Library, v. 46, n. 9, p. 3788–3798, 2019.
- GALLICCHIO, C.; MICHELI, A. Deep echo state network (deepesn): A brief survey. **arXiv preprint arXiv:1712.04323**, 2017.
- GALVÁN, I. M. et al. A lazy approach for machine learning algorithms. In: SPRINGER. **IFIP International Conference on Artificial Intelligence Applications and Innovations**. [S.l.], 2009. p. 517–522.
- GARDNER, M. W.; DORLING, S. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. **Atmospheric environment**, Elsevier, v. 32, n. 14-15, p. 2627–2636, 1998.
- GERRARD, P.; ZAFONTE, R.; GIACINO, J. T. Coma recovery scale—revised: evidentiary support for hierarchical grading of level of consciousness. **Archives of physical medicine and rehabilitation**, Elsevier, v. 95, n. 12, p. 2335–2341, 2014.

- GHAHRAMANI, Z. Unsupervised learning. In: SPRINGER. **Summer School on Machine Learning**. [S.l.], 2003. p. 72–112.
- GOLKOWSKI, D. et al. Simultaneous eeg–pet–fmri measurements in disorders of consciousness: an exploratory study on diagnosis and prognosis. **Journal of neurology**, Springer, v. 264, n. 9, p. 1986–1995, 2017.
- GOLLWITZER, S. et al. Early prediction of delayed cerebral ischemia in subarachnoid hemorrhage based on quantitative eeg: a prospective study in adults. **Clinical Neurophysiology**, Elsevier, v. 126, n. 8, p. 1514–1523, 2015.
- GOODFELLOW, I. et al. **Deep learning**. [S.l.]: MIT press Cambridge, 2016. v. 1.
- GRAVES, A.; MOHAMED, A.-r.; HINTON, G. Speech recognition with deep recurrent neural networks. In: IEEE. **2013 IEEE international conference on acoustics, speech and signal processing**. [S.l.], 2013. p. 6645–6649.
- GUYTON, A. C. **Tratado de fisiologia médica**. [S.l.]: Elsevier Brasil, 2006.
- HAAS, L. F. Hans berger (1873–1941), richard caton (1842–1926), and electroencephalography. **Journal of Neurology, Neurosurgery & Psychiatry**, BMJ Publishing Group Ltd, v. 74, n. 1, p. 9–9, 2003.
- HAN, J.; MORAGA, C. The influence of the sigmoid function parameters on the speed of backpropagation learning. In: SPRINGER. **International Workshop on Artificial Neural Networks**. [S.l.], 1995. p. 195–201.
- HAUPT, W. F. et al. Coma and cerebral imaging. **Springerplus**, SpringerOpen, v. 4, n. 1, p. 1–11, 2015.
- HAYKIN. **Uso de Redes Neurais Artificiais para o desenvolvimento de modelos de previsão da condição de pavimentos de aeroportos**. 2021. <https://www.researchgate.net/figure/Figura-2-Modelo-de-um-Neuronio-Artificial-HAYKIN-2001_fig2_305463781>. Accessed: 2021-03-22.
- HE, K. et al. Deep residual learning for image recognition. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2016. p. 770–778.
- HECHT-NIELSEN, R. Theory of the backpropagation neural network. In: **Neural networks for perception**. [S.l.]: Elsevier, 1992. p. 65–93.
- HENRY, J. C. Electroencephalography: basic principles, clinical applications, and related fields. **Neurology**, AAN Enterprises, v. 67, n. 11, p. 2092–2092, 2006.
- HERMANS, M. C. et al. Quantification of eeg reactivity in comatose patients. **Clinical neurophysiology**, Elsevier, v. 127, n. 1, p. 571–580, 2016.
- HERWIG, U.; SATRAPI, P.; SCHÖNFELDT-LECUONA, C. Using the international 10–20 eeg system for positioning of transcranial magnetic stimulation. **Brain topography**, Springer, v. 16, n. 2, p. 95–99, 2003.
- HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural computation**, MIT Press, v. 9, n. 8, p. 1735–1780, 1997.

