



MONITORAMENTO EM LINHA BASEADO EM AI E ESPECTROSCOPIA DA CONCENTRAÇÃO ENANTIOMÉRICA DE PRAZIQUANTEL

MARLON CONCEIÇÃO DOS SANTOS

Monografia em Engenharia Química

Orientadores:

Maurício Bezerra de Souza Júnior, D.Sc.

Argimiro Resende Secchi, D.Sc

Felipe Coelho Cunha, D.Sc

Dezembro de 2022

**MONITORAMENTO EM LINHA BASEADO EM AI E ESPECTROSCOPIA DA
CONCENTRAÇÃO ENANTIOMÉRICA DE PRAZIQUANTEL**

Marlon Conceição dos Santos

Monografia em Engenharia Química submetida ao Corpo Docente da Escola de Química, como parte dos requisitos necessários à obtenção do grau de Engenheiro Químico.

Aprovado por:

Idelfonso Bessa dos Reis Nogueira, D.Sc., NTNU

Amaro Gomes Barreto Junior, D.Sc., UFRJ

Orientado por:

Maurício Bezerra de Souza Júnior, D.Sc., DEQ/EQ/UFRJ

Argimiro Resende Secchi, D.Sc., PEQ/COPPE/UFRJ

Felipe Coelho Cunha, D.Sc, EPQB/EQ/UFRJ

Rio de Janeiro, RJ – Brasil

Dezembro de 2022

S237m Santos, Marlon Conceição dos Monitoramento em linha baseado em AI e espectroscopia da concentração enantiomérica de praziquantel / Marlon Conceição dos Santos. -- Rio de Janeiro, 2022.

77 f.

Orientador: Maurício Bezerra de Souza Júnior.

Coorientadores: Argimiro Resende Secchi & Felipe Coelho Cunha.

Trabalho de conclusão de curso (graduação) - Universidade Federal do Rio de Janeiro, Escola de Química, Bacharel em Engenharia Química, 2022.

1. Machine Learning. 2. Redes Neurais. 3. Dados experimentais. 4. Leito Móvel Simulado. 5. Praziquantel. I. Souza Júnior, Maurício Bezerra de, orient. II. Secchi, Argimiro Resende & Cunha, Felipe Coelho, coorient. III. Título.

AGRADECIMENTOS

Gostaria de agradecer a Deus por sempre ter me ajudado. Agradecer também à minha família por todo apoio. Aos meus orientadores e coorientadores pelo auxílio ao longo do desenvolvimento deste trabalho e aos professores por serem essenciais na minha formação profissional.

Resumo da Monografia apresentada à Escola de Química como parte dos requisitos necessários para obtenção do grau de Engenheiro Químico.

MONITORAMENTO EM LINHA BASEADO EM AI E ESPECTROSCOPIA DA CONCENTRAÇÃO ENANTIOMÉRICA DE PRAZIQUANTEL

Marlon Conceição dos Santos

Dezembro, 2022

Orientadores: Prof. Maurício Bezerra de Souza Júnior, D.Sc.

