UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

BARRY MALICK BARQUE

PREDIÇÃO DE MICROBIOMA SAUDÁVEL BASEADA EM MICRO-ORGANISMOS PRESENTES NO CORAL *MUSSISMILIA HISPIDA*, UTILIZANDO UMA REDE NEURAL PROFUNDA

MEDIANEIRA

2021

BARRY MALICK BARQUE

PREDIÇÃO DE MICROBIOMA SAUDÁVEL BASEADA EM MICRO-ORGANISMOS PRESENTES NO CORAL *MUSSISMILIA HISPIDA,* UTILIZANDO UMA REDE NEURAL PROFUNDA

Microbiome Prediction Healthy Based on Microorganisms present in coral *Mussismilia Hispida*, Using a Deep Neural Network

Trabalho de conclusão de curso de graduação apresentado como requisito para obtenção do título de Bacharel em Ciência da Computação da Universidade Tecnológica Federal do Paraná (UTFPR).
Orientador: Prof. Dr. Pedro Luiz de Paula Filho.
Coorientadora: Profa. Dra. Deborah Catharine de Assis Leite.

MEDIANEIRA

2021

BARRY MALICK BARQUE

PREDIÇÃO DE MICROBIOMA SAUDÁVEL BASEADA EM MICRO-ORGANISMOS PRESENTES NO CORAL *MUSSISMILIA HISPIDA,* UTILIZANDO UMA REDE NEURAL PROFUNDA

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Ciência da Computação da Universidade Tecnológica Federal do Paraná (UTFPR).

Data de aprovação: 08 de dezembro de 2021

_____

Pedro Luiz de Paula Filho
Doutorado
Universidade Tecnológica Federal Do Paraná

_____

Arnaldo Candido Junior
Doutorado
Universidade Tecnológica Federal Do Paraná

_____

Jorge Aikes Junior
Mestrado
Universidade Tecnológica Federal Do Paraná

MEDIANEIRA

2021

Dedico esse trabalho de conclusão de curso a minha família, aos meus amigos, e a todas as pessoas que me apoiaram e me incentivaram até aqui.

AGRADECIMENTOS

Gostaria de agradecer à todos que de alguma forma fizeram parte deste trabalho, incluindo as instituições de ensino de Togo e do Brasil por onde passei bem como todos os professores que fizeram parte dessa jornada.

A minha família, minha namorada, por todo amor, carinho, incentivo e apoio em todos os momentos até aqui.

Ao meu orientador e minha coorientadora, que me ajudaram com suas orientações e contribuições riquíssimas para o desenvolvimento deste trabalho.

A todos os professores e colegas da universidade, que ajudaram de forma direta e indireta na conclusão deste trabalho.

Em suma, agradeço a todos os citados e também as pessoas não citadas aqui mas que de alguma forma contribuíram para a realização deste trabalho de conclusão de curso.

RESUMO

Um dos ecossistemas mais diversificados e produtivos do mundo marinho são os corais, fornecendo além do turismo, uma contribuição econômica importante aos países que possuem-os no litoral. Graças a técnicas de sequenciamento de genoma como o 16S sRNA é possível identificar os micro-organismos que formam o microbioma dos corais, que tem um papel importante na saúde destes últimos. A geração de grande quantidades de dados graças ao baixo custos de sequenciamento de genoma desde 2005 oferece uma abertura para o uso de redes neurais artificias para o avanço das ciências como a biologia e a medicina. Neste trabalho foi realizado a predição do microbioma saudável baseada em micro-organismos presentes no coral *Mussismilia hispida* coletados em cinco recifes localizados próximos a uma área marinha protegida ("Parque Natural Municipal do Recife de Fora"), utilizando uma rede neural convolucional e alguns algorítimos clássicos de aprendizagem de máquina como a SVM e a árvore de decisão, comparando os seus resultados obtidos em vários experimentos.

Palavras-chave: Inteligência artificial; Redes neurais; Recifes de coral; Microbiologia.

ABSTRACT

One of the most diversified and productive ecosystems in the marine world are corals, providing, in addition to tourism, an important economic contribution to countries that have them on the coast. Thanks to genome sequencing techniques such as 16S sRNA, it is possible to identify the microorganisms that make up the coral microbiome, which play an important role in the health of the latter. The generation of large amounts of data thanks to the low cost of genome sequencing since 2005 offers an possibility for the use of artificial neural networks for the advancement of sciences such as biology and medicine. In this work, the prediction of healthy microbiome based on microorganisms present in the coral *Mussismilia hispida* collected in five reefs located near a marine protected area ("Parque Natural Municipal do Recife de Fora") was performed, using a convolutional neural network and some classical machine learning algorithms such as SVM and decision tree, comparing their results obtained in several experiments.

Keywords: Artificial Intelligence; Neural Networks; Coral Reefs; Microbiology.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE SIGLAS

ANNs	Redes Neurais Artificias
CNNs	Redes Neurais Convolucionais
DNA	Ácido Desoxirribonucleico
GAN	Redes Adversariais Generativas
IA	Inteligência Artificial
MLPs	*Multi-Layers* Perceptron
NA	Neurônio Artificial
RNA	Ácido Ribonucleico
RNN	Redes Neurais Recorrentes
SVMs	Máquinas de Vector de Suportes

SUMÁRIO

1 INTRODUÇÃO

Os recifes de corais são um dos ecossistemas mais diversificados e produtivos do planeta, com uma contribuição econômica anual variando de trinta a trezentos e setenta e cinco bilhões de dólares. Além da contribuição econômica, serve também de barreiras naturais contra as tempestades, erosão e ciclone. Contudo, os recifes de corais consistem nos habitats que mais sofreram com as mudanças climáticas além da poluição causada pelos seres humanos nas últimas décadas. Os corais abrigam comunidades complexas de micro-organismos, incluindo dinoflagelados, fungos, bactérias e arqueias que são denominados coletivamente de microbioma do coral (COURTIAL *et al.*, 2021).

Os microrganismos presentes nos recifes de corais têm um papel importante na manutenção da saúde do hospedeiro (coral) e na resiliência do ecossistema sob estresse ambiental; entretanto, eles também são participantes importantes em ciclos de *feedback* positivos que intensificam o declínio dos recifes de corais (COURTIAL *et al.*, 2021; ZILBERBERG *et al.*, 2016).

O baixo custo de sequenciamento massivo do Ácido Desoxirribonucleico (DNA) desde 2008 proporcionou uma geração de grande quantidades de dados na área da microbiologia (THOMPSON; THOMPSON, 2020).

Portanto, neste trabalho é desenvolvida uma rede neural para predição baseada em micro-organismos presentes no microbioma do coral abordando técnicas de *deep learning*.

1.1 Objetivo geral e específicos

O objetivo geral deste trabalho consiste em realizar a predição de microbioma saudável baseada em micro-organismos presentes no coral *Mussismilia hispida*, utilizando uma rede neural profunda comparada à alguns algoritmos clássicos de aprendizagem de máquina. Este trabalho é composto pelo seguintes objetivos específicos:

- Definir uma abordagem neural para fazer a predição do microbioma saudável;
- Classificar a saúde do microbioma baseada na correlação entre os micro-organismos presentes nos corais.

- Comparar a rede neural profunda com alguns algoritmos clássicos de aprendizagem de máquina.

1.2 Justificativa

Como foi mencionado anteriormente, os micro-organismos presentes nos recifes de corais são muito importantes na manutenção da saúde dos mesmos. Apesar das capacidades adaptativas dos corais, observações e modelos preveem que mais de 90% dos recifes do mundo serão afetados por grandes episódios de branqueamento[1] até 2050 (COURTIAL *et al.*, 2021).

O recife de corais é um dos habitats mais diversificados do mar, contando com mais de $1/4$ das espécies de peixes que foram identificados, incluindo vários outros organismos citados na introdução (COURTIAL *et al.*, 2021). Esses números demonstram a urgência e a necessidade de agir para preservar este ecossistema único e essencial para a economia de muitos países.

A Inteligência Artificial (IA) vem sendo muito útil em diversas áreas, principalmente na biologia e na medicina, como o uso de algoritmo genético para descoberta virtual de drogas por meio da técnica de *docking* molecular (MAGALHÃES *et al.*, 2004); de Redes Neurais Convolucionais (CNNs) para diagnósticos de doenças e classificação de espécies (BARRÉ *et al.*, 2017);

de Redes Adversariais Generativas (GAN) (SHEN *et al.*, 2019), autoencoders e de Redes Neurais Recorrentes (RNN) para gerar moléculas candidatas a medicamentos para tratar algumas doenças se baseando no padrão molecular de algumas moléculas (GUPTA *et al.*, 2018).

Portanto, este trabalho explora uma forma de predição baseada em micro-organismos nos corais utilizando uma técnica de *Deep Learning*, trazendo uma base de dados e um trabalho livremente acessível para futuras utilizações e melhorias.

[1] O branqueamento do coral é uma síndrome que causa a descoloração dos corais devido a perda do zooxantelas

2 REFERENCIAL TEÓRICO

Neste capítulo serão apresentadas algumas definições sobre os recifes de corais, os micro-organismos e o impacto da simbiose desses micro-organismos nos recifes de corais. Também serão definidos conceitos sobre técnicas de inteligência artificial, redes neurais artificiais, técnicas de classificação utilizando aprendizado de máquina, bem como trabalhos correlatos na área.

2.1 Recifes de coral

Sob o ponto de vista geomorfológico, um recife de coral é uma estrutura rochosa, construída por organismos marinhos portadores de esqueleto calcário (ZILBERBERG *et al.*, 2016). Os recifes de corais encontram-se geralmente nas regiões rasas tropicais com aproximadamente 30% das costas ocupadas (ZILBERBERG *et al.*, 2016). Na biologia os recifes de corais são uma formação de acúmulo de organismos comumente chamados de corais (BURKE *et al.*, 2012; ZILBERBERG *et al.*, 2016). Na Figura 1 está ilustrado um exemplo de uma formação de um recife de coral.

Figura 1 – Recife de corais

Fonte: (ZILBERBERG *et al.*, 2016)

2.1.1 Os corais

Um coral é um animal que vive em simbiose com algas unicelulares chamadas de *Zooxantelas* (LEONARD; FABER, 2019). Existem mais de 6.000 espécies de corais no mundo, dentro dessas espécies, há corais duros ou *Scleractinia* que possuem um

esqueleto calcário como é o caso do *Mussismilia Hispida* na Figura 2, corais moles ou *Alcyonacea* que não possuem um esqueleto calcário exemplificado na Figura 3 e os corais pretos ou *Antipatharia* (BURKE *et al.*, 2012; ZILBERBERG *et al.*, 2016).

Figura 2 – Coral *Mussismilia hispida*

Fonte: (VERON, 2021)

Figura 3 – Zoanthus sociatus

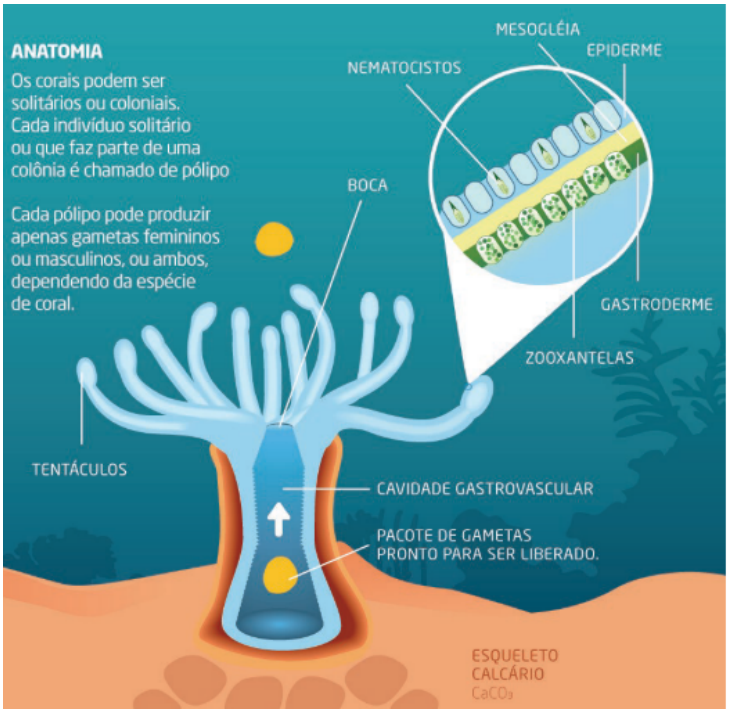
Fonte: (AQUASYMBIO, 2021)

Os corais formam colônias de pólipos, representado na Figura 4, que são organismos parecidos com pequenos sacos que tem por dentro uma cavidade gastrovascular, cada um com uma única abertura que serve como boca e ânus. Esses orifícios são geralmente rodeados por tentáculos que servem para capturar as presas (JUGANT, 2012; LEONARD; FABER, 2019).

Os pólipos vivem em simbiose com as algas que produzem 80% de suas necessidades de nutrientes durante o dia. Estes nutrientes são constituídos principalmente de oxigênio e açúcar, em troca os pólipos fornecem abrigo e dióxido de carbono a essas

algas unicelulares para realizarem a fotossíntese (JAUBERT, 2019). Durante a noite, os pólipos estendem seus tentáculos que possuem células de defesas ou organelas chamadas de nematocistos ou cnidas, representados na Figura 4, para se alimentarem de pequenos animais planctônicos e partículas orgânicas suspensas. Os nematocistos em contato com uma presa liberam toxinas capazes de imobilizar a mesma (GODOY *et al.*, 2020).