- HOMAN, R. W.; HERMAN, J.; PURDY, P. Cerebral location of international 10–20 system electrode placement. **Electroencephalography and clinical neurophysiology**, Elsevier, v. 66, n. 4, p. 376–382, 1987.
- HUANG, J.; XU, X.; ZHANG, T. Emotion classification using deep neural networks and emotional patches. In: IEEE. **2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)**. [S.l.], 2017. p. 958–962.
- IERACITANO, C. et al. A novel multi-modal machine learning based approach for automatic classification of eeg recordings in dementia. **Neural Networks**, Elsevier, v. 123, p. 176–190, 2020.
- INUBUSHI, M.; YOSHIMURA, K. Reservoir computing beyond memory-nonlinearity trade-off. **Scientific reports**, Nature Publishing Group, v. 7, n. 1, p. 1–10, 2017.
- ISHU, K. et al. Identification of motion with echo state network. In: IEEE. **Oceans’ 04 MTS/IEEE Techno-Ocean’04 (IEEE Cat. No. 04CH37600)**. [S.l.], 2004. v. 3, p. 1205–1210.
- JAEGER, H. The “echo state” approach to analysing and training recurrent neural networks-with an erratum note. **Bonn, Germany: German National Research Center for Information Technology GMD Technical Report**, v. 148, n. 34, p. 13, 2001.
- _____. Reservoir riddles: Suggestions for echo state network research. In: IEEE. **Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005**. [S.l.], 2005. v. 3, p. 1460–1462.
- _____. Echo state network. **scholarpedia**, v. 2, n. 9, p. 2330, 2007.
- JENNETT, B. Development of glasgow coma and outcome scales. **Nepal Journal of Neuroscience**, v. 2, n. 1, p. 24–28, 2005.
- JIANG, J.; TRUNDLE, P.; REN, J. Medical image analysis with artificial neural networks. **Computerized Medical Imaging and Graphics**, Elsevier, v. 34, n. 8, p. 617–631, 2010.
- JIAO, Z. et al. Deep convolutional neural networks for mental load classification based on eeg data. **Pattern Recognition**, Elsevier, v. 76, p. 582–595, 2018.
- JIRAYUCHAROENSAK, S.; PAN-NGUM, S.; ISRASENA, P. Eeg-based emotion recognition using deep learning network with principal component based covariate shift adaptation. **The Scientific World Journal**, Hindawi, v. 2014, 2014.
- JUNG, K.-H.; PARK, H.; HWANG, W. Deep learning for medical image analysis: Applications to computed tomography and magnetic resonance imaging. **Hanyang Medical Reviews**, Hanyang University School of Medicine, v. 37, n. 2, p. 61–70, 2017.
- KANDA, R. G. et al. Sugestões práticas para a descrição do eeg de rotina. **Revista Neurociências**, v. 22, n. 3, p. 464–468, 2014.
- KAWAI, Y.; PARK, J.; ASADA, M. A small-world topology enhances the echo state property and signal propagation in reservoir computing. **Neural Networks**, Elsevier, v. 112, p. 15–23, 2019.