Prof. Argimiro Resende Secchi, D.Sc

Felipe Coelho Cunha, D.Sc

A esquistossomose é uma doença tropical negligenciada, cujo tratamento é feito por meio do medicamento praziquantel. O tratamento para crianças é feito pelo medicamento em forma líquida, que apresenta amargor, devido à presença do enantiômero dextrógiro D-PZQ (ingrediente inativo). Assim, durante a formulação é necessário um enriquecimento do componente levógiro da mistura racêmica, ou seja, o L-PZQ, a partir da separação de tais componentes. O Leito Móvel Simulado (LMS) pode ser utilizado para a separação de enantiômeros. Para o controle automático do LMS, é necessário o monitoramento das concentrações dos enantiômeros. Em estudos prévios na Escola de Química da UFRJ, foi construído um LMS, o qual possui um detector quiral, que mede sinais de UV e dicroísmo circular simultaneamente. Neste trabalho, são desenvolvidos modelos matemáticos baseados em aprendizado de máquina para correlacionar os sinais de UV e dicroísmo circular com as concentrações de cada enantiômero no LMS. Optou-se pela utilização das redes neuronais devido à relação não linear previamente observada entre as concentrações e medições. O desenvolvimento da rede foi feito em Python, utilizando o MLPRegressor, um estimador que se baseia nas redes do tipo “Perceptron Multi-camada”. Devido à ampla gama de possibilidades para o projeto da rede, foi necessário resolver um problema de otimização, buscando o melhor conjunto de hiperparâmetros para a predição da concentração dos enantiômeros. Para isso, foi utilizado o software Optuna. Como forma de validação foram feitas comparações com algoritmo “Random Forest”, diferentes abordagens de pré-processamento, usando como critério de desempenho o coeficiente de determinação (R^2) e a abordagem polinomial desenvolvida no trabalho de CUNHA et al (2020). Com base em todos os dados experimentais (57 pontos) e validação (18 pontos), as redes neuronais apresentaram um R^2 de 0,82 e 0,83. No caso do “Random Forest”, o coeficiente foi de 0,82 e 0,66 e, no caso da abordagem polinomial, 0,54 e 0,59. Dessa forma, os resultados habilitam o uso das redes para monitorar o processo, pela boa capacidade preditiva verificada, a qual pode ainda ser melhorada com a disponibilidade de mais dados.

ÍNDICE

1. INTRODUÇÃO	1
2. FUNDAMENTOS E REVISÃO BIBLIOGRÁFICA	5
2.1. PRAZIQUEL	5
2.2. LEITO MÓVEL SIMULADO	6
2.3. O DICROÍSMO CIRCULAR E A INTERPRETAÇÃO DE SEUS SINAIS	9
2.4. FUNDAMENTOS DE APRENDIZADO DE MÁQUINA E REDES NEURONAIS	10
2.4.1. APRENDIZADO DE MÁQUINA	10
2.4.2. REDES NEURONAIS	11
2.4.3. RANDOM FOREST	19
2.5. REVISÃO DA LITERATURA	20
2.6. CONCLUSÕES DA REVISÃO DA LITERATURA	24
3. MATERIAIS E MÉTODOS	25
3.1. DADOS	25
3.2. REDES NEURONAIS	27
3.3. COMPARAÇÃO DE REDES COM OUTRAS ABORDAGENS	32
4. RESULTADOS	35
4.1. DADOS	35
4.2. TREINAMENTO E VALIDAÇÃO	36
4.3. COMPARAÇÃO COM OUTROS ALGORITMOS	46
4.4. CONCLUSÕES DO CAPÍTULO	48
5. CONCLUSÕES E SUGESTÕES	52
6. REFERÊNCIAS	55
APÊNDICE A	61
APÊNDICE B	63
MLPREGRESSOR x STANDARDSCALER X INDEPENDENTE:	63
MLPREGRESSOR X MINMAX X IDEPENDENTE:.....	64
MLPREGRESSOR X MINMAX X PADRÃO:	64
RANDOM FOREST X STANDARDSCALER X PADRÃO:	65
RANDOM FOREST X STANDARD SCALER X INDEPENDENTE:.....	65
RANDOM FOREST X MINMAX X INDEPENDENTE:.....	66
RANDOM FOREST X MINMAX X PADRÃO:	67

ÍNDICE DE FIGURAS

Figura 2-1 - Estrutura do praziquantel.....	6
Figura 2-2 - Ilustração de um leito móvel verdadeiro.....	7
Figura 2-3 - Estrutura de uma unidade LMS.....	8
Figura 2-4 - Posição das redes neuronais e do aprendizado de máquina dentro da inteligência artificial.....	11
Figura 2-5 - Exemplo de organização de uma rede neuronal do tipo feedforward.....	13
Figura 2-6 - Ativação do neurônio artificial.....	14
Figura 2-7 - Exemplo de rede feedforward com três camadas.....	17
Figura 2-8 - Algoritmo Random Forest.....	20
Figura 3-1 - Unidade LMS.....	26
Figura 3-2 - Exemplo de validação cruzada.....	31
Figura 3-3 - Descrição da otimização utilizando validação cruzada e obtenção da rede ótima.....	32
Figura 4-1 - Conjunto completo de dados experimentais.....	35
Figura 4-2 - Dados experimentais utilizados no modelo de CUNHA et al, 2020.....	36
Figura 4-3 - Evolução da variável “Intermediate” em função do “step”.....	40
Figura 4-4 - Evolução do valor ótimo.....	41
Figura 4-5 – Coordenadas paralelas.....	42
Figura 4-6 - Importância de cada hiperparâmetro para o valor da função objetivo.....	43
Figura 4-7 - Gráfico de fatia.....	44
Figura 4-8 - Ilustração da rede neuronal obtida.....	45
Figura 4-9 - Gráficos comparativos com todos os dados experimentais.....	51
Figura 4-10 - Gráficos comparativos com os dados de validação.....	51