Figura 4 – Nematocistos nos tentáculos de um pólipo

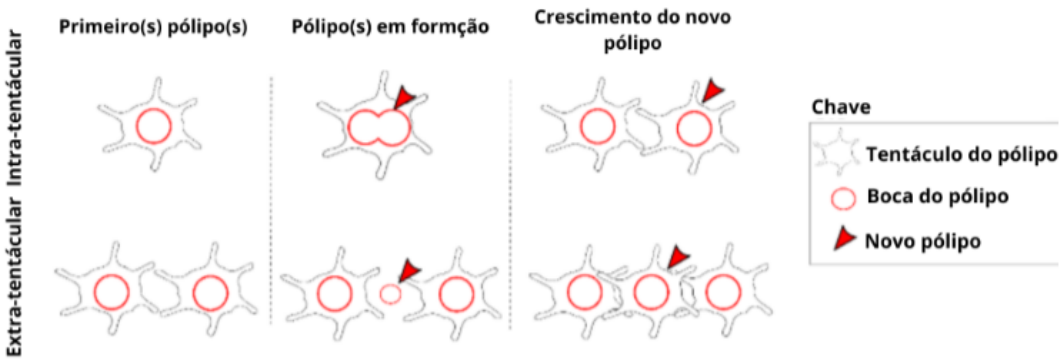


Fonte: (GODOY *et al.*, 2020)

A reprodução dos corais pode ser de forma tanto assexuada quanto de forma sexuada. A forma assexuada representada na Figura 5 é a mais comum, ela acontece durante toda a vida dos pólipos e serve para o crescimento do coral. Nesta reprodução, pode acontecer a liberação de uma larva que é uma cópia idêntica do pólipo, o brotamento que pode ser intra-tentacular por divisão do pólipo em dois, ou extra-tentacular na qual aparece geralmente um novo pólipo entre dois já existentes, ou ainda por fragmentação (GODOY *et al.*, 2020; PUISAY, 2018).

Os corais podem ter colônias fêmeas, machos ou na maioria do tempo hermafroditas como é o caso do coral *Mussismilia hispida*. No caso deste último os gametas masculinos e femininos são produzidos pelo mesmo pólipo e serão liberados posteriormente até atingirem a superfície do mar onde acontece a fecundação (GODOY *et al.*,

Figura 5 – Diferentes tipos de brotamento no coral



Fonte: Adaptado de (PUISAY, 2018)

2020; PUISAY, 2018).

2.1.2 Microbiota dos corais e sua importância

Além das zooxantelas, os corais vivem em simbiose com alguns outros micro-organismos como bactérias, vírus, fungos, protozoários, e arqueias distribuídos em toda parte do pólipo que formam a microbiota do coral (COURTIAL *et al.*, 2021; ZILBERBERG *et al.*, 2016).

Há estudos que demostraram que bactérias presentes na microbiota de uma espécie de coral são bastante parecidas com as bactérias existentes na microbiota de uma mesma espécie encontrada em uma localização diferente, por outro lado, existe o fator ambiental que pode influenciar a pré-dominância das bactérias existentes no coral (ZILBERBERG *et al.*, 2016). Essas bactérias têm papéis muito importantes na saúde do coral, protegendo-o das bactérias, dos vírus e outros organismos patógenos. As bactérias diazotróficas presentes no coral fixam o diazoto para fornecer uma fonte de azoto para o hospedeiro, que por sua vez será necessário para a sobrevivência do coral em caso de branqueamento (COURTIAL *et al.*, 2021; ZILBERBERG *et al.*, 2016).

Existem estudos que mostraram que as bactérias presentes, além de serem diferentes, habitam em uma concentração maior nas águas circundantes do que no coral, por outro lado a poluição da água pode reduzir a concentração das bactérias no hospedeiro (CATHARINE; LEITE, 2016).

2.1.3 Método de Sequenciamento de Genoma

A primeira geração de sequenciamento do DNA iniciou na década 1970 com a análise de fragmentos de detecção de posição dos nucleotídeos graças ao uso de eletroforese em gel de poliacrilamida. O genoma Ácido Ribonucleico (RNA) bacteriófago MS2 foi o primeiro sequenciamento de genoma completo na história realizado nas décadas de 1970 (THOMPSON; THOMPSON, 2020).

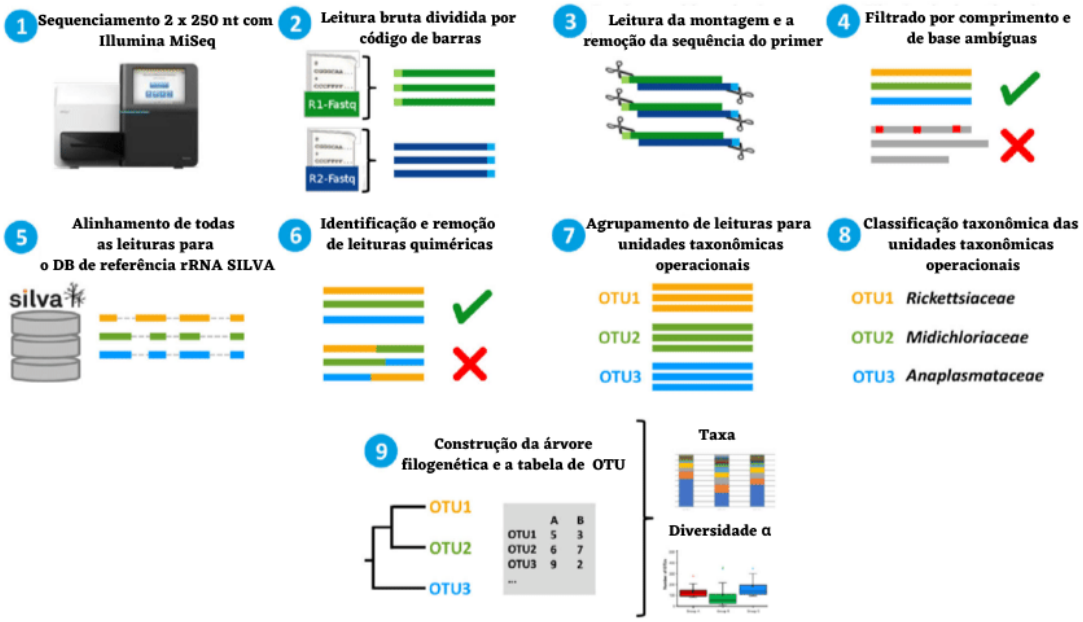
Hoje em dia o sequenciamento direcionado ou *16S rRNA amplicon sequencing* é a metodologia mais utilizada para análises de microbioma. Graças ao gene que codifica a região 16S presentes nos ribossomos de organismos procariotos, possuindo regiões conservadas e variadas, é possível diferenciar e identificar as bactérias presentes em uma dada amostra (RIYUZO, 2020).

A partir de 2008 os preços para sequenciamento de DNA despencaram, o que favoreceu o sequenciamento de DNA em larga escala, gerando assim grandes quantidades de dados a serem tratados (THOMPSON; THOMPSON, 2020). Geralmente as bactérias são agrupadas segundo a árvore classificada em unidades taxonômicas (OTUs). Um dos softwares de código aberto na área da bioinformática utilizado para tratamentos de sequências, análise de alfa e beta diversidade, além da geração da árvore filogenética é o QIIME (BOLYEN *et al.*, 2019).

O processo de sequenciamento completo até a obtenção das bactérias está representado na Figura 6, na qual a primeira parte representa a leitura das sequências pela máquina na qual são gerados dois arquivos, um lida de esquerda para direita e outra de direita para esquerda. Na terceira etapa são removidas as leituras iniciais chamadas de *primer*, após a remoção do primer, são agrupadas as sequências por tamanho e removidas aquelas que contêm ambiguidades. Na quinta etapa são armazenados os resultados das sequências em uma base de dados, na qual posteriormente serão removidas as leituras quiméricas ou incorretas. Na sétima e oitava etapa, são agrupadas e identificadas as sequências por unidades taxonômicas que são compostas por uma sequência de bases: A (Adenina), G (Guanina), T (Timina) e C (Citosina); estão representadas na Figura 7 as sequências de um possível OTUs na qual $C1P1HH1\_15$ representa o código relacionado a um micro-organismo. Na nona etapa é feita a geração da árvore filogenética e a tabela de abundância de cada OTU

nos organismos A e B.

Figura 6 – Representação do processo de sequenciamento 16sRNA



Fonte: Adaptado de Regier *et al.* (2019)

Figura 7 – Representação do sequenciamento de um OTU em um arquivo Fastq

```
>C1P1HH1_15
TACGTAGGGTGCGAGCGTTGTCCGGAATTACTGGGCGTAAAGAGCTC
GTAGGTGGTCTGTCGCGTCATTTGTGAAAGCCCGGGGCTTAACTCCG
GGTTGGCAGGTGATACGGGCATGACTGGAGTACTGTAGGGGAGACTG
GAATTCCTGGTGTAGCGGTGAAATGCGCAGATATCAGGAGGAACACC
GGTGGCGAAGGCGGGTCTCTGGGCAGTAACTGACGCTGAGGAGCGAA
AGCATGGGTAGCGAACAGG
```

Fonte: (LEITE *et al.*, 2018)

Recentemente são utilizadas, também, técnicas de *machine learning* para identificar padrões nos conjuntos de dados obtidos graças ao sequenciamento do DNA.

2.2 Redes Neurais Artificiais

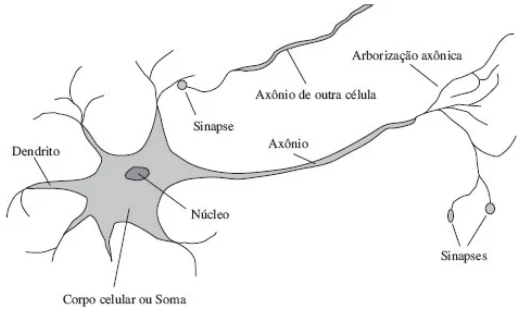
As Redes Neurais Artificias (ANNs) foram inspiradas no funcionamento de um cérebro animal, que é constituído principalmente de várias células excitáveis de neurônios (AMTHOR, 2016).

2.2.1 Neurônio biológico

Os neurônios presentes no cérebro de um ser humano estão estimados a 10^{11} (10 bilhões) (GURNEY, 1997; RUSSELL; NORVIG, 2010). O cérebro humano como qualquer *máquina* complexa é composto por vários tipos de neurônios que se comunicam entre si para efetuar uma tarefa complexa (AMTHOR, 2016).

Os neurônios do cérebro podem ser classificados em vários tipos, como os neurônios sensoriais que têm como função informar ao restante do cérebro sobre o estado do ambiente externo e interno do corpo, os neurônios de comunicação que têm por função a transmissão de sinais entre uma região do cérebro e outra, os neurônios motores que são responsáveis por controlar o comportamento dos músculos e alguns órgãos (AMTHOR, 2016). O neurônio é uma célula nervosa constituída basicamente de dendritos que garantem a recepção dos estímulos, o axônio que é a saída do neurônio, as sinapses inibidoras e excitadoras que servem de conexão com outros neurônios, e o corpo celular contém o núcleo da célula, no qual serão processadas as informações recebidas por meio dos dendritos graças as reações electro-químicas, como está representado na Figura 8 (AMTHOR, 2016).

Figura 8 – Representação de um neurônio biológico



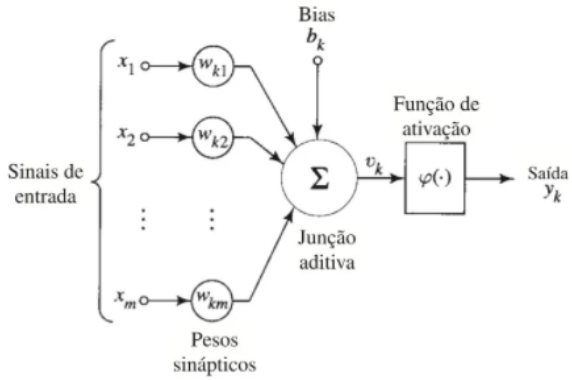
Fonte: (RUSSELL; NORVIG, 2010)

2.2.2 Neurônio Artificial

Um Neurônio Artificial (NA) tem uma estrutura funcional parecida com o neurônio biológico, como representado na Figura 9 (HAYKIN, 2007).

O funcionamento de um neurônio k é definido por um conjunto de sinal de entrada $[x_1, x_2, \ldots + x_m]$ que são multiplicadas cada uma por seus respectivos pesos sinápticos $[w_{k1}, w_{k2}, \ldots + w_{km}]$. Este resultado é somado e armazenado em (u_k) repre-

Figura 9 – Representação de um neurônio artificial não-linear



Fonte: (HAYKIN, 2007)

sentado na Equação 1, que de novo é somado ao *bias* (b_k) do neurônio. O resultado dessa operação é o potencial de ativação (v_k) representado na Equação 2, que por sua vez é aplicado na função de ativação $(\varphi(.))$ que consequentemente gera a saída do neurônio (y_k) representado na Equação 3 (FURTADO, 2019).

$$u_k = \sum_{j=1}^{m} w_{kj} x_j \tag{1}$$

$$v_k = u_k + b_k \tag{2}$$

$$y_k = \varphi(v_k) \tag{3}$$

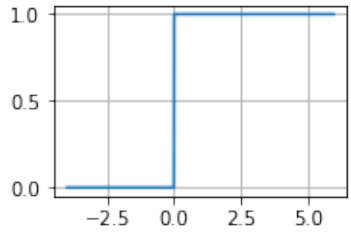
A função de ativação $\varphi(.)$ tem por principal objetivo limitar a saída do neurônio, impedindo o crescimento da saída do neurônio para o infinito (FURTADO, 2019). Abaixo estão alguns tipos de funções de ativação:

- Função limiar, definida pela Equação 4 e representada graficamente na Figura 10. Ela assume o valor 1 para qualquer valor v superior ou igual a 0, e 0 para qualquer valor v inferior a 0 (HAYKIN, 2007) ;

$$\varphi(v) = \begin{cases} 1 \; se \; v \geq 0 \\ 0 \; se \; v < 0 \end{cases} \tag{4}$$

- Função sigmoide, definida pela Equação 5 e representada graficamente na Figura 11. Esta função, assume valores contínuos que variam entre 0 e 1. Ela

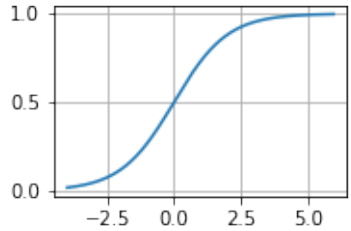
Figura 10 – Representação gráfica da função de ativação Limiar



Fonte: Autoria própria (2021)

possui um parâmetro (a) responsável pela inclinação da sigmoide. Quando o parâmetro (a) se aproxima do infinito, a função sigmoide se torna uma função limiar (SHARMA *et al.*, 2020);

Figura 11 – Representação gráfica da função de ativação Sigmoide

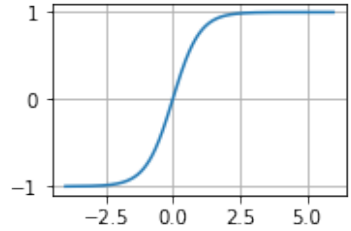


Fonte: Autoria própria (2021)

$$\varphi(v) = \frac{1}{1 + e^{-av}} \tag{5}$$

- Função tangente hiperbólica, definida pela Equação 6 e representada graficamente na Figura 12, ela varia entre -1 e 1 (WIKISTAT, 2015);

Figura 12 – Representação gráfica da função de ativação Tanh

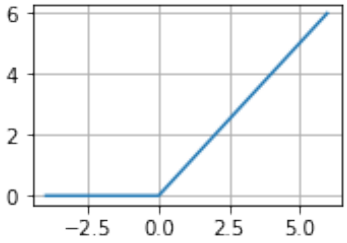


Fonte: Autoria própria (2021)

$$\varphi(v) = \tanh v \tag{6}$$

- Rectified Liniear Unit (ReLU), esta função varia entre 0 e $+\infty$ assumindo o valor de v se v for maior ou igual a 0, e 0 se v for menor que 0. Ela é definida pela Equação 7 e representada graficamente na Figura 13 (SHARMA *et al.*, 2020);

Figura 13 – Representação gráfica da função de ativação ReLU

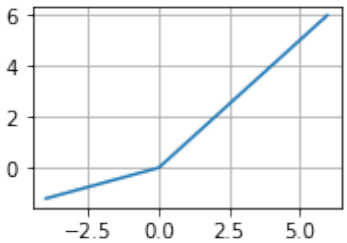


Fonte: Autoria própria (2021)

$$\varphi(v) = \begin{cases} v \; se \; v \geq 0 \\ 0 \; se \; v < 0 \end{cases} \tag{7}$$

- Leak ReLU, é uma versão adaptada da ReLU, varia entre $-\infty$ e $+\infty$, ela assume o valor do v se v for maior ou igual a 0 e é igual ao produto do v e do inverso de uma constante dada (a) se v for menor que 0. Ela é definida pela Equação 8 e representada graficamente pela Figura 14 (SHARMA *et al.*, 2020).