- KAWAI, Y. et al. Echo in a small-world reservoir: Time-series prediction using an economical recurrent neural network. In: IEEE. **2017 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)**. [S.l.], 2017. p. 126–131.
- KAWANO, D. Métodos de neurociência aplicados à publicidade: uma apresentação do eletroencefalograma (eeg), ressonância magnética funcional (fmri) e eyetracking. **Entre Meios**, v. 10, p. 1–16, 2013.
- KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2012. p. 1097–1105.
- KUCEWICZ, M. T. et al. Dissecting gamma frequency activity during human memory processing. **Brain**, Oxford University Press, v. 140, n. 5, p. 1337–1350, 2017.
- KUMAR, A.; KOLEKAR, M. H. Machine learning approach for epileptic seizure detection using wavelet analysis of eeg signals. In: IEEE. **2014 International Conference on Medical Imaging, m-Health and Emerging Communication Systems (MedCom)**. [S.l.], 2014. p. 412–416.
- KUO, F. Y.; SLOAN, I. H. Lifting the curse of dimensionality. **Notices of the AMS**, v. 52, n. 11, p. 1320–1328, 2005.
- KUSTERMANN, T. et al. Electroencephalography-based power spectra allow coma outcome prediction within 24 h of cardiac arrest. **Resuscitation**, Elsevier, v. 142, p. 162–167, 2019.
- LAKSHMI, M. R.; PRASAD, T.; PRAKASH, D. V. C. Survey on eeg signal processing methods. **International Journal of Advanced Research in Computer Science and Software Engineering**, v. 4, n. 1, 2014.
- LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **nature**, Nature Publishing Group, v. 521, n. 7553, p. 436–444, 2015.
- LEHEMBRE, R. et al. Resting-state eeg study of comatose patients: a connectivity and frequency analysis to find differences between vegetative and minimally conscious states. **Functional neurology**, CIC Edizioni Internazionali, v. 27, n. 1, p. 41, 2012.
- LEMAIRE, J.-J. et al. Electrical modulation of neuronal networks in brain-injured patients with disorders of consciousness: A systematic review. In: ELSEVIER. **Annales francaises d’anesthesie et de reanimation**. [S.l.], 2014. v. 33, n. 2, p. 88–97.
- LI, D.; HAN, M.; WANG, J. Chaotic time series prediction based on a novel robust echo state network. **IEEE Transactions on Neural Networks and Learning Systems**, IEEE, v. 23, n. 5, p. 787–799, 2012.
- LI, F. et al. Deep models for engagement assessment with scarce label information. **IEEE Transactions on Human-Machine Systems**, IEEE, v. 47, n. 4, p. 598–605, 2016.
- LI, Y. et al. Human emotion recognition with electroencephalographic multidimensional features by hybrid deep neural networks. **Applied Sciences**, Multidisciplinary Digital Publishing Institute, v. 7, n. 10, p. 1060, 2017.