ÍNDICE DE TABELAS

Tabela 3-1 - Volume útil das zonas da unidade LMS.	27
Tabela 3-2 - Descrição dos parâmetros de maior impacto na rede.	28
Tabela 3-3 - Possibilidades para os parâmetros individuais.	29
Tabela 3-4 - Parâmetros do Random Forest.	33
Tabela 3-5 - Resumo da metodologia.	34
Tabela 4-1 - Variáveis de entrada manual.	38
Tabela 4-2 - Resumo da rotina de otimização via Optuna e validação cruzada.	39
Tabela 4-3 - Conjunto ótimo de hiperparâmetros.	44
Tabela 4-4 - Pesos da rede otimizada.	44
Tabela 4-5 - Bias da rede otimizada.	45
Tabela 4-6 - Avaliação do desempenho do "Transfer Learn" com os dados experimentais.	46
Tabela 4-7 - Diferentes abordagens utilizadas para a obtenção do estimador.	46
Tabela 4-8 - Resultados para as redes do tipo "Padrão" (duas saídas em conjunto).	47
Tabela 4-9 - Resultados para as redes do tipo "Independente".	47
Tabela 4-10 - Resultados da rede otimizada diretamente com os dados experimentais.	48
Tabela 4-11 - Comparação com os resultados do modelo de CUNHA,2021.	48
Tabela 4-12 - Resumo dos resultados obtidos.	50
Tabela A-1 - Dados experimentais utilizados para treinamento e validação da rede.	61
Tabela B-1 - Conjunto de hiperparâmetros da rede otimizada do estimador A.	63
Tabela B-2 – "Scores" do estimador A com os dados experimentais.	63
Tabela B-3 - Conjunto de hiperparâmetros da otimização para o estimador B.	63
Tabela B-4 – "Scores" do estimador B com os dados experimentais.	63
Tabela B-5 - Conjunto ótimo de parâmetros do estimador A.	64
Tabela B-6 – Score do estimador A com os dados experimentais	64
Tabela B-7 - Conjunto ótimo de parâmetros para o estimador B.	64
Tabela B-8 – "Scores" do estimador B para os dados experimentais.	64
Tabela B-9 - Conjunto ótimo de parâmetros do estimador.	65
Tabela B-10 – "Scores" com os dados experimentais.	65
Tabela B-11 - Conjunto de hiperparâmetros obtidos da otimização.	65
Tabela B-12 – "Scores" dados experimentais.	65
Tabela B-13 - Conjunto de hiperparâmetros obtidos da otimização estimador A.	66
Tabela B-14 – "Scores" dados experimentais estimador A.	66
Tabela B-15 - Conjunto de hiperparâmetros obtidos da otimização estimador B.	66
Tabela B-16 – "Scores" dados experimentais estimador B.	66
Tabela B-17 - Conjunto de hiperparâmetros obtidos da otimização estimador B.	66
Tabela B-18 – "Scores" dados experimentais estimador A.	67
Tabela B-19 - Conjunto de hiperparâmetros obtidos da otimização estimador B.	67
Tabela B-20 – "Scores" dados experimentais estimador B.	67
Tabela B-21 - Conjunto de hiperparâmetros obtidos da otimização.	67
Tabela B-22 – "Scores" dados experimentais.	67