Figura 14 – Representação gráfica da função de ativação Leak ReLU



Fonte: Autoria própria (2021)

$$\varphi(v) = \begin{cases} v \; se \; v \geq 0 \\ \frac{v}{a} \; se \; v < 0 \end{cases} \tag{8}$$

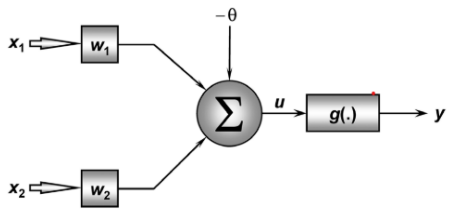
2.2.3 Perceptron

O Perceptron projetado por Rosenblatt (1958, 1968), é a forma mais básica utilizada para classificação de padrões usando uma rede neural do tipo *feedforward* (HAYKIN, 2007). O perceptron clássico consegue ter um bom desempenho em conjunto de dados linearmente separáveis como o problema dos operadores lógicos E (AND), OU (OR), $\neg E$ (NAND) e $\neg OU$ (NOR), mas em conjunto de dados não-linearmente separáveis como OU-Exclusivo (XOR), ele se mostra ineficiente. Para resolver esse último problema, foram propostas as redes multi-camadas ou *Multi-Layers* Perceptron (MLPs) que são uma associação de pelo menos dois Perceptron (FURTADO, 2019).

O Perceptron com duas entradas ilustrado na Figura 15, realiza as somas dos produtos das suas entradas $[x_1, x_2, \ldots, X_n]$ com os seus pesos respectivos $[w_1, w_2, \ldots, w_n]$, que são inicializados aleatoriamente, no qual n é a quantidade de neurônios. É subtraído um limiar θ a este último resultado. O resultado obtido por esta última operação denotada (u), passa em uma função de grau $g(u)$ que por final será a saída (y) do neurônio. Este processo pode ser descrito matematicamente pela Equação 9 (SILVA *et al.*, 2010).

$$y = \begin{cases} 1 \; se \; \sum_{i=1}^{n} w_i x_i - \theta \geq 0 \\ -1 \; se \; \sum_{i=1}^{n} w_i x_i - \theta < 0 \end{cases} \tag{9}$$

Figura 15 – Representação de um Perceptron com duas entradas

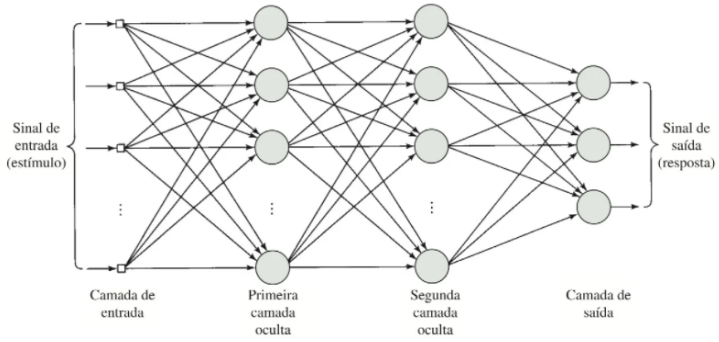


Fonte: (SILVA *et al.*, 2010)

A estrutura das redes neurais MLPs representada na Figura 16, é constituída por três camadas, a camada de entrada (*input layer*) que é responsável pela geração dos sinais da rede, uma ou mais camadas ocultas (*hidden layers*) e a última camada de saída (*output layer*) (JAIN, 2016). Na última camada é utilizada uma função de ativação dependendo do problema: regressão ou classificação (WIKISTAT, 2015).

As redes MLPs são treinadas utilizando vários tipos de algoritmos, mas um dos algoritmos mais utilizado é a retro-propagação do erro conhecido como o *Backpropagation* (HAYKIN, 2007).

Figura 16 – Representação de um perceptron com duas camadas



Fonte: (HAYKIN, 2007)

O *Backpropagation* é um algoritmo clássico muito utilizado no treinamento das redes neurais MLPs. Ele é geralmente associado ao algoritmo do gradiente descendente. Neste algoritmo é aplicada a descida do gradiente para minimizar a função de custo mais conhecido como *Loss Function* (ROJAS, 1996; YAMASHITA *et al.*, 2018). Este algoritmo tem por principal finalidade os ajustes dos pesos sinápticos conforme o erro cometido pela rede durante o treinamento (GURNEY, 1997).

O *Backpropagation* é composto por duas fases principais, a primeira é conhecida como a fase de *forward* e a segunda fase é conhecida como a fase de *backward*.

Na fase de *forward*, é passado um conjunto de dados que percorre a rede camada por camada até obter uma saída, onde a saída do neurônio anterior é a entrada do neurônio da próxima camada. A saída de um determinado neurônio j pode ser obtido pela Equação 10 e pela Equação 11, na qual, n é o n-ésimo padrão de treinamento, m representa a quantidade de neurônio na camada, w_{ji} representa o peso sináptico do neurônio , x_i a entrada do neurônio, $v_j(n)$ o potencial de ativação do neurônio e y_j a saída do neurônio j (SILVA *et al.*, 2010).

$$v_j(n) = \sum_{i=0}^{m} w_{ji}(n)x_i(n) \tag{10}$$

$$y_j = \varphi(v_j(n)) \tag{11}$$

O erro de saída do neurônio pode ser obtido pela Equação 12 e seu erro instantâneo como $\frac{1}{2}e_j^2(n)$, sendo $d_j(n)$ a saída desejada e $y_j(n)$ a saída real do neurônio (HAYKIN, 2007).

$$e_j(n) = d_j(n) - y_j(n) \tag{12}$$

Seguindo este raciocínio a soma dos erros instantâneos de todos os neurônios podem ser definida pela Equação 13, na qual C representa o conjunto de neurônios das camadas de saída (GURNEY, 1997; ROJAS, 1996).

$$E(n) = \frac{1}{2}\sum_{j \in C} e_j^2(n) \tag{13}$$

Na fase *backward*, o erro é utilizado para ajustar os pesos w_{ji} , e o cálculo está dado pela Equação 14 equivalente a Equação 15, na qual δ é o gradiente local definido pela Equação 16, η é a taxa de aprendizagem e $\varphi_j'(v_j(n))$ é a derivada da função de ativação aplicada na saída do neurônio $v_j(n)$ (HAYKIN, 2007). Ademais quanto menor a taxa de aprendizagem η, menor serão as correções efetuadas nos pesos, causando uma lenta convergência da rede. Por outro lado quanto maior a taxa de aprendizagem η maior serão as correções aplicadas nos pesos, causando uma oscilação do algoritmo que consequentemente impede a convergência da rede (HAYKIN, 2007).

$$\Delta w_{ji}(n) = -\eta \frac{\delta E(n)}{\delta w_{ji}} \tag{14}$$

$$\Delta w_{ji}(n) = \eta \delta_i(n) y_i(n) \tag{15}$$

$$\delta_i(n) = e_j(n)\varphi_j'(v_j(n)) \tag{16}$$

O processo de ajustes de pesos se repete até minimizar o erro (SILVA *et al.*, 2010).

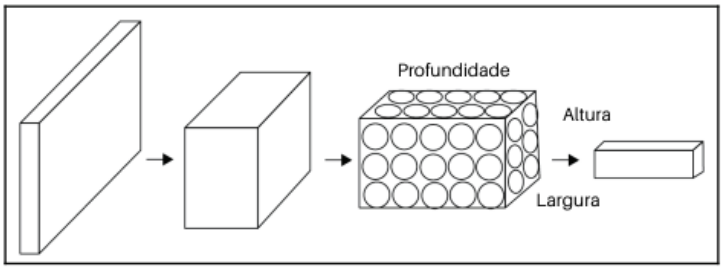
As redes neurais MLPs são um dos mais antigos métodos de aprendizagem profunda, elas foram fundamentais para criação de novas redes mais profundas com arquiteturas complexas como as redes neurais convolucionais para reconhecimento de imagem e para séries temporais, as redes neurais recorrentes para dados sequenciais como textos e séries temporais (WIKISTAT, 2015).

2.3 Redes Neurais Convolucionais

As Redes Neurais Convolucionais em inglês *Convolution Neural Network* (CNNs ou ConvNets), classificadas como redes neurais profundas, são geralmente utilizadas para classificação de imagens e têm uma arquitetura parecida as redes neurais MLPs (SEWAK *et al.*, 2018).

O processo de treinamento das CNNs é parecida com as redes neurais MLPs fazendo o uso do algoritmo *Backpropagation* para a atualização dos pesos (LIU *et al.*, 2017). A diferença entre as MLPs e as CNNs é associada as seus neurônios ocultos. As camadas em uma arquitetura CNNs tradicional são divididas em três, a saber, as camadas convolucionais, as camadas de *pooling* e as camadas totalmente conectadas ou *feedforward* (ACHARYA *et al.*, 2017). Cada neurônio é organizado em três dimensões ou seja, em altura, largura e profundidade como representado na Figura 17 (SEWAK *et al.*, 2018).

Figura 17 – Representação de um neurônio em CNN



Fonte: Adaptado de Sewak *et al.* (2018)

2.3.1 Camada convolucional

É a camada principal das CNNs, formada por uma combinação de operações linear e não-linear (ALZUBAIDI *et al.*, 2021; YAMASHITA *et al.*, 2018). Ela é composta por um conjunto de filtros contendo núcleos, os núcleos em formatos de matriz contêm valores de pesos aleatórios no início do treinamento, que são alterados durante o processo da aprendizagem para extrair características do conjunto de dados (ALZUBAIDI *et al.*, 2021; YAMASHITA *et al.*, 2018). Nesta camada o neurônio não está conectado a todos os neurônios da camada anterior, mas é conectado aos neurônios de uma determinada região especial conhecido como campo receptivo local (RANJBAR *et al.*,

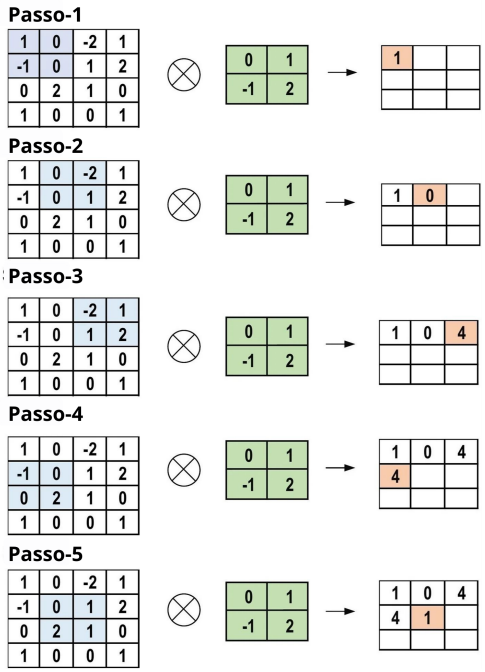
2020).

Na arquitetura CNNs os neurônios compartilham os pesos, o que reduz a quantidade dos mesmos, tornando consecutivamente o treinamento menos custoso em relação as redes neurais MLPs (ALZUBAIDI *et al.*, 2021; YAMASHITA *et al.*, 2018).

Nesta fase, o conjunto de dados de entrada são varridos pelos núcleos fazendo o produto de elemento por elemento que são somados no final (RANJBAR *et al.*, 2020).

Na Figura 18 está exemplificado o processo da fase de convolução de uma CNN com um conjunto de dados $4X4$ e um núcleo de $2X2$.

Figura 18 – Representação do processo na fase de convolução



Fonte: Adaptado de Alzubaidi *et al.* (2021)

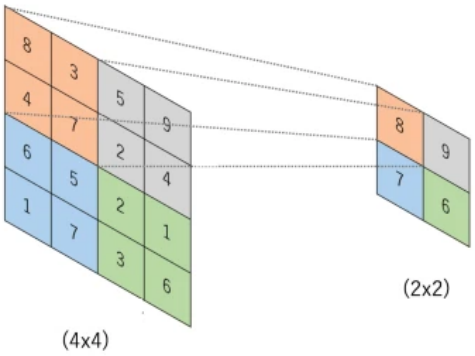
2.3.2 Camada de *Pooling*

Esta camada não efetua nenhum aprendizado e é geralmente aplicada após a fase de convolução (RANJBAR *et al.*, 2020). Ela diminua os mapas de caraterísticas geradas na fase de convolução aplicando técnicas de redução de mapa, o que ajuda a evitar o problema da maldição dos dados ou *overfitting* (SEWAK *et al.*, 2018) .

Existem vários tipos de técnicas *pooling* sendo as mais utilizadas o *Max pooling* e o *Average Pooling* (RANJBAR *et al.*, 2020; SEWAK *et al.*, 2018).