- LIPTON, Z. C.; BERKOWITZ, J.; ELKAN, C. A critical review of recurrent neural networks for sequence learning. **arXiv preprint arXiv:1506.00019**, 2015.
- LU, N. et al. A deep learning scheme for motor imagery classification based on restricted boltzmann machines. **IEEE transactions on neural systems and rehabilitation engineering**, IEEE, v. 25, n. 6, p. 566–576, 2016.
- LUKOŠEVIČIUS, M. A practical guide to applying echo state networks. In: **Neural networks: Tricks of the trade**. [S.l.]: Springer, 2012. p. 659–686.
- LUKOŠEVIČIUS, M.; JAEGER, H. Reservoir computing approaches to recurrent neural network training. **Computer Science Review**, Elsevier, v. 3, n. 3, p. 127–149, 2009.
- MAESTROVIRTUALE. **Eletroencefalograma (EEG): o que é e como é usado?** 2021. <<https://maestrovirtuale.com/eletroencefalograma-eeg-o-que-e-e-como-e-usado/>>. Accessed: 2021-05-28.
- MALIK, Z. K.; HUSSAIN, A.; WU, Q. J. Multilayered echo state machine: a novel architecture and algorithm. **IEEE Transactions on cybernetics**, IEEE, v. 47, n. 4, p. 946–959, 2016.
- MCKEOWN, D.; BONSER, R.; KELLUM, J. Management of the heartbeating brain-dead organ donor. **British journal of anaesthesia**, Oxford University Press, v. 108, n. suppl_1, p. i96–i107, 2012.
- MIROWSKI, P. et al. Classification of patterns of eeg synchronization for seizure prediction. **Clinical neurophysiology**, Elsevier, v. 120, n. 11, p. 1927–1940, 2009.
- MOHRI, M.; ROSTAMIZADEH, A.; TALWALKAR, A. **Foundations of machine learning**. [S.l.]: MIT press, 2018. 2 p.
- MULERT, C.; LEMIEUX, L. **EEG-fMRI: physiological basis, technique, and applications**. [S.l.]: Springer Science & Business Media, 2009.
- NAIR, V.; HINTON, G. E. Rectified linear units improve restricted boltzmann machines. In: **Proceedings of the 27th international conference on machine learning (ICML-10)**. [S.l.: s.n.], 2010. p. 807–814.
- NAQA, I. E.; MURPHY, M. J. What is machine learning? In: **machine learning in radiation oncology**. [S.l.]: Springer, 2015. p. 3–11.
- NASTESKI, V. An overview of the supervised machine learning methods. **Horizons**. b, v. 4, p. 51–62, 2017.
- NATSCHLÄGER, T.; MAASS, W.; MARKRAM, H. The "liquid computer": A novel strategy for real-time computing on time series. **Special issue on Foundations of Information Processing of TELEMATIK**, v. 8, n. ARTICLE, p. 39–43, 2002.
- NIEDERMEYER, E.; SILVA, F. L. D. **Electroencephalography—Basic principles, clinical applications, and related fields**. [S.l.]: Urban & Schwarzenberg,, 2005. v. 5. 1309–1309 p.
- NOIRHOMME, Q. et al. Automated analysis of background eeg and reactivity during therapeutic hypothermia in comatose patients after cardiac arrest. **Clinical EEG and neuroscience**, Sage Publications Sage CA: Los Angeles, CA, v. 45, n. 1, p. 6–13, 2014.

- NOOR, N. S. E. M.; IBRAHIM, H. Machine learning algorithms and quantitative electroencephalography predictors for outcome prediction in traumatic brain injury: A systematic review. **IEEE Access**, IEEE, v. 8, p. 102075–102092, 2020.
- NUNEZ, P. L.; SRINIVASAN, R. et al. **Electric fields of the brain: the neurophysics of EEG**. [S.l.]: Oxford University Press, USA, 2006.
- OH, S. L. et al. A deep learning approach for parkinson’s disease diagnosis from eeg signals. **Neural Computing and Applications**, Springer, v. 32, n. 15, p. 10927–10933, 2020.
- O’SHEA, K.; NASH, R. An introduction to convolutional neural networks. **arXiv preprint arXiv:1511.08458**, 2015.
- PAIVA, M. R. de et al. Avaliação visual dos padrões eletroencefalográficos de pacientes clinicamente em coma. **Brazilian Journal of Health Review**, v. 1, n. 2, p. 447–455, 2018.
- PASCANU, R.; MIKOLOV, T.; BENGIO, Y. On the difficulty of training recurrent neural networks. In: **International conference on machine learning**. [S.l.: s.n.], 2013. p. 1310–1318.
- PEARLMUTTER, B. A. Gradient calculations for dynamic recurrent neural networks: A survey. **IEEE Transactions on Neural networks**, IEEE, v. 6, n. 5, p. 1212–1228, 1995.
- PEREIRA, A. et al. Cross-subject eeg event-related potential classification for brain-computer interfaces using residual networks. 2018.
- PERRI, C. D. et al. Measuring consciousness in coma and related states. **World journal of radiology**, Baishideng Publishing Group Inc, v. 6, n. 8, p. 589, 2014.
- POSNER, J. B. et al. **Plum and Posner’s diagnosis of stupor and coma**. [S.l.]: OUP USA, 2007. v. 71.
- PRATSCHKE, J. et al. Brain death and its influence on donor organ quality and outcome after transplantation1. **Transplantation**, LWW, v. 67, n. 3, p. 343–348, 1999.
- PUGIN, D. et al. T114. predicting coma outcome using resting-state fmri and machine learning. **Clinical Neurophysiology**, Elsevier, v. 129, p. e46, 2018.
- RABIN, M. J. A. et al. Sensitivity learning oriented nonmonotonic multi reservoir echo state network for short-term load forecasting. In: IEEE. **2013 International Conference on Informatics, Electronics and Vision (ICIEV)**. [S.l.], 2013. p. 1–6.
- RAMOS, C. D. **Caracterização do eletroencefalograma normal em situação de vigília: elaboração da base de dados e análise quantitativa**. [S.l.]: Universidade Federal de Uberlândia, 2017.
- RAMOS, C. D.; CARNEIRO, M. G.; DESTRO-FILHO, J. B. Machine learning as a tool for the prognosis of comatose patients using electroencephalographic signals (submitted). **IEEE Transactions on Neural Networks and Learning Systems**, v. 29, n. 8, p. 3361–3373, 2021.