NOMENCLATURA

B	Vetor com biases de cada camada
CD	Dicroísmo Circular
D-PZQ	Enantiômero dextrogiro do praziquantel
E	Vetor erro
HPLC	Cromatografia Líquida de Alta Eficiência
LADES	Laboratório de Desenvolvimento de Software para Controle e Otimização de Processos
LMS	Leito Móvel Simulado
LMV	Leito Móvel Verdadeiro
L-PZQ	Enantiômero levogiro do praziquantel
MLP	Perceptron Multicamada
PZQ	Praziquantel
R ²	Coefficiente de determinação
STL	Sem Transfer Learn
TL	Transfer Learn
UV-vis	Ultravioleta - visível
W	Vetor de pesos
α	Parâmetro do modelo polinomial
β	Paramêtro do modelo polinomial
ε	Momento
η	Taxa de aprendizagem
μ	Média
σ	Desvio padrão populacional
\overline{A}_D	Luz polarizada circularmente para a direita (ultravioleta-visível)
\overline{A}_L	Luz polarizada circularmente para a esquerda (ultravioleta-visível)
\hat{C}_D	Concentração predita do D-PZQ
\hat{C}_L	Concentração predita do L-PZQ
\hat{S}_{CD}	Sinal de dicroísmo circular predito
\hat{S}_{UV}	Sinal de ultravioleta-visível predito
A_D	Luz polarizada circularmente para a direita (dicroísmo circular)

A_L	Luz polarizada circularmente para a esquerda (dicroísmo circular);
C_D	Concentração experimental do D-PZQ
C_L	Concentração experimental do L-PZQ
S_{CD}	Sinal de dicroísmo circular experimental
S_{UV}	Sinal de ultravioleta-visível experimental

1. INTRODUÇÃO

As doenças tropicais negligenciadas são aquelas causadas por agentes infecciosos ou parasitas e são consideradas endêmicas nas regiões mais pobres do mundo, onde o acesso à água potável, saneamento e a cuidados de saúde são precários. Essas enfermidades afetam mais de 1 bilhão de pessoas em todo o mundo, e são causadas por uma variedade de patógenos. Elas são classificadas como “negligenciadas”, pois estão quase ausentes da agenda global de saúde ou estão associadas à exclusão social (FIOCRUZ, 2022). Uma dessas doenças é a esquistossomose, sendo causada pelo parasita *Schistosoma mansoni*. A pessoa é infectada quando entra em contato com água doce onde existam caramujos infectados pelos vermes causadores da esquistossomose. Em casos graves, o paciente desenvolve sintomas como emagrecimento, fraqueza acentuada e aumento do volume do abdômen, conhecido popularmente como barriga d’água. Se a doença não for tratada, o paciente pode sofrer lesão em órgãos como fígado e baço (FIOCRUZ, 2022).

O tratamento da esquistossomose é feito por meio de um medicamento à base de praziquantel. O praziquantel é um derivado pirazino-isoquinoléinico, do grupo dos tioxantônicos, que oferece larga margem de segurança para o tratamento da esquistossomose e efeitos colaterais leves (MINISTÉRIO DA SAÚDE, 2014). Devido a essas características é o medicamento utilizado pelos Programas de Controle da Esquistossomose no mundo (MINISTÉRIO DA SAÚDE, 2018). Além disso, ainda não há relatos de cepas resistentes ao medicamento em áreas endêmicas no Brasil (MINISTÉRIO DA SAÚDE, 2014). A administração do praziquantel é de forma oral, utilizando comprimidos. Dessa maneira, essa forma de tratamento não é indicada para crianças em idade pré-escolar (menos de cinco anos de idade) devido à dificuldade de ingestão do medicamento. Como alternativa, utiliza-se a forma líquida do medicamento para as crianças nessa faixa de idade. O principal desafio na produção dessa forma líquida é o amargor resultante da presença do enantiômero dextrógiro D-PZQ (ingrediente inativo), assim durante a formulação é necessário que exista um enriquecimento do componente levógiro da mistura racêmica, ou seja, o L-PZQ a partir da separação de tais componentes (CUNHA, 2021).