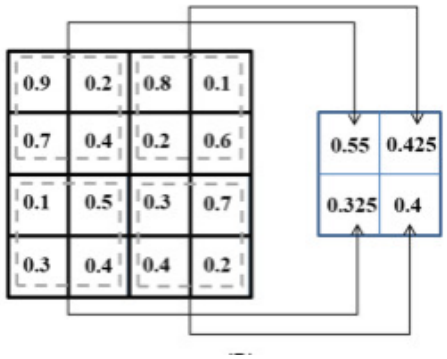
O *Max pooling*, basicamente retorna o maior valor durante o processo de filtro,

Figura 19 – Representação do processo de pooling usando o Max pooling



Fonte: Adaptado de Yamashita *et al.* (2018)

Figura 20 – Representação do processo de pooling usando o Average pooling



Fonte: Adaptado de Ranjbar *et al.* (2020)

descartando os menores valores como está representado na Figura 19 em um conjunto de dados $4X4$ reduzido em $2X2$ (YAMASHITA *et al.*, 2018).

O *Average Pooling* retorna a media dos valor durante o processo de filtro, como está representado na Figura 20.

2.3.3 Camada Totalmente Conectada

Geralmente a camada totalmente conectada é a última camada das redes neurais CNNs, localizada após a camada de *pooling* e pode ser seguida ou não de outras camadas totalmentes conectadas (RANJBAR *et al.*, 2020; SEWAK *et al.*, 2018).

Em uma CNNs de classificação, a última camada totalmente conectada resulta na probabilidade de cada classe utilizando as funções de ativação não-lineares como ReLU, e a Tangente hiperbólica (YAMASHITA *et al.*, 2018).

2.3.4 Dropout

O *Dropout* é um método de regularização utilizado quando o modelo funciona bem em conjuntos de dados de treinos, mas é ineficiente em dados de testes, sinônimo de *overfitting* (ALZUBAIDI *et al.*, 2021; SEWAK *et al.*, 2018). Este método escolhe aleatoriamente os neurônios que não serão utilizados a cada fase de treino, entretanto usados na fases de testes. (NANDINI *et al.*, 2021).

Um outro método muito utilizado que é parecido ao *Dropout* é o *Drop-Weights*, que ao invés de suprimir os neurônios, suprime os pesos deles, cortando assim a conexão entre eles (YAMASHITA *et al.*, 2018).

2.4 Máquinas de vector de suporte

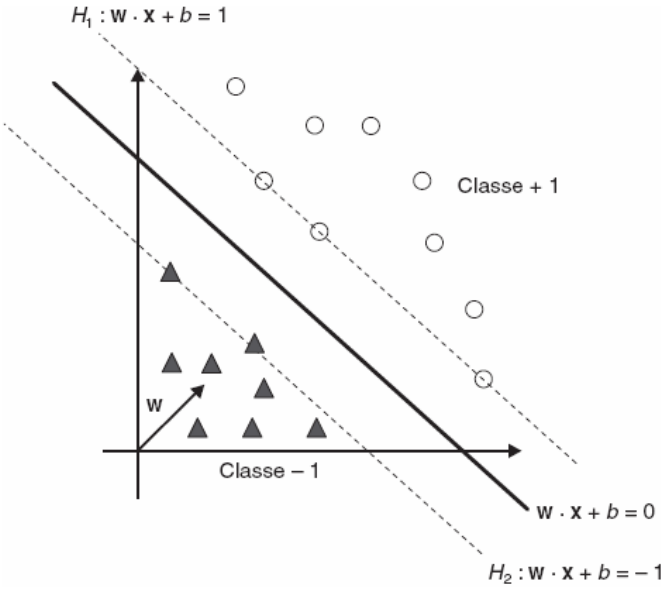
As Máquinas de Vector de Suportes (SVMs) são tipos de rede *feedforward* utilizadas geralmente para a classificação com dados linearmente separáveis, mas em alguns casos são também utilizadas para dados não linearmente separáveis fazendo uso de hiperplanos (HAYKIN, 2007). Considerando um conjunto de dados de entrada X e Y a saída representada pelo conjunto $\{-1,1\}$. Um hiperplano está definido pela equação 17 na qual $w.x$ é o produto escalar entre os vetores w e x, e b representa um número real. A Equação 17 pode ser utilizada para a divisão dos dados de entrada graças a equação 18 e está representada pela Figura 21 (HAYKIN, 2007). Na Figura 21 $H1$ representa a fronteira para a classe $+1$ e $H2$ representada a fronteira para a classe -1, essa fronteiras são chamadas de hiperplano canônico.

$$h(x) = w.x + b \tag{17}$$

$$y = \begin{cases} 1 \; se \; w.x + b > 0 \\ -1 \; se \; w.x + b < 0 \end{cases} \tag{18}$$

Os SVMs, para os dados não linearmente separáveis utilizam dois conceitos. O primeiro conceito chamada de *One vs Rest* ou seja um contra todos, esse conceito faz a separação binaria de n classes, assim cada classificador C_i é responsável por classificar a classe i das outras. Assim dado um novo valor x, esse novo valor pertencerá ao classificador que obteve o maior entre os n classificadores como está representada na Equação 19.

Figura 21 – Ilustração de hiperplanos canônicos e separador



Fonte: Adaptado de Faceli *et al.* (2021)

$$C(x) = \arg max_{1<i<n}(C_i(x)) \tag{19}$$

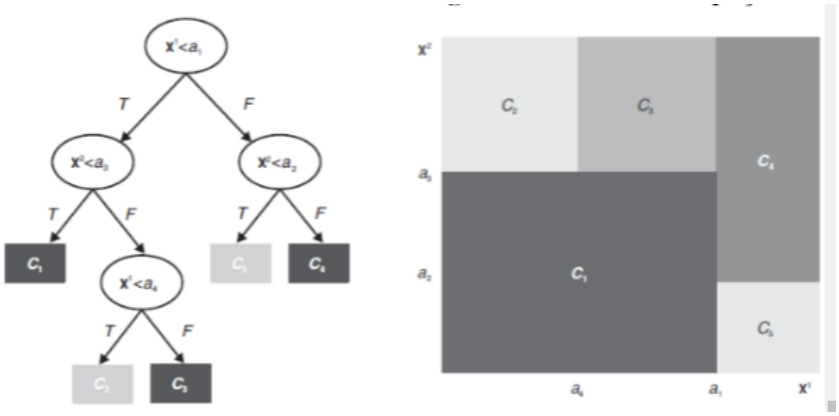
O segundo conceito é o todos contra todos, na qual o novo valor x pertence à classe com a maior quantidade de votos depois do sistema de votação (GONÇALVES, 2008).

2.5 Árvore de decisão

Uma árvore de decisão é um algoritmo de aprendizagem de máquina geralmente utilizada como método de classificação e de regressão, ela é um tipo de grafo direcionado que em cada nó gera duas ou mais nós folha. Cada ramo de um nó é definido por uma condição e as condições são testes que estão caracterizadas por um operador lógico como $>$, $<$, $=$... e um valor domínio de atributo. Na classificação o atributo é escolhido por uma regra de medida chamada *good of split*, que determina quão bem o atributo representa a classe como está ilustrado na Figura 22 (FACELI *et al.*, 2021).

O J48 é um dos algoritmo mais utilizado para o método de classificação e é baseado no C4.5 que é um algoritmo que necessita que os dados sejam numéricos e categóricos. O J48 utiliza vários métodos para a classificação dos dados, um dos

Figura 22 – Uma árvore de decisão e as regiões de decisão no espaço de objetos



Fonte: Adaptado de Faceli *et al.* (2021)

método consiste em possibilitar a substituição dos nós folhas com o nó principal, reduzindo assim a número de testes a serem realizados. Este método é efetuado em cinco etapas. Na primeira etapa é criado um nó principal para a árvore, na segunda etapa se todos os exemplos são positivas, então é retornado o valor "positivo"se não é retornado o valor "negativo". Na terceira etapa é calculado a entropia do estado atual. Na quarta etapa é calculado a entropia de cada atributo. Na quinta etapa é selecionado e removido o atributo com o maior *Info Gain* ou seja ganho de informação, e em seguida o processo é repetido até o agrupamento dos todos os dados (JEHAD *et al.*, 2020).

2.6 Trabalho Correlato

Existem poucos trabalhos relacionados a classificação baseados em microbioma de um organismo usando CNNs.

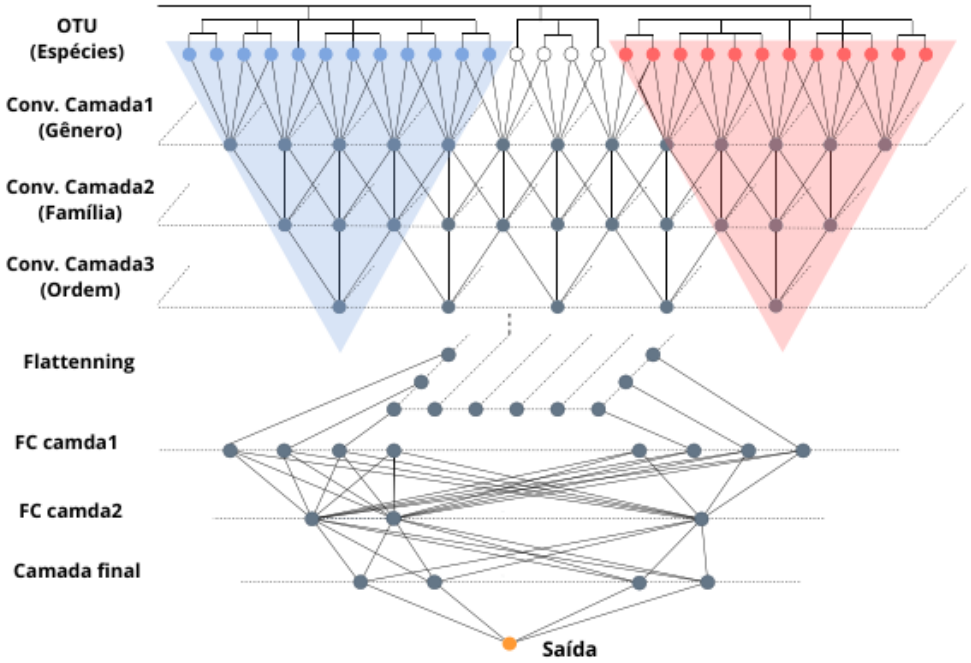
Wang *et al.* (2021) apresentam um método de classificação baseado em microbioma, na qual a rede neural desenvolvida *MDeep* foi comparada as várias outras redes existentes que fazem o mesmo.

O *Mdeep* é uma CNN regularizadora filogenética que recebe como dados de entrada os taxa ou unidades taxonômicas operacionais (OTUs), que são unidades de classificação taxonômicas de seres vivos como o reino, a ordem, o gênero ou ainda espécies. Os dados de entradas passam por várias camadas de convolução para determinar a correlação filogenética entre os taxa[1]. Por fim passam por várias

[1] Plural de taxón, é uma unidade taxónomica associada à classificação de seres vivos.

outras camadas completamente conectadas. O método de *dropout* é usado para evitar o *overfitting*. A analogia conceitual entre *MDeep* e os níveis taxonômicos da árvore filogenética está representada na Figura 23.

Figura 23 – Representação da analogia entre o MDeep e os níveis taxonômica



Fonte: Adaptado de Wang *et al.* (2021)

O *MDeep* foi treinado em conjuntos de dados reais para predição de microrganismos baseada em microbioma intestinal de gêmeos humanos onde ela superou várias redes concorrentes.

3 MATÉRIAS E MÉTODOS

Neste capítulo são apresentados a metodologia e o material necessário que foram aplicados para a classificação da saúde das amostras de corais coletados.

3.1 Métodos

Após a apresentação teórica sobre a importância do microbioma nos corais, as ANNs e alguns algoritmos de aprendizagem de máquina, neste capítulo é explanada a metodologia que foi aplicada para predição da saúde do coral baseado no microbioma do hospedeiro ANNs. A sequência da metodologia aplicada está representada na Figura 24.

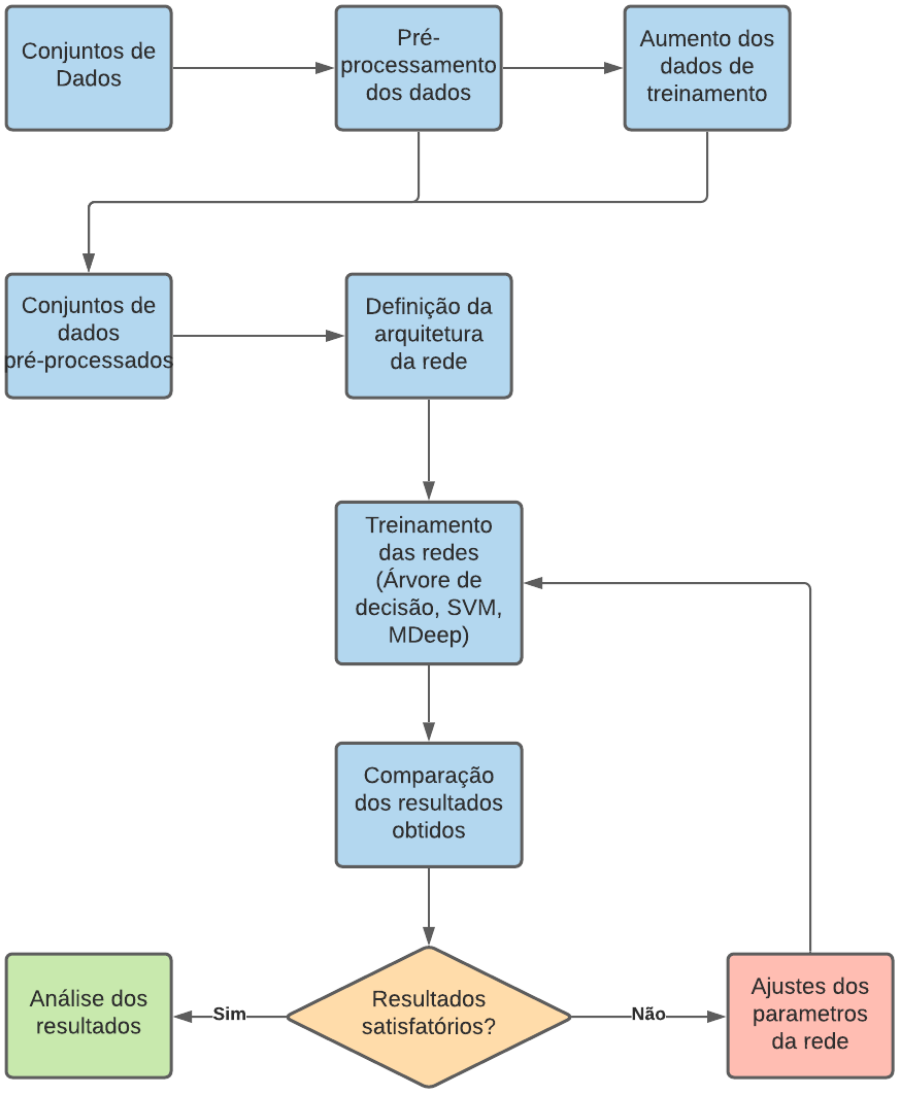
3.1.1 Conjuntos de dados

Este trabalho utiliza como conjuntos de dados, os dados referentes ao trabalho desenvolvido em Leite *et al.* (2018), compostos pelo coral *Mussismilia hispida* coletados em cinco recifes localizados próximos a uma área marinha protegida (“Parque Natural Municipal do Recife de Fora”) de Porto Seguro, Bahia, Brasil. O primeiro ponto da coleta está localizado aproximadamente a 2 km, o segundo a 4 km, o terceiro a 6 km, o quarto a 8 km e o quinto ponto a $9{,}4$ km da foz do rio Buranhém. Neste trabalho são utilizados principalmente os dados dos pontos 1, 3 e 5 por seus microbiomas terem sidos identificados.