- RAWAT, W.; WANG, Z. Deep convolutional neural networks for image classification: A comprehensive review. **Neural computation**, MIT Press, v. 29, n. 9, p. 2352–2449, 2017.
- RICARDO, J. A. et al. The impact of eeg in the diagnosis and management of patients with acute impairment of consciousness. **Arquivos de neuro-psiquiatria**, SciELO Brasil, v. 70, p. 34–39, 2012.
- RODRÍGUEZ, D. **Multilayer Perceptron Machine Learning project**. 2020. <<https://github.com/d-r-e/multilayer-perceptron>>. Accessed: 2021-06-09.
- ROMO-VAZQUEZ, R. et al. Eeg ocular artefacts and noise removal. In: IEEE. **2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society**. [S.l.], 2007. p. 5445–5448.
- ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. **Psychological review**, American Psychological Association, v. 65, n. 6, p. 386, 1958.
- RUBIN, D. I.; DAUBE, J. R. Clinical neurophysiology. **Oxford University Press, USA**, Oxford University Press, USA, v. 3, p. 886–886, 2009.
- RUIJTER, B. J. et al. The prognostic value of discontinuous eeg patterns in postanoxic coma. **Clinical neurophysiology**, Elsevier, v. 129, n. 8, p. 1534–1543, 2018.
- RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. **Learning internal representations by error propagation**. [S.l.], 1985.
- _____. Learning representations by back-propagating errors. **nature**, Nature Publishing Group, v. 323, n. 6088, p. 533–536, 1986.
- SABETI, M. et al. A new approach for eeg signal classification of schizophrenic and control participants. **Expert Systems with Applications**, Elsevier, v. 38, n. 3, p. 2063–2071, 2011.
- SANTANA, F. L. d. Generalizações do conceito de distância, i-distâncias, distâncias intervalares e topologia. Universidade Federal do Rio Grande do Norte, 2012.
- SARAVANAN, R.; SUJATHA, P. A state of art techniques on machine learning algorithms: a perspective of supervised learning approaches in data classification. In: IEEE. **2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)**. [S.l.], 2018. p. 945–949.
- SATHYA, R.; ABRAHAM, A. Comparison of supervised and unsupervised learning algorithms for pattern classification. **International Journal of Advanced Research in Artificial Intelligence**, Citeseer, v. 2, n. 2, p. 34–38, 2013.
- SAUNDERS, M. G. Eeg changes in metabolic disorders. **American Journal of EEG Technology**, Taylor & Francis, v. 8, n. 2, p. 41–58, 1968.
- SCHMITT, S. E. et al. Extreme delta brush: a unique eeg pattern in adults with anti-nmda receptor encephalitis. **Neurology**, AAN Enterprises, v. 79, n. 11, p. 1094–1100, 2012.