O Leito Móvel Simulado (LMS) consiste na combinação dos conceitos de Cromatografia Líquida de Alto Desempenho (HPLC) e a eficiência de um processo contínuo

em contracorrente, podendo ser utilizado para a separação de enantiômeros. Foi proposto em 1961 por Donald Broughton e Clarence Gerhold. O principal objetivo desse processo é superar as limitações associadas à separação por HPLC, uma vez que neste tipo de cromatografia a solução de alimentação precisa ser adicionada em pulsos, reduzindo o contato entre a fase sólida e líquida, e o processo é realizado em batelada. O LMS parte da concepção de um processo denominado Leito Móvel Verdadeiro (LMV) no qual a fase sólida está em movimento em contracorrente com a fase líquida, buscando garantir um maior contato entre as fases (CUNHA, 2021).

Atualmente, é comum que muitas unidades de LMS não operem de forma otimizada, mas em regiões próximas da ótima devido às dificuldades associadas ao processo de LMS. Como as variáveis controladas estão associadas às concentrações de saída, espera-se que fatores como tempo morto sejam obstáculos para o controle de processo. Além disso, para um controle em malha fechada adequado, é necessário que haja um sistema de medição em linha confiável a fim de garantir que o controlador realize as alterações corretas nas variáveis manipuladas (vazão de solvente e tempo de chaveamento, por exemplo) (CUNHA, 2021).

Aprendizado de máquina representa um subgrupo de técnicas pertencentes à Inteligência Artificial. A partir do aprendizado de máquina, surgem técnicas inspiradas no sistema nervoso humano, como por exemplo as redes neurais que, nesse caso, seriam um subgrupo do aprendizado de máquina. Além disso, ao mimetizar as características humanas como o aprendizado, essas ferramentas apresentam um grande potencial para solução de problemas não lineares. Dessa forma, o aprendizado de máquina vem ganhando cada vez mais espaço atualmente (NOGUEIRA PEREIRA, 2021).

Na Escola de Química/UFRJ existe uma unidade LMS que foi construída no trabalho desenvolvido por CUNHA (2021). A unidade conta com um detector quiral e um computador com um software de sistema supervisório (CUNHA et al., 2019; HOLANDA et al., 2022). As partes elétrica e eletrônica também foram desenvolvidas, tendo as bombas e o detector quiral sinais analógicos, enquanto as válvulas, geram sinais digitais.

Os trabalhos desenvolvidos pelo LADES (Laboratório de Desenvolvimento de Software para Controle e Otimização de Processos) também contribuíram para o projeto da unidade LMS. O software utilizado no sistema supervisório foi desenvolvido no grupo utilizando o LabView. Além disso, softwares como o SimoBed e SimoCon (ANDRADE NETO et al., 2016), também desenvolvidos ao LADES, foram utilizados para realizar as simulações e controle na unidade

LMS. Diferentes modos de operação da unidade LMS também foram avaliados e otimizados nos trabalhos de CALDERON-SUPELANO et al. (2020, 2021, 2022)

Na unidade LMS, CUNHA (2021) também desenvolveu estudos de modelagem e simulação de medição de concentração. A partir das medições do detector quiral, que mede sinais de UV e dicroísmo circular simultaneamente, obtiveram-se modelos matemáticos que permitem correlacionar os sinais de UV e dicroísmo circular com as concentrações de cada enantiômero (CUNHA et al., 2020). Essas correlações foram feitas em baixas concentrações, observando-se uma relação não linear entre os sinais e as concentrações.

As redes neurais possuem vantagens em relação a outras técnicas computacionais devido às suas propriedades de baixo custo computacional e de serem um aproximador universal, e mostram-se como uma boa alternativa para a predição de correlações não lineares. Com a utilização dessas redes, a informação é distribuída em cada um de seus nós, permitindo que o processamento de informação seja feito de maneira mais flexível. As redes neurais também possuem capacidade de aprendizagem, uma característica intrínseca do aprendizado de máquina, que permite direcionar a solução para a região de menor erro com base na informação já obtida do conjunto de dados, reduzindo o custo computacional (BAUGHMAN; LIU, 2014). As redes neurais já foram usadas previamente no tratamento de sinais não lineares de elementos de medição. Por exemplo, no trabalho desenvolvido por FERREIRA et al. (2001), as redes neurais foram utilizadas para interpretar os sinais de biossensores para o controle de um sistema de fermentação alcoólica, concluiu-se que essas redes fornecem correlações adequadas entre os sinais provenientes desses biossensores e as concentrações medidas. Vale ressaltar que a principal dificuldade associada ao uso desses sensores está justamente ligada à resposta não linear para diferentes concentrações (FERREIRA et al., 2001).