Os micro-organismos presentes no coral foram identificados graças as técnicas de sequenciamento genético, e as análises de bioinformática subsequentes feitas no software QIIME 2.0 (BOLYEN *et al.*, 2019), previamente publicados por Leite *et al.* (2018).

Os dados disponibilizados em Leite *et al.* (2018) após análise de bioinformática no software QIIME 2.0, são compostos por uma tabela de abundância das bactérias presentes nas 68 amostras de coral coletadas em três pontos diferentes e em quatro estações do ano, como está representada na Tabela 1. A primeira coluna, é a coluna que referencia as amostras, na qual C seguido de um número representa a estação do ano; P seguido de um número representa o ponto onde foi coletada a amostra; AH ou

Figura 24 – Fluxograma de atividades



Fonte: Autoria própria (2021)

AG são relacionados às amostras consideradas não saudáveis e HH representa as amostras saudáveis. A segunda coluna e as demais são as colunas que identificam as bactérias e as quantidades em cada amostra. Além da Tabela de abundância está disponibilizado um arquivo *NWK* que contém a árvore filogenética rotulada dos 9.488 tipos de bactérias identificadas.

Tabela 1 – Tabela de abundância das bactérias

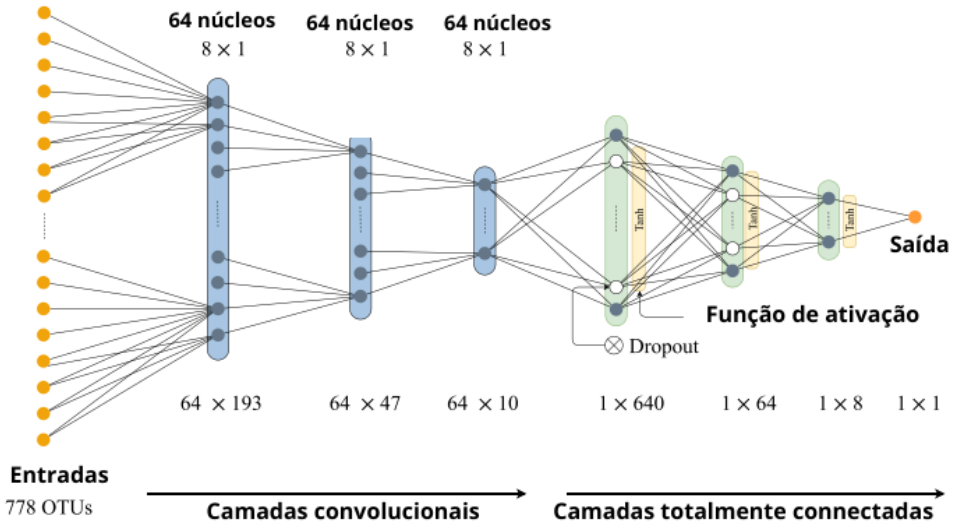
Amostras	4bccbdb96fb331b5bd8aec33cbb8a34e	c9b702e51c8e9a8d7235f376ae865078	c9b702e51c8e9a8d7235f376ae865078
C1P1AG1	1185	0	686
C1P1AG3	1335	0	3279
C1P1AH2	1490	0	1687
C1P1HH1	535	0	211
C1P1HH2	49	0	53

Fonte: Adaptado de Leite *et al.* (2018)

3.1.2 Arquitetura da Rede *Mdeep*

A arquitetura da *MDeep* desenvolvida em Wang *et al.* (2021) é uma rede essencial com regularização filogenética, constituída por três camadas convolucionais, três camadas completamente conectadas usando a função de ativação tangente hiperbólica e uma entrada que recebe os n OTUs. No *MDeep* foram utilizados 778 OTUs como dados de entrada equivalente a quantidade de neurônios presentes na camada de entrada da rede neural, no caso deste trabalho são utilizados 9.488 OTUs como dados de entrada. A arquitetura da rede neural utilizada para a classificação da saúde do coral, utilizando a tabela de abundância dos micro-organismos como dados de entrada está representada na Figura 25 e sua implementação disponível no *Github*[1].

Figura 25 – Representação da arquitetura de MDeep



Fonte: Adaptado de Wang *et al.* (2021)

3.1.3 Pré-processamentos

Antes da passagem dos dados no *Mdeep*, é gerado uma matriz de correlação baseada na distância entre dois taxa identificadas na árvore filogenética graças a função *cophenetic* do R[2]. A partir da tabela de abundância é necessário fazer inicialmente a substituição de cada identificação da amostra, 0 para as amostras não saudáveis e 1 para aquelas que estão saudáveis como está representada na Tabela 2; após esta

[1] https://github.com/lichen-lab/MDeep
[2] https://www.r-project.org/

etapa é feita a reorganização das colunas da tabela de abundância conforme a ordem das bactérias na matriz de correlação; por fim, a tabela de abundância é subdividida em três partes: uma primeira parte para o treinamento, a segunda parte para a validação e a terceira parte para os testes. Devido ao baixo número de amostras disponíveis, foi feito um aumento dos dados de treino segundo uma taxa de variação entre 1% e 10% em relação à quantidade de bactéria em cada amostra. A sequência do processamento dos dados até a classificação das amostras está representado na Figura 27.

A biblioteca *Scipy* não é capaz de gerar o dendrograma a partir da matriz de correlação composta por 9.488 linhas e 9.488 colunas devido ao problema de recursão. Para resolver o problema de recursividade foi substituída uma parte do código do arquivo *HAC.py* precisamente da linha 16 à linha 26 por um outro trecho de código como está representado na Figura 26. Além disso é feito a comparação entre o *MDeep* e alguns algoritmo clássicas de aprendizagem de máquina como a SMO e a árvore de decisão J48 utilizando a tabela de abundância.

Figura 26 – Código para geração do dendrograma

Fonte: Autoria própria (2021)

Tabela 2 – Tabela de abundância pré-processada

Amostras	4bccbdb96fb331b5bd8aec33cbb8a34e	c9b702e51c8e9a8d7235f376ae865078	c9b702e51c8e9a8d7235f376ae865078
0	1185	0	686
0	1335	0	3279
0	1490	0	1687
1	535	0	211
1	49	0	53

Fonte: Adaptado de Leite *et al.* (2018)

3.2 Ferramentas utilizadas

No pré-processamento de dados é utilizada a linguagem R que é uma linguagem multi-paradigma voltada a manipulação e análise de dados, o RStudio[3] na versão

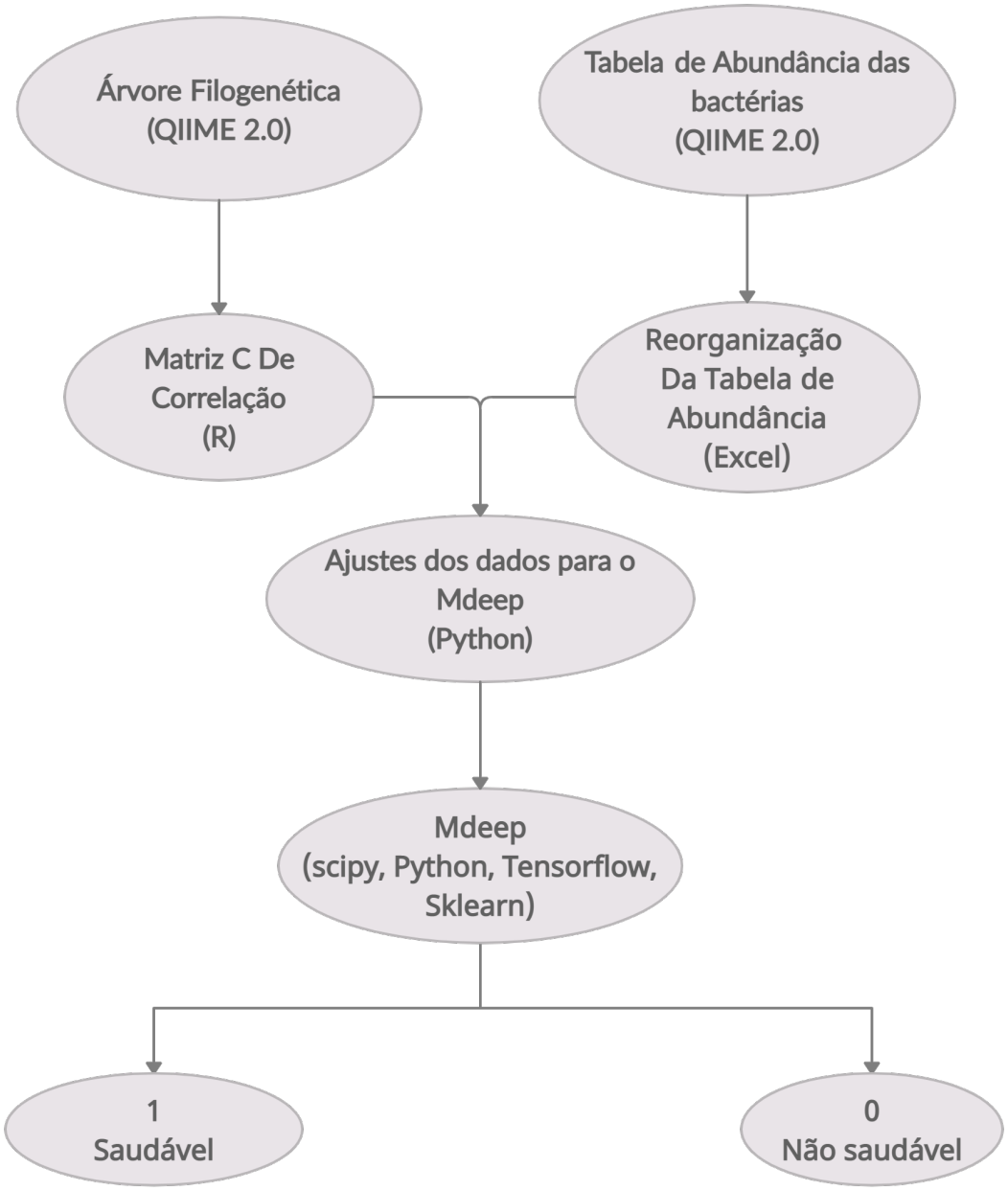
[3] https://www.rstudio.com/

4.1.0, que é um ambiente integrado para a linguagem R, que possui um console, editor de realce de sintaxe que suporta execução direta de código, além de ferramentas para plotagem, histórico, depuração e gerenciamento de espaço de trabalho.

Para a implementação da rede neural, é utilizada a linguagem Python[4] na versão 3.6 que é uma linguagem de alto nível, além de ser popular nas áreas como ciência de dados e engenharias, e indicada para a criação das ANNs. Além da linguagem Python, são utilizadas algumas bibliotecas necessárias para a implementação da rede como o TensorFlow[5] na versão 1.12.0, que é uma biblioteca de código aberto e oferece suporte para processamento com GPU com CUDA, o Scipy[6] na versão 1.2.1, que é uma biblioteca *open source* que serve para a ilustração dos agrupamentos dos micro-organismos gerando o dendrograma, O Scikit-learn[7] na versão 0.20.3, que é uma biblioteca utilizada para aprendizagem de máquina voltada a linguagem de programação Python e o Matplotlib na versão 3.1.0 que serve para a criação dos gráficos. Para a comparação da rede *Mdeep* com alguns algoritmo de aprendizgem de máquina como a SMO e a árvore de decisão J48 é utilizada a ferramenta Weka[8] que é uma coleção de algoritmos de aprendizado de máquina para tarefas de mineração de dados.

[4] https://www.python.org/
[5] https://www.tensorflow.org/
[6] https://www.scipy.org/
[7] https://scikit-learn.org/
[8] https://www.cs.waikato.ac.nz/ml/weka/

Figura 27 – Fluxograma dos dados



Fonte: Autoria própria (2021)

4 RESULTADOS E DISCUSSÃO

Neste capitulo, é explanado os experimentos assim com os resultados deste trabalho realizados com a árvore de decisão, a SMO e o *MDeep* utilizando em primeiro com os dados iniciais e em seguida os dados aumentados.

4.1 Experimentos com os dados iniciais

Nesta seção, foram utilizados os dados reais mencionados em Leite *et al.* (2018) composta por 68 amostras de coral e classificado a saúde das amostras utilizando a árvore de decisão, a SMO, e o *MDeep* variando seus parâmetros.

4.1.1 Experimentos com a árvore de decisão

Neste experimento com 68 amostras, primeiramente são carregados os dados da tabela de abundância no Weka, em seguida, é necessário fazer discretização na variavél alvo inserindo 1 na propriedade *attributeindices* e 2 na propriedade *bins*.