SCHORR, B. et al. Coherence in resting-state eeg as a predictor for the recovery from unresponsive wakefulness syndrome. **Journal of neurology**, Springer, v. 263, n. 5, p. 937–953, 2016.

SERGIO, A. T.; LUDERMIR, T. B. Pso for reservoir computing optimization. In: SPRINGER. **International Conference on Artificial Neural Networks**. [S.l.], 2012. p. 685–692.

SHAMWELL, J. et al. Single-trial eeg rsvp classification using convolutional neural networks. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. **Micro-and Nanotechnology Sensors, Systems, and Applications VIII**. [S.l.], 2016. v. 9836, p. 983622.

SHANMUGANATHAN, S. Artificial neural network modelling: An introduction. In: **Artificial neural network modelling**. [S.l.]: Springer, 2016. p. 1–14.

SHARANREDDY, M.; KULKARNI, P. Automated eeg signal analysis for identification of epilepsy seizures and brain tumour. **Journal of medical engineering & technology**, Taylor & Francis, v. 37, n. 8, p. 511–519, 2013.

SHARMA, S. Activation functions in neural networks. **towards data science**, v. 6, 2017.

SHOEB, A. H.; GUTTAG, J. V. Application of machine learning to epileptic seizure detection. In: **Proceedings of the 27th International Conference on Machine Learning (ICML-10)**. [S.l.: s.n.], 2010. p. 975–982.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. **arXiv preprint arXiv:1409.1556**, 2014.

SIMPRAGA, S. et al. Eeg machine learning for accurate detection of cholinergic intervention and alzheimer’s disease. **Scientific reports**, Nature Publishing Group, v. 7, n. 1, p. 1–11, 2017.

SONG, M. et al. Prognostication of chronic disorders of consciousness using brain functional networks and clinical characteristics. **Elife**, eLife Sciences Publications Limited, v. 7, p. e36173, 2018.

_____. Prognostic models for prolonged disorders of consciousness: an integrative review. **Cellular and Molecular Life Sciences**, Springer, p. 1–17, 2020.

SÖRNMO, L.; LAGUNA, P. **Bioelectrical signal processing in cardiac and neurological applications**. [S.l.]: Academic Press, 2005. v. 8.

STEPPACHER, I.; KAPS, M.; KISSLER, J. Against the odds: a case study of recovery from coma after devastating prognosis. **Annals of clinical and translational neurology**, Wiley Online Library, v. 3, n. 1, p. 61–65, 2016.

STERN, V. M. Iii.6 - intelligence code machine. In: WECHSLER, H. (Ed.). **Neural Networks for Perception**. Academic Press, 1992. p. 128 – 146. ISBN 978-0-12-741252-8. Disponível em: <<http://www.sciencedirect.com/science/article/pii/B9780127412528500133>>.