O objetivo deste trabalho é obter uma rede neuronal capaz de prever, adequadamente, as concentrações dos enantiômeros L-PZQ e D-PZQ de uma unidade LMS, a partir dos sinais obtidos de um detector quiral.

Para atingir o objetivo, os objetivos específicos serão:

- Realizar a otimização dos parâmetros da rede, obtendo-se a que possui maior coeficiente de determinação;
- Comparar diferentes abordagens de normalização dos dados de entrada;

- Avaliar o treinamento das redes para prever as concentrações em conjunto ou separadamente e como isso afetaria o coeficiente de determinação;
- Comparar com a abordagem de “Random Forest”, para verificar a qualidade do modelo obtido.

Este documento está organizado em 5 capítulos, referências e apêndices:

- **Capítulo 1:** Fornece uma contextualização do trabalho, abordando a problemática em questão e como ela afeta a sociedade. Além disso, também apresenta o objetivo principal e os objetivos específicos do trabalho, evidenciando as dificuldades associadas à medição da concentração dos enantiômeros de praziquantel, bem como a aplicabilidade das redes neurais;
- **Capítulo 2:** Apresenta a fundamentação teórica mais detalhada dos conceitos envolvidos. Descreve o LMS utilizado e como é feita a obtenção dos sinais usados como base para o treinamento da rede neuronal. Destacam-se também os fundamentos de aprendizado de máquina e redes neurais. Por último, é feita uma revisão sobre relacionados ao tema em questão;
- **Capítulo 3:** Apresentar a metodologia utilizada, explicando o funcionamento do treinamento e validação, bem como a comparação de redes com outras abordagens;
- **Capítulo 4:** Apresentar os resultados obtidos e os dados utilizados para treinamento, descrevendo as estratégias usadas para esta etapa. Também buscou-se descrever, de forma mais detalhada, a comparação com outros algoritmos;
- **Capítulo 5:** Conclusões do trabalho desenvolvido e sugestões para trabalhos futuros.

2. FUNDAMENTOS E REVISÃO BIBLIOGRÁFICA

Neste capítulo serão detalhadas as informações referentes ao praziquantel, incluindo a sua estrutura química e a consequência de suas características. Também é apresentado o funcionamento do leito móvel simulado (LMS), bem como os conceitos pertinentes a esse processo cromatográfico. Além disso, abordados os conceitos associados ao dicroísmo circular e a interpretação dos seus sinais, destacando como ele participa da determinação da concentração dos enantiômeros em uma mistura racêmica.

Na sequência, discutidos os fundamentos do aprendizado de máquina, redes neurais e do algoritmo “Random Forest”. Na penúltima seção do capítulo, são revisados os trabalhos da literatura com objetivos relacionados às medições em linha em LMS, incluindo a aplicação das redes neurais para interpretação de sinais espectroscópicos e predição a partir de dados de sensores.

Na última seção do capítulo, é feita uma discussão sobre os pontos em comum entre este trabalho e os apresentados. Além disso, evidencia-se a existência de trabalhos na Escola de Química na linha de LMS, mas que nenhum deles utilizou o aprendizado de máquina para relacionar os sinais de medição e as concentrações dos enantiômeros.