Executando o algoritmo J48 com o *Cross-Validation Folds* igual a 10 e *MinNumObj* igual a 10, o J48 obteve aproximadamente uma acurácia de $57{,}35\%$. Na matriz de confusão gerada na Tabela 3, pode ser notada que foram classificadas corretamente 5 amostras não saudáveis e incorretamente 18 amostras não saudáveis. Se tratando das amostras saudáveis, foram classificadas corretamente 34 amostras e incorretamente 11. A árvore gerada pela J48 está representada graficamente na Figura 28, na qual o identificador $a2e521c0c4cf25a6cb3408fad8a4198b$ representa a bactéria do gênero *Erwinia*, o identificador $d743eb845926f3b3dc117867a1b4892e$ representa a bactéria da espécie *Pseudomonas balearica*, *(0.5-inf)* determina a saúde do microbioma como saudável e *(-inf-0.5)* determina a saúde do microbioma como não saudável.

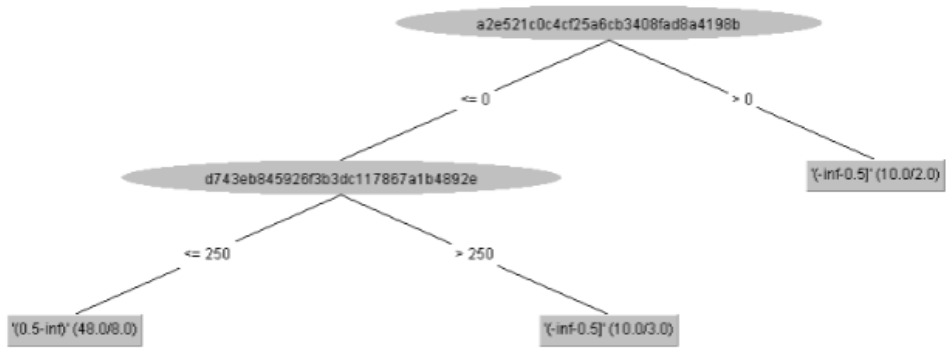
Tabela 3 – Resultados algoritmo J48

Matriz de confusão		
	Saudável	Não saudável
Saudável	5	18
Não Saudável	11	34

Fonte: Autoria própria (2021)

Aplicando a seleção de atributos utilizando a função *InfoGainAttributesE-*

Figura 28 – Arvore gerada pelo do algoritmo J48



Fonte: Autoria própria (2021)

val, foram reduzidos o número de atributos de 9.488 para 15, como está representado na Tabela 4. Executando mais uma vez o algoritmo, o J48 obteve aproximadamente uma acurácia de $70{,}58\%$, classificando corretamente 8 amostras não saudáveis e incorretamente 15 amostras; sobre as amostras saudáveis, foram classificadas corretamente 42 amostras e incorretamente 3 como está representada na Tabela 5. Na árvore gerada graficamente pela J48 na Figura 28, na qual o identificador $a2e521c0c4cf25a6cb3408fad8a4198b$ representa a bactéria do gênero *Erwinia*, o identificador $d743eb845926f3b3dc117867a1b4892e$ representa a bactéria da espécie *balearica*, e como resultado *(0.5-inf)* determina a saúde do microbioma como saudável e *(-inf-0.5)* determina a saúde do microbioma como não saudável. Na árvore gerada, pode ser notada que se houver presença da bactéria *Erwinia*, a árvore classifica a amostra como não saudável e se não houver a bactéria *Erwinia*, mas houver a presença da bactéria *Pseudomonas balearica* com uma quantidade maior que 250 a árvore classifica a amostra como não saudável também, caso esta bactéria obtiver uma quantidade menor ou igual a 250, a amostra é classificada como saudável.

Tabela 4 – Tabela dos 15 atributos selecionados

ID	Taxon
69dcb02812bac8ba6b70818e5d76ee75	k__Bacteria
7ba1f53a629f64d76aa6aa598aebb65d	k__Bacteria; p__Proteobacteria; c__Alphaproteobacteria; o__Rhizobiales; f__Hyphomicrobiaceae
c73470d06f32215bd6db496bb345d8ec	k__Bacteria; p__Proteobacteria; c__Alphaproteobacteria
d743eb845926f3b3dc117867a1b4892e	k__Bacteria; p__Proteobacteria; c__Gammaproteobacteria; o__Pseudomonadales; f__Pseudomonadaceae; g__Pseudomonas; s__balearica
175499458967f277f9f5d9d95b3fb750	k__Bacteria; p__Proteobacteria; c__Betaproteobacteria; o__Burkholderiales; f__Oxalobacteraceae; g__Ralstonia; s__
fa4034b1b3570b7a09c597b730570f03	k__Bacteria; p__Proteobacteria; c__Betaproteobacteria; o__Burkholderiales; f__Comamonadaceae
d808fb59707b01e9c7c0c24f792b8654	k__Bacteria; p__Proteobacteria; c__Gammaproteobacteria; o__Pseudomonadales; f__Pseudomonadaceae; g__Pseudomonas; s__
088eb09b6b8d6f1c98cd01ebea58c2e6	k__Bacteria; p__Firmicutes; c__Bacilli; o__Bacillales; f__[Exiguobacteraceae]; g__Exiguobacterium; s__
6570e98f6d39bacffe712be49da85a72	k__Bacteria; p__Proteobacteria; c__Gammaproteobacteria; o__Aeromonadales; f__Aeromonadaceae; g__; s__
102adec1a05d8f2be15e62ed6902edb7	k__Bacteria; p__Bacteroidetes; c__Flavobacteriia; o__Flavobacteriales; f__[Weeksellaceae]; g__Cloacibacterium; s__
a2e521c0c4cf25a6cb3408fad8a4198b	k__Bacteria; p__Proteobacteria; c__Gammaproteobacteria; o__Enterobacteriales; f__Enterobacteriaceae; g__Erwinia
4b92879a63fd26d78d05e2ea09fb2c6e	k__Bacteria; p__Actinobacteria; c__Actinobacteria; o__Actinomycetales; f__Mycobacteriaceae; g__Mycobacterium; s__
c4557feee1ae08c9987ec0b3ba074a04	k__Bacteria; p__Proteobacteria; c__Alphaproteobacteria; o__Rhizobiales; f__Aurantimonadaceae; g__; s__
da8dbb9356ed5e68256a9c9ec39bdbdc	k__Bacteria; p__Proteobacteria; c__Alphaproteobacteria; o__Sphingomonadales; f__Sphingomonadaceae; g__Sphingomonas; s__
eec75af11130febf1a4af7b2928274fc	k__Bacteria; p__Firmicutes; c__Bacilli; o__Lactobacillales; f__Lactobacillaceae; g__Lactobacillus; s__delbrueckii

Fonte: Autoria própria (2021)

Tabela 5 – Resultados do algoritmo J48

Matriz de confusão		
	Saudável	Não saudável
Saudável	8	15
Não Saudável	3	42

Fonte: Autoria própria (2021)

4.1.2 Experimentos com o SMO

Neste experimento com 68 amostras, primeiramente são carregados os dados da tabela de abundância no Weka, em seguido é necessário fazer discretização dos dados inserindo 1 na propriedade *attributeindices* e 2 na propriedade *bins*.

Executando o algoritmo SMO com o *Cross-Validation Folds* igual a 10, o SMO obteve uma acurácia de $66{,}17\%$. Graças a matriz de confusão gerada na Tabela 6, na qual pode ser notado que foram classificadas corretamente 7 amostras não saudáveis e incorretamente 16 amostras não saudáveis. No que se refere as amostras saudáveis foram classificadas corretamente 38 amostras e incorretamente 7.

Tabela 6 – Resultados do algoritmo SMO

Matriz de confusão		
	Saudável	Não saudável
Saudável	7	16
Não Saudável	7	38

Fonte: Autoria própria (2021)

Aplicando a seleção de atributos utilizando a função *InfoGainAttributesEval* foram reduzidos o número de atributos de 9.488 para 15, como está representado na Tabela 4. Executando mais uma vez o algoritmo, o SMO obteve uma acurácia de $72{,}05\%$, classificando corretamente 5 amostras não saudáveis e incorretamente 18 amostras; referente às amostras saudáveis foram classificadas corretamente 44 e incorretamente 1 amostra como está representada na Tabela 7.

Tabela 7 – Resultados do algoritmo SMO

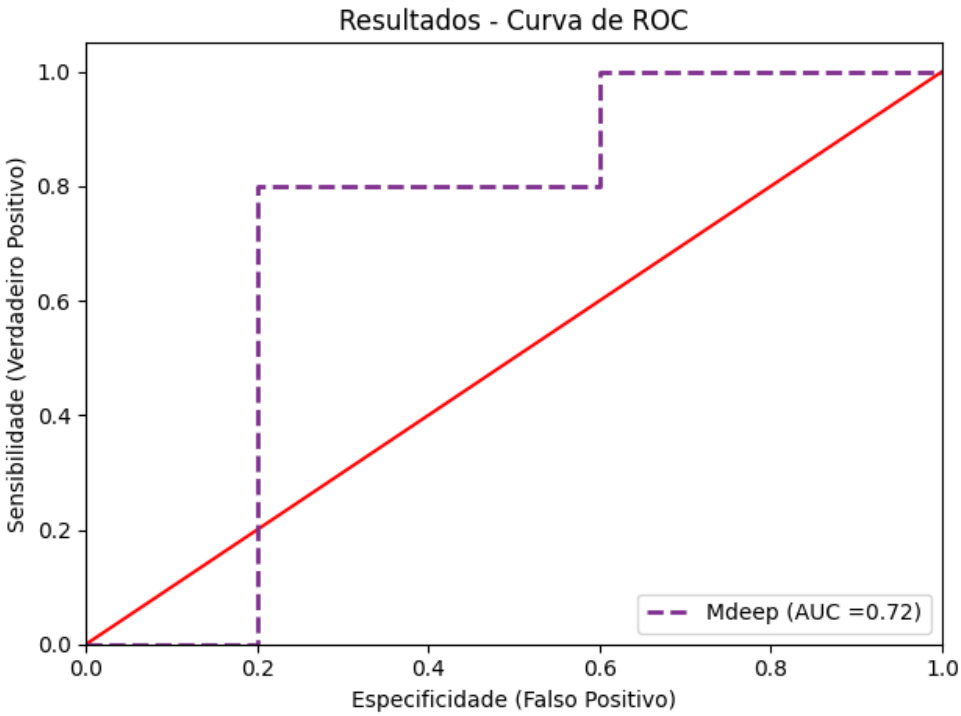
Matriz de confusão		
	Saudável	Não saudável
Saudável	5	18
Não Saudável	1	44

Fonte: Autoria própria (2021)

4.1.3 Experimentos com o Mdeep

Os dados iniciais são compostos por 68 amostras distribuídas da seguinte forma: 47 amostras para o treinamento, 10 amostras para a validação e 11 amostras para o os testes. Na fase de treinamento com 9.488 OTUs como dados de entrada, duas camadas convolucionais utilizando a função de ativação ReLU e uma camada totalmente conectada utilizando a tangente hiperbólica como função de ativação, o *Mdeep* obteve uma acurácia de $85{,}11\%$ ao longo de 500 épocas, com uma taxa de aprendizagem igual a 1^{-4}, uma taxa de *dropout* igual a $0{,}5$. Na fase de validação, o *Mdeep* obteve uma acurácia 72% ao longo de 500 épocas, com uma taxa de aprendizagem igual a 1^{-4}, uma taxa de *dropout* igual a $0{,}5$. O resultado da fase de validação pode ser visualizado no gráfico de *ROC* que é um gráfico obtido pela razão entre os verdadeiros positivos sobre os positivos totais e os falsos positivos sobre os negativos totais como está representado na Figura 29.

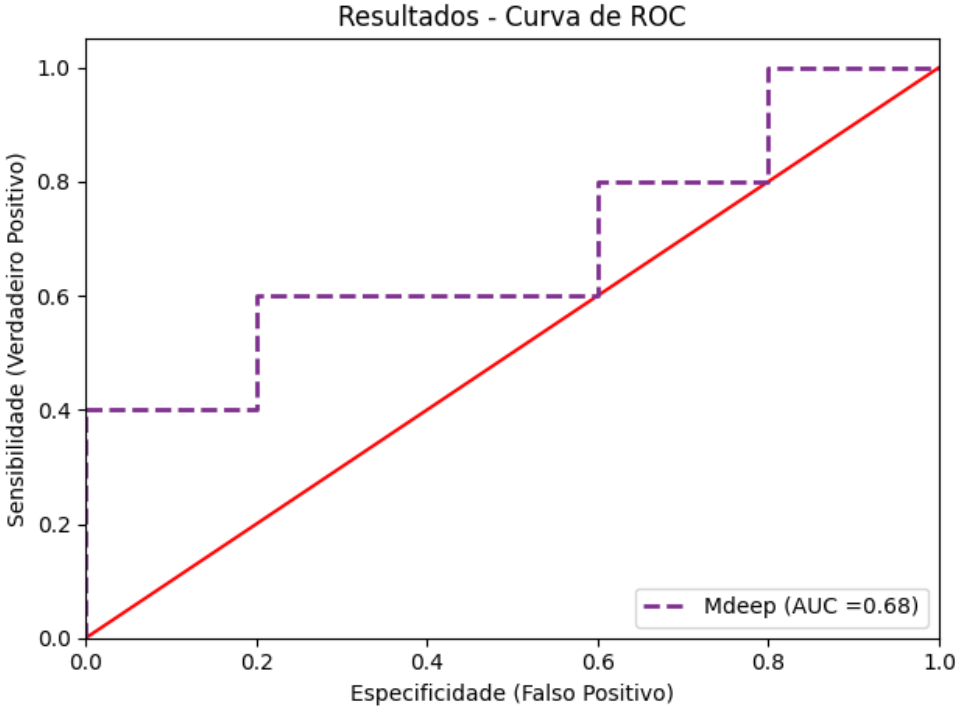
Figura 29 – Curva de ROC com dados de validação



Fonte: Autoria própria (2021)

Aplicando a seleção de atributos para 15, como está representado na Tabela 4, o *Mdeep* obteve uma acurácia de $91{,}49\%$ para os dados de treino e 68% com os dados de validação, como está representado na Figura 33.

Figura 30 – Curva de ROC com dados de validação com a seleção de atributos



Fonte: Autoria própria

4.2 Experimentos com os dados aumentados

Nesta seção foram utilizados os dados treinos aumentados para 5 vezes com um total de 256 amostras, com o propósito de melhorar o desempenho da árvore de decisão, a SMO e o *MDeep*.