- STURM, I. et al. Interpretable deep neural networks for single-trial eeg classification. **Journal of neuroscience methods**, Elsevier, v. 274, p. 141–145, 2016.
- SUBHA, D. P. et al. Eeg signal analysis: a survey. **Journal of medical systems**, Springer, v. 34, n. 2, p. 195–212, 2010.
- SUPRATAK, A. et al. Deepsleepnet: A model for automatic sleep stage scoring based on raw single-channel eeg. **IEEE Transactions on Neural Systems and Rehabilitation Engineering**, IEEE, v. 25, n. 11, p. 1998–2008, 2017.
- SUTTON, R. S.; BARTO, A. G. **Reinforcement learning: An introduction**. [S.l.]: MIT press, 2018.
- SZEGEDY, C. et al. Going deeper with convolutions. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2015. p. 1–9.
- TANAKA, G. et al. Recent advances in physical reservoir computing: A review. **Neural Networks**, Elsevier, v. 115, p. 100–123, 2019.
- TJEPKEMA-CLOOSTERMANS, M. C. et al. Outcome prediction in postanoxic coma with deep learning. **Critical care medicine**, LWW, v. 47, n. 10, p. 1424–1432, 2019.
- TOLONEN, A. et al. Quantitative eeg parameters for prediction of outcome in severe traumatic brain injury: development study. **Clinical EEG and neuroscience**, Sage Publications Sage CA: Los Angeles, CA, v. 49, n. 4, p. 248–257, 2018.
- TRAMBAIOLLI, L. R. et al. Improving alzheimer’s disease diagnosis with machine learning techniques. **Clinical EEG and neuroscience**, SAGE Publications Sage CA: Los Angeles, CA, v. 42, n. 3, p. 160–165, 2011.
- TRAN, B. X. et al. Global evolution of research in artificial intelligence in health and medicine: a bibliometric study. **Journal of clinical medicine**, Multidisciplinary Digital Publishing Institute, v. 8, n. 3, p. 360, 2019.
- TRAUB, S. J.; WIJDICKS, E. F. Initial diagnosis and management of coma. **Emergency Medicine Clinics**, Elsevier, v. 34, n. 4, p. 777–793, 2016.
- URSPRUNG, S. et al. Radiomics of computed tomography and magnetic resonance imaging in renal cell carcinoma—a systematic review and meta-analysis. **European radiology**, Springer, p. 1–9, 2020.
- VANEGAS, M. I. et al. Machine learning for eeg-based biomarkers in parkinson’s disease. In: IEEE. **2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)**. [S.l.], 2018. p. 2661–2665.
- VIDYARATNE, L. et al. Deep recurrent neural network for seizure detection. In: IEEE. **2016 International Joint Conference on Neural Networks (IJCNN)**. [S.l.], 2016. p. 1202–1207.
- VIGÁRIO, R. N. Extraction of ocular artefacts from eeg using independent component analysis. **Electroencephalography and clinical neurophysiology**, Elsevier, v. 103, n. 3, p. 395–404, 1997.

- WANG, Z.; OATES, T. Imaging time-series to improve classification and imputation. In: **Twenty-Fourth International Joint Conference on Artificial Intelligence**. [S.l.: s.n.], 2015.
- WATTS, D. J.; STROGATZ, S. H. Collective dynamics of ‘small-world’ networks. **nature**, Nature Publishing Group, v. 393, n. 6684, p. 440, 1998.
- WHAT Is Consciousness? 2021. <<https://courses.lumenlearning.com/wsu-sandbox/chapter/what-is-consciousness/>>. Accessed: 2021-07-22.
- XIA, Y.; JAHANCHAH, C.; MANDIC, D. P. Quaternion-valued echo state networks. **IEEE transactions on neural networks and learning systems**, IEEE, v. 26, n. 4, p. 663–673, 2014.
- YILDIRIM, Ö. et al. Arrhythmia detection using deep convolutional neural network with long duration ecg signals. **Computers in biology and medicine**, Elsevier, v. 102, p. 411–420, 2018.
- YILDIZ, I. B.; JAEGER, H.; KIEBEL, S. J. Re-visiting the echo state property. **Neural networks**, Elsevier, v. 35, p. 1–9, 2012.
- YIN, J.; MENG, Y.; JIN, Y. A developmental approach to structural self-organization in reservoir computing. **IEEE transactions on autonomous mental development**, IEEE, v. 4, n. 4, p. 273–289, 2012.
- ZAREMBA, W.; SUTSKEVER, I.; VINYALS, O. Recurrent neural network regularization. **arXiv preprint arXiv:1409.2329**, 2014.
- ZHAO, R. et al. Deep learning and its applications to machine health monitoring. **Mechanical Systems and Signal Processing**, Elsevier, v. 115, p. 213–237, 2019.
- ZIER, J. et al. Improved time to notification of impending brain death and increased organ donation using an electronic clinical decision support system. **American Journal of Transplantation**, Wiley Online Library, v. 17, n. 8, p. 2186–2191, 2017.