2.1. PRAZIQUANTEL

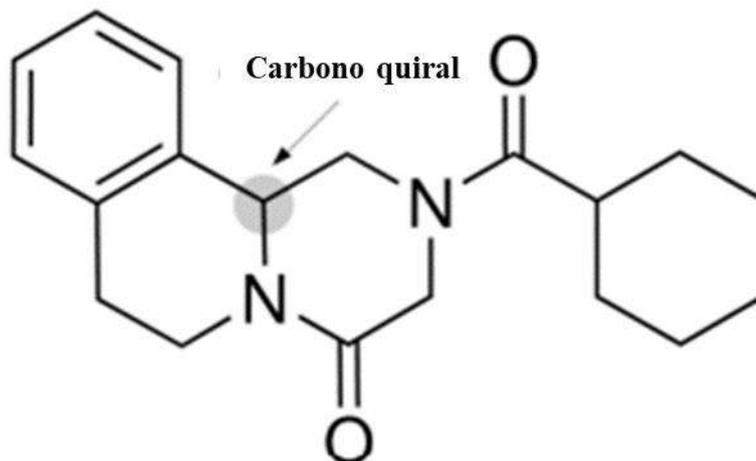
O praziquantel é um medicamento altamente ativo contra um amplo espectro de trematódeos. É a principal droga utilizada no tratamento da esquistossomose e está incluído em programas comunitários de controle dessa doença. É um medicamento de rápida absorção e o seu mecanismo de ação não está totalmente compreendido, mas se acredita que ele cause o rompimento do tegumento do parasita (BENNETT; DOLIN; BLASER, 2020).

Apesar de ser amplamente utilizado, estudos mostraram que a resistência ao medicamento pode ocorrer no *Schistosoma mansoni* mesmo que em um nível baixo (DOENHOFF et al., 2002)

Com relação à sua estrutura química, ele é considerado opticamente ativo, devido à presença de um carbono quiral em sua estrutura, como mostra a Figura 2-1. O PZQ é

administrado como uma mistura racêmica na composição de 1:1 dos seus enantiômeros (MEYER et al., 2009)

Figura 2-1 - Estrutura do praziquantel.



Fonte: Adaptado de CUNHA (2021).

Na mistura produzida, o L-PZQ é a forma ativa, enquanto a forma D-PZQ, inativa. Além disso, o D-PZQ contribui para o gosto amargo na administração do medicamento (MEYER et al., 2009).

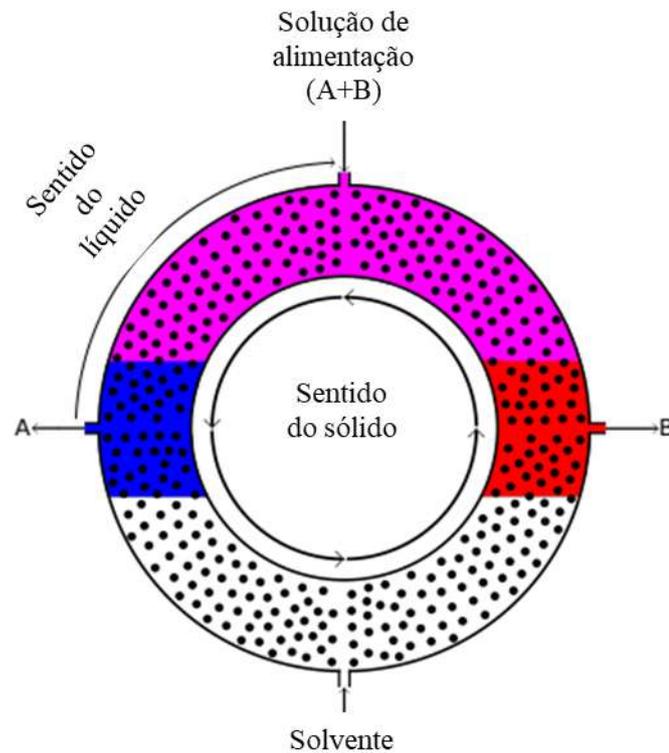
2.2. LEITO MÓVEL SIMULADO

O leito móvel simulado (LMS) é um processo cromatográfico de múltiplas colunas que vem se tornando uma das técnicas mais utilizadas para a separação contínua de enantiômeros. No geral, quando comparado à cromatografia de coluna única, ele possui uma maior eficiência, reduzindo o consumo de solvente. Além disso, o LMS pode ser utilizado em pequenas e grandes escalas (RAJENDRAN; PAREDES; MAZZOTTI, 2009).