4.2.1 Experimentos com a árvore de decisão

Com o aumento dos dados de treinos para 5 vezes neste experimento, as amostras passam de 68 a 256. Após o carregamento dos dados da tabela de abundância no Weka, é necessário fazer a discretização dos dados inserindo 1 na propriedade *attributeindices* e 2 na propriedade *bins*.

Executando o algoritmo J48 com o *Cross-Validation Folds* igual a 10 e *MinNumObj* igual a 15, o J48 obteve aproximadamente uma acurácia de $86{,}32\%$. Na matriz de confusão gerada na Tabela 8, na qual pode ser observado que foram classificadas corretamente 58 amostras não saudáveis e incorretamente 21 amostras não saudáveis. No caso das amostras saudáveis foram classificadas corretamente 163 amostras e incorretamente 14. A árvore ge-

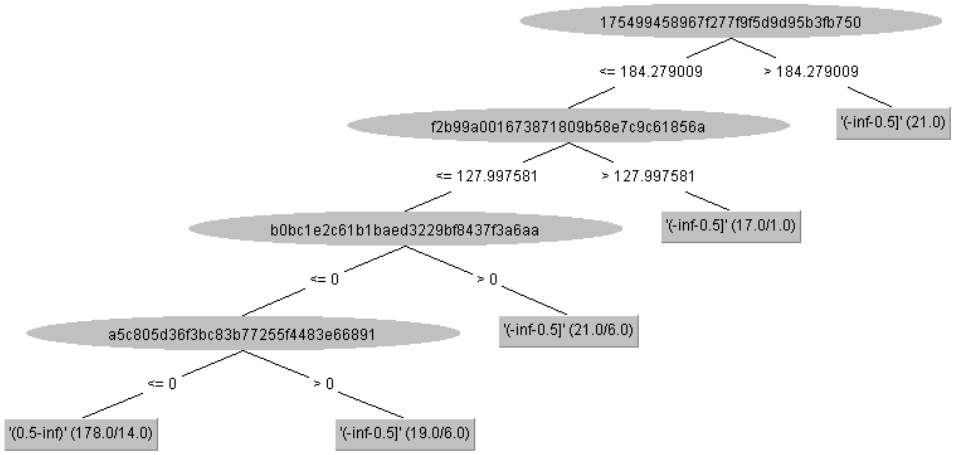
rada pelo algoritmo está representada na Figura 31, na qual o identificador $175499458967f277f9f5d9d95b3fb750$ representa a bactéria do gênero *Ralstonia*, o identificador $f2b99a001673871809b58e7c9c61856a$ representa a bactéria do gênero *Pseudomonas*, o identificador $b0bc1e2c61b1baed3229bf8437f3a6aa$ representa a bactéria da especie *Microbispora rosea*, o identificador $a5c805d36f3bc83b77255f4483e66891$ representa a bactéria da familia *Xenococcaceae*, e como resultado *(0.5-inf)* determina a saúde do microbioma como saudável e *(-inf-0.5)* determina a saúde do microbioma como não saudável. Na árvore gerada pode ser observado que se houver presença da bactéria *Ralstonia* com uma quantidade maior que $184{,}279009$, a amostra é classificada como não saudável se não houver uma quantidade menor ou igual a $184{,}279009$ e obtiver presença da bactéria *Pseudomonas* com uma quantidade maior que $127{,}99781$, a amostra é classificada como não saudável, se não houver presença da bactéria *Pseudomonas* com uma quantidade menor ou igual a $127{,}99781$ e obtiver presença da bactéria *rosea*, a amostra é classificada como não saudável, se não obtiver presença da bactéria *Xenococcaceae*, a amostra é classificada como não saudável, se não a amostra é classificada como saudável.

Tabela 8 – Resultados do algoritmo J48

Matriz de confusão		
	Saudável	Não saudável
Saudável	58	21
Não Saudável	14	163

Fonte: Autoria própria (2021)

Figura 31 – Árvore gerada pelo algoritmo J48



Fonte: Autoria própria (2021)

Aplicando a seleção de atributos utilizando a função *InfoGainAttributesEval*, e selecionando os 15 atributos mais relevantes, como está representado na Tabela 4. Após execução, o J48 obteve aproximadamente uma acurácia de 78%, classificando corretamente 42 amostras não saudáveis e incorretamente 37 amostras; no caso das amostras saudáveis foram classificadas corretamente 159 amostras e incorretamente 17 como está representada na Tabela 9.

Tabela 9 – Resultados do algoritmo J48

Matriz de confusão		
	Saudável	Não saudável
Saudável	42	37
Não Saudável	18	159

Fonte: Autoria própria (2021)

4.2.2 Experimentos com o SMO

Com o aumento dos dados de treinos para 5 vezes neste experimento, as amostras passam de 68 a 256. Sendo carregado primeiro os dados da tabela de abundância no Weka, após isso, é necessário fazer a filtragem utilizando a função de discretização dos dados inserindo 1 na propriedade *attributeindices* e 2 na propriedade *bins*.

Executando o algoritmo SMO com o *Cross-Validation Folds* igual a 10, o SMO obteve aproximadamente uma acurácia de 97.26%. Graças a matriz de confusão gerada na Tabela 10 na qual o termo saudável representa a amostra não saudáveis e o termo não saudável as amostras saudáveis, pode ser notado que foram classificadas corretamente 73 amostras não saudáveis e incorretamente 6 amostras não saudáveis. No caso das amostras saudáveis foram classificadas corretamente 176 amostras e incorretamente 1.

Tabela 10 – Resultados do algoritmo SMO

Matriz de confusão		
	Saudável	Não saudável
Saudável	73	6
Não Saudável	1	176

Fonte: Autoria própria (2021)

Aplicando a seleção de atributos utilizando a função *InfoGainAttributesEval*, e selecionando os 15 atributos mais relevantes, como está representado na Tabela 4.

Após execução o SMO obteve aproximadamente uma acurácia de $81{,}64\%$, classificando corretamente 42 amostras não saudáveis e incorretamente 37 amostras; em que se refere às amostras saudáveis foram classificadas corretamente 167 amostras e incorretamente 10 como está ilustrada na Tabela 11.

Tabela 11 – Resultados do algoritmo SMO

Matriz de confusão		
	Saudável	Não saudável
Saudável	42	37
Não Saudável	10	167

Fonte: Autoria própria (2021)

4.2.3 Experimentos com o Mdeep

Neste experimento foi feito o aumento dos dados de treinos para 5 vezes passando os dados de treinamento de 47 para 235, com uma variação determinada randomicamente entre 1% e 10% em relação à quantidade de cada bactéria presente na amostra, além de aleatoriamente definir se essa variação deveria ser positiva ou negativa.

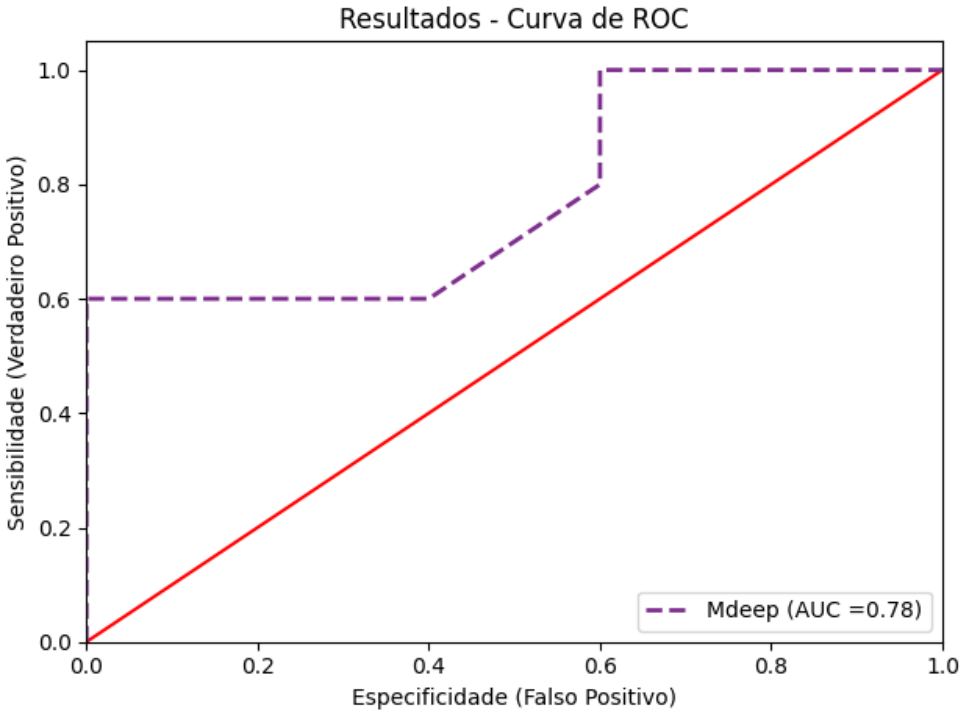
Com os dados aumentados, utilizando 9.488 OTUs como dados de entrada, duas camadas convolucionais utilizando a função de ativação RelU e uma camada totalmente conectada utilizando a tangente hiperbólica como função de ativação, o *Mdeep* obteve uma acurácia de $93{,}19\%$ na fase de treinamento ao longo de 500 épocas, com uma taxa de aprendizagem igual a 1^{-4}, uma taxa de *dropout* igual a $0{,}5$. Na fase de validação, o *Mdepp* obteve 78% ao longo de 500 épocas, com uma taxa de aprendizagem igual a 1^{-4}, uma taxa de *dropout* igual a $0{,}5$. O resultado para a fase de validação pode ser visualizado na Figura 32.

Aplicando a seleção de atributos para 15, como está representado na Tabela 4, o *Mdeep* obteve uma acurácia de $97{,}87\%$ para os dados de treino e 40% com os dados de validação, como está representado na Figura 33.

4.2.4 Análise dos experimentos

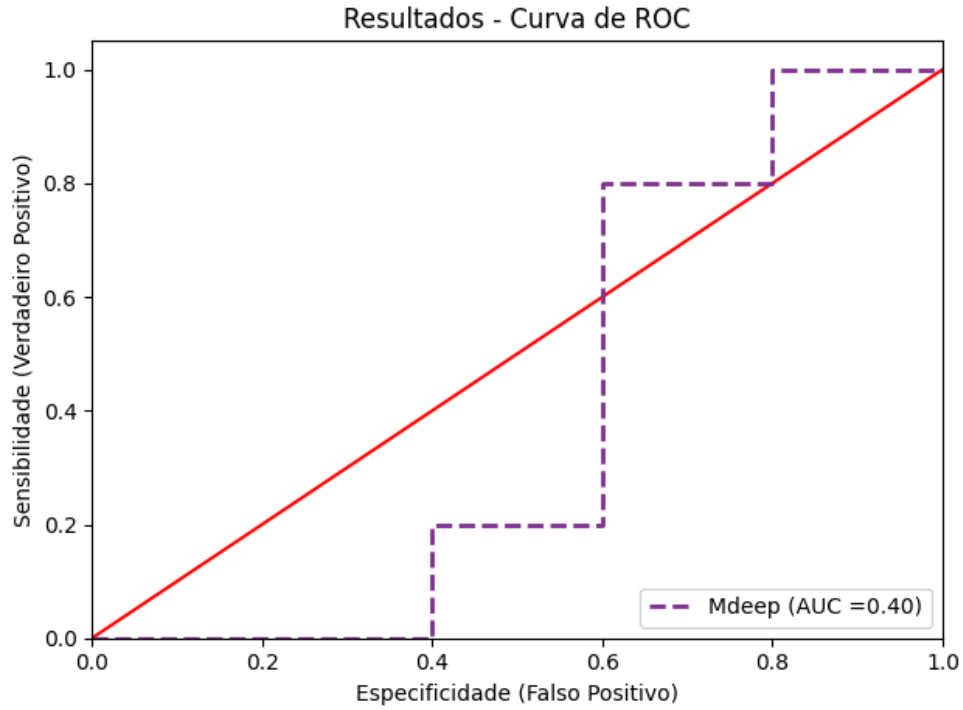
Analisando a Tabela 12 pode-se observar que, com os dados reais, o *MDeep* obteve uma acurácia de 72% com dados de validação, superando a rede SMO e J48

Figura 32 – Curva de ROC com dados de validação



Fonte: Autoria própria (2021)

Figura 33 – Curva de ROC com dados de validação com a seleção de atributos



Fonte: Autoria própria (2021)

da Weka que obtiveram respectivamente uma acurácia de $66{,}17\%$ e $57{,}35\%$. Aplicando a seleção dos atributos mais significativos para 15, favoreceu a classificação das

amostras pelo algoritmo SMO e J48 da Weka, obtendo respectivamente uma acurácia de $72{,}12\%$ e $70{,}58\%$, enquanto o *Mdeep* ficou desfavorecido com uma acurácia de 68%. Com os dados aumentados, o *MDeep* obteve uma acurácia de 78% com dados de validação, ou seja um aumento de 6%, enquanto a SMO obteve o melhor resultado com uma acurácia de $97{,}26\%$, ou seja um aumento de $31{,}09\%$ e a J48 obteve uma acurácia de $86{,}32\%$, ou seja um aumento de $28{,}97\%$. Aplicando a seleção dos atributos mais significativos para 15, o algoritmo SMO e J48, e o *Mdeep*, obtiveram respectivamente uma acurácia de 78%, $81{,}64\%$ e 40%, ou seja uma diminuição de respectiva de $8{,}32\%$, $15{,}62\%$, 28% em relação ao dados aumentada sem seleção de atributos.

Tabela 12 – Tabela comparativa entre o J48, o SMO e o Mdeep

	Resultados com dados reais (%)		Resultados com dados aumentados (%)	
Método	Sem seleção de atributos	Com seleção de atributos	Sem seleção de atributos	Com seleção de atributos
J48	57,35	70,58	86,32	78
SMO	66,17	**72,12**	**97,26**	**81,64**
Mdeep	**72,00**	68,00	78,00	40,00

Fonte: Autoria própria (2021)

A árvore gerada pela J48 identificou também algumas bactérias que foram identificada como bactérias chaves para a identificação da saúde do coral *Mussimilia Hispida* em Leite *et al.* (2018), como no caso da *Pseudomonas*, da *Pseudomonas balearica* e da *Ralstonia*.

5 CONSIDERAÇÕES FINAIS

Neste capítulo, são apesentados as conclusões sobre os resultados obtidos com os experimentos e a comparação dos métodos utilizados para a classificação do microbioma das amostras de corais, além de apresentar os possíveis trabalhos futuros.

5.1 Conclusão

Este trabalho possibilitou fazer a predição da saúde do coral baseado no seu microbioma utilizando uma rede neural convolucional, o *MDeep* e alguns algoritmo clássicas de aprendizagem de máquina como a SMO e J48 da Weka.

Com os dados reais, o *MDeep* obteve uma acurácia de $85{,}11\%$ com os dados de treino e uma acurácia aproximada a 72% com dados de validação, superando a rede SMO e J48 da Weka que obtiveram respectivamente uma acurácia de $66{,}17\%$ e $57{,}35\%$. Esta performance da *Mdeep* com os dados iniciais pode ser explicada pela sua exploração das informações filogenéticas e ao agrupamento dos OTUs com base na correlação filogenética antes do dados na camada de entrada. Aplicando a seleção dos atributos mais significativos para 15, favoreceu a classificação das amostras pelo algoritmo SMO e J48 da Weka, tendo respectivamente uma acurácia de $72{,}12\%$ e $70{,}58\%$. Neste primeiro experimento com os dados reais, a árvore de decisão conseguiu destacar duas bactérias (*Erwinia*, *balearica*) importantes para a classificação da saúde do coral.

No segundo experimento com os dados aumentados para 5 vezes, o *MDeep* teve uma acurácia de $93{,}19\%$ com os dados de treino e uma acurácia aproximada a 78% com dados de validação, ou seja um aumento de 6%, enquanto a SMO obteve uma acurácia de $97{,}26\%$, ou seja um aumento de $31{,}09\%$ e a J48 obteve uma acurácia de $86{,}32\%$, ou seja um aumento de $28{,}97\%$. Neste segundo experimento a árvore gerada pela J48 destacou quatro principais bactérias (*Ralstonia*, *Pseudomona*, *rosea*, *Xenococcaceae*) relacionadas à saúde do coral. Selecionando mais uma vez os 15 atributos mais significativos, a SMO e a J48 obtiveram uma diminuição na suas acurácias, certamente devida à limitação dos atributos.

Pode ser concluído segundo a acurácia obtida com o *MDeep* utilizando os dados reais que o *MDeep* consegue fazer a predição da saúde do coral baseando-se

na sua microbioma com dados reais e a um aumento de dados consideráveis por ser uma rede neural profunda por consequência necessita um número maior de dados. Os algoritmo clássica de aprendizagem de máquina como a SMO e a J48 são os melhores que se adaptaram ao aumento de dados.

5.2 Trabalhos Futuros

Uma próxima etapa interessante será obter mais dados e fazer um estudo aplicando o método de regressão oferecido pelo *Mdeep* se baseando na distâncias dos pontos da coleta das amostras, sabendo que quanto mais for distante o ponto de coleta da foz do rio Buranhém mais poluída é água ao redor do coral, além de acrescentar uma melhoria na rede *Mdeep* ou desenvolver uma rede com um mesmo propósito.

REFERÊNCIAS

ACHARYA, U. Rajendra; OH, Shu Lih; HAGIWARA, Yuki; TAN, Jen Hong; ADAM, Muhammad; GERTYCH, Arkadiusz; TAN, Ru San. A deep convolutional neural network model to classify heartbeats. **Computers in Biology and Medicine**, Elsevier Ltd, v. 89, p. 389–396, 2017. ISSN 18790534. Disponível em: http://dx.doi.org/10.1016/j.compbiomed.2017.08.022.

ALZUBAIDI, Laith; ZHANG, Jinglan; HUMAIDI, Amjad J.; AL-DUJAILI, Ayad; DUAN, Ye; AL-SHAMMA, Omran; SANTAMARÍA, J.; FADHEL, Mohammed A.; AL-AMIDIE, Muthana; FARHAN, Laith. **Review of deep learning: concepts, CNN architectures, challenges, applications, future directions**. Springer International Publishing, 2021. v. 8. ISSN 21961115. ISBN 4053702100444. Disponível em: https://doi.org/10.1186/s40537-021-00444-8.

AMTHOR, F. **Neuroscience For Dummies**. Wiley, 2016. (For dummies). ISBN 9781119224891. Disponível em: https://books.google.bj/books?id=WkjhCgAAQBAJ.

AQUASYMBIO. Zoanthus sociatus. 2021. Disponível em: http://www.aquasymbio.fr/en/symbiodinium-pulchrorum-zoanthus-sociatus-1.

BARRÉ, Pierre; STÖVER, Ben C.; MÜLLER, Kai F.; STEINHAGE, Volker. Leafnet: A computer vision system for automatic plant species identification. **Ecological Informatics**, v. 40, p. 50–56, 2017. ISSN 1574-9541. Disponível em: https://www.sciencedirect.com/science/article/pii/S1574954116302515.

BOLYEN; JR, Rideout; MR, Dillon; NA ABNET CC, Al-Ghalith GA Bokulich. Reproducible, interactive, scalable and extensible microbiome data science using qiime 2. 2019. Disponível em: https://doi.org/10.1038/s41587-019-0209-9.

BURKE, L; REYTAR, K; SPALDING, M; PERRY, A; INSTITUTE, World resources. **Récifs coralliens en péril revisité - Synthèse à l'intention des décideurs**. *[S.l.: s.n.]*, 2012. 58 p. ISBN 9781569737873.

CATHARINE, Deborah; LEITE, De Assis. Microbioma de corais endêmicos brasileiros: aspectos sobre evolução e resiliência do holobionte. 2016.

COURTIAL; ALLEMAND; FURLA, Centre; SCIENTIFIQUE, Directeur; PAOLA, Furla; MARINE, Symbiose. Coraux : les ingénieurs des océans sont menacés. p. 1–10, 2021.

FACELI; CARVALHO; LORENA; GAMA; ALMEIDA, AGOSTINHO DE. **Inteligência artificial: uma abordagem de aprendizado de máquina**. Grupo GEN, 2021. ISBN 9788521637509. Disponível em: https://integrada.minhabiblioteca.com.br/books/9788521637509.

FURTADO, Maria Inês Vasconcellos. **Redes Neurais Artificiais: Uma Abordagem Para Sala de Aula**. *[S.l.: s.n.]*, 2019. ISBN 9788572473262.

GODOY, Leandro; GARRIDO, Amana; PEREIRA, Cristiano; ZILBERBERG, Carla; DE, Débora. O mundo peculiar dos corais : ciclo de vida e formação de recifes. p. 22–27, 2020.

GONÇALVES, André R. Fundamentos e Aplicações de Técnicas de Aprendizado de Máquina. 2008. Disponível em: https://andreric.github.io/posts/2018/05/blog-post-2/.

GUPTA, Anvita; MÜLLER, Alex T.; HUISMAN, Berend J.H.; FUCHS, Jens A.; SCHNEIDER, Petra; SCHNEIDER, Gisbert. Generative Recurrent Networks for De Novo Drug Design. **Molecular Informatics**, v. 37, n. 1, 2018. ISSN 18681751.

GURNEY, Kevin. **An Introduction to Neural Networks**. USA: Taylor Francis, Inc., 1997. ISBN 1857286731.

HAYKIN, S. **Redes Neurais: Princípios e Prática**. Artmed, 2007. ISBN 9788577800865. Disponível em: https://books.google.com.br/books?id=bhMwDwAAQBAJ.

JAIN, Aarshay. Fundamentals of deep learning – starting with artificial neural network. 2016. Disponível em: https://www.analyticsvidhya.com/blog/2016/03/introduction-deep-learning-fundamentals-neural-networks/.

JAUBERT. **Les récifs coralliens**. 2019. 275–297 p.

JEHAD; REHAM, Yousif; SUHAD. Fake News Classification Using Random Forest and Decision Tree (J48). **Al-Nahrain Journal of Science**, v. 23, n. 4, p. 49–55, 2020. ISSN 26635453.

JUGANT, Sophie. Importance des récifs coralliens pour les poissons récifaux: exemple des Demoiselles (Pomacentridae), dans l'archipel des Maldives. 2012. Disponível em: http://oatao.univ-toulouse.fr/8644/.

LEITE, Deborah C. A.; SALLES, Joana F.; CALDERON, Emiliano N.; CASTRO, Clovis B.; BIANCHINI, Adalto; MARQUES, Joseane A.; ELSAS, Jan Dirk van; PEIXOTO, Raquel S. Coral bacterial-core abundance and network complexity as proxies for anthropogenic pollution. **Frontiers in Microbiology**, v. 9, p. 833, 2018. ISSN 1664-302X. Disponível em: https://www.frontiersin.org/article/10.3389/fmicb.2018.00833.

LEONARD; FABER, Tuteur André. Travail personnel 2018-2019. 2019.

LIU, Weibo; WANG, Zidong; LIU, Xiaohui; ZENG, Nianyin; LIU, Yurong; ALSAADI, Fuad E. A survey of deep neural network architectures and their applications. **Neurocomputing**, Elsevier B.V., v. 234, n. November 2016, p. 11–26, 2017. ISSN 18728286. Disponível em: http://dx.doi.org/10.1016/j.neucom.2016.12.038.

MAGALHÃES, Camila S. de; BARBOSA, Hélio J.C.; DARDENNE, Laurent E. A genetic algorithm for the ligand-protein docking problem. 2004. Disponível em: https://doi.org/10.1590/S1415-47572004000400022.

NANDINI, Gangi Siva; KUMAR, A.P. Siva; K, Chidananda. Dropout technique for image classification based on extreme learning machine. **Global Transitions Proceedings**, v. 2, n. 1, p. 111–116, 2021. ISSN 2666-285X. 1st International Conference on Advances in Information, Computing and Trends in Data Engineering (AICDE - 2020). Disponível em: https://www.sciencedirect.com/science/article/pii/S2666285X21000157.

PUISAY, Antoine. La reproduction sexuée et asexuée des coraux face aux changements environnementaux : Implications pour la conservation et la restauration des récifs coralliens Antoine Puisay HAL Id : tel-02906023. n. July 2018, 2018.

RANJBAR, Sajad; NEJAD, Fereidoon Moghadas; ZAKERI, Hamzeh; GANDOMI, Amir H. 3 - computational intelligence for modeling of asphalt pavement surface distress. *In*: SAMUI, Pijush; KIM, Dookie; IYER, Nagesh R.; CHAUDHARY, Sandeep (Ed.). **New Materials in Civil Engineering**. Butterworth-Heinemann, 2020. p. 79–116. ISBN 978-0-12-818961-0. Disponível em: https://www.sciencedirect.com/science/article/pii/B978012818961000003X.

REGIER, Yvonne; KOMMA, Kassandra; WEIGEL, Markus; KRAICZY, Peter; LAISI, Arttu; PULLIAINEN, Arto T.; HAIN, Torsten; KEMPF, Volkhard A.J. Combination of microbiome analysis and serodiagnostics to assess the risk of pathogen transmission by ticks to humans and animals in central Germany 11 Medical and Health Sciences 1108 Medical Microbiology. **Parasites and Vectors**, Parasites Vectors, v. 12, n. 1, p. 1–17, 2019. ISSN 17563305.

RIYUZO, RAQUEL. Análise de microbioma a partir de sequências de 16sr rna: Asv ou otu? 2020. Disponível em: https://blog.varsomics.com/analise-de-microbioma-a-partir-de-sequencias-de-16sr-rna-asv-ou-otu/.

ROJAS, Raúl. **Neural Networks: A Systematic Introduction**. Berlin, Heidelberg: Springer-Verlag, 1996. ISBN 3540605053.

RUSSELL, Stuart; NORVIG, Peter. **Artificial Intelligence: A Modern Approach**. 3. ed. *[S.l.]*: Prentice Hall, 2010.

SEWAK, Mohit; KARIM, Rezaul; PUJARI, Pradeep. **Practical Convolutional Neural Networks**. *[S.l.: s.n.]*, 2018. 199 p. ISBN 9781788392303.

SHARMA, Siddharth; SHARMA, Simone; ATHAIYA, Anidhya. Activation Functions in Neural Networks. **International Journal of Engineering Applied Sciences and Technology**, v. 04, n. 12, p. 310–316, 2020. ISSN 2455-2143.

SHEN, Guohua; HORIKAWA, Tomoyasu; MAJIMA, Kei; KAMITANI, Yukiyasu. Deep image reconstruction from human brain activity. **PLOS Computational Biology**, Public Library of Science, v. 15, n. 1, p. 1–23, 01 2019. Disponível em: https://doi.org/10.1371/journal.pcbi.1006633.

SILVA, I. N.; SPATTI, D. H.; FLAUZINO, R. A. Redes Neurais Artificiais Para Engenharia e Ciências Aplicadas. **São Paulo: Artliber**, 2010.

THOMPSON, Fabiano; THOMPSON, Cristiane. Biotecnologia marinha. p. 855, 2020.

VERON, Charlie (J.E.N.). Mussismilia hispida. 2021. Disponível em: https://www.sealifebase.ca/summary/Mussismilia-hispida.html.

WANG, Ye; BHATTACHARYA, Tathagata; JIANG, Yuchao; QIN, Xiao; WANG, Yue; LIU, Yunlong; SAYKIN, Andrew J.; CHEN, Li. A novel deep learning method for predictive modeling of microbiome data. **Briefings in bioinformatics**, v. 22, n. 3, p. 1–14, 2021. ISSN 14774054.

WIKISTAT. Neural Networks and Introduction to Deep Learning. p. 1–17, 2015. Disponível em: http://klab.tch.harvard.edu/academia/classes/BAI/pdfs/intro-deep-learning.pdf.

YAMASHITA, Rikiya; NISHIO, Mizuho; Kinh Gian Do, Richard; TOGASHI, Kaori. Convolutional neural networks: an overview and application in radiology. **Smart Innovation, Systems and Technologies**, Insights into Imaging, v. 195, p. 21–30, 2018. ISSN 21903026.

ZILBERBERG, Carla; ABRANTES, Douglas Pinto; MARQUES, Joseane Aparecida; MACHADO, Laís Feitosa; MARANGONI, Laura Fernandes de Barros. **Conhecendo os Recifes Brasileiros: Rede de Pesquisas Coral Vivo**. *[s.n.]*, 2016. 360 p. p. ISBN 978-85-7427-057-9. Disponível em: http://coralvivo.org.br/arquivos/documentos/Livro-Zilberberg-et-al-2016-Conhecendo-os-Recifes-Brasileiros-Rede-de-Pesquisas-Coral-Vivo.pdf%0Ahttp://coralvivo.org.br/wp-content/uploads/arquivos/2308file-3.pdf.