O LMS surge de um processo idealizado denominado leito móvel verdadeiro (LMV), no qual existe o movimento da fase sólida em contracorrente com a fase líquida. Esse movimento é idealizado, pois poderia causar queda de pressão e contaminação do extrato pela

fase sólida. O LMV poderia ser pensado como sendo uma coluna circular com a fase sólida em movimento e com as entradas para recebimento do solvente e da mistura (CUNHA, 2021), ilustrado na Figura 2-2.

Figura 2-2 - Ilustração de um leito móvel verdadeiro.



Fonte: Adaptado de CUNHA (2021).

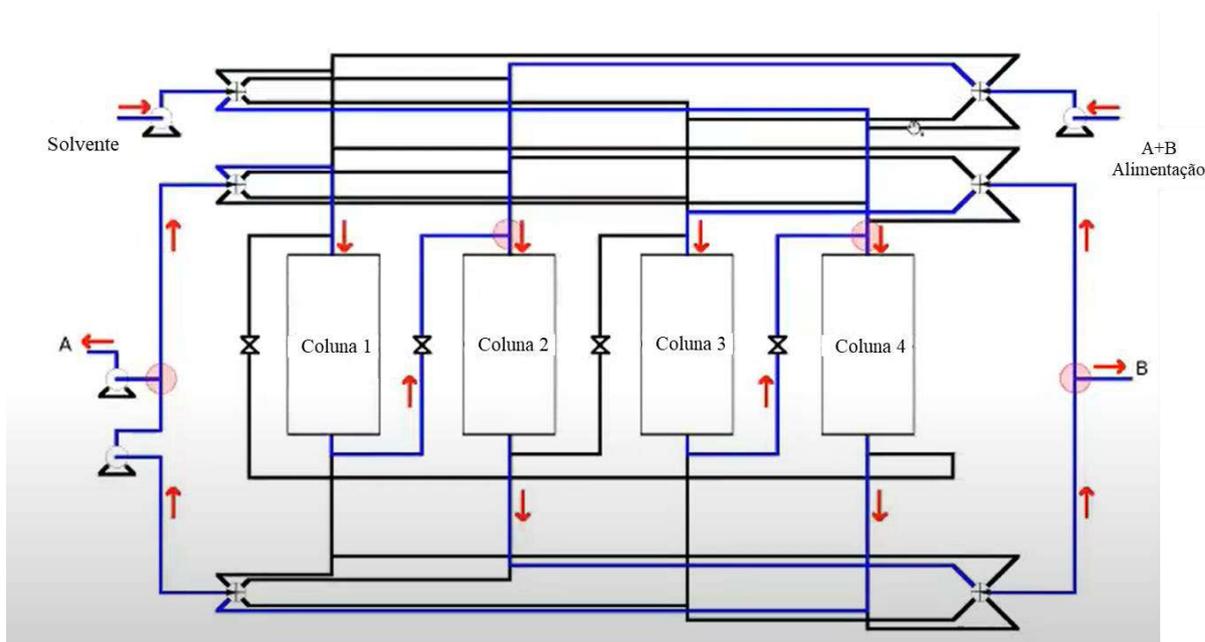
A forma de operação do LMV também fornece uma estimativa das vazões a serem utilizadas no LMS. A transição de um LMV para um LMS consiste em modificações na estrutura idealizada acima (CUNHA, 2021):

- A coluna circular é dividida em quatro zonas que possuem as entradas e saídas situadas entre as mesmas. Cada zona pode conter uma ou mais colunas, como ilustrado na Figura 2-3 para uma coluna por zona;
- O movimento da fase sólida deixa de existir e, para garantir o movimento relativo entre as fases, as entradas e saídas passam a trocar as suas posições periodicamente. Assim, o movimento contracorrente também é mantido. Esta etapa é

responsável pelo termo “simulado”, uma vez que a fase sólida não está se movimentando efetivamente.

Com relação à estrutura de uma unidade LMS, ela é composta por uma série de colunas e um conjunto de bombas para garantir a circulação da mistura e do solvente (ANDRADE NETO et al., 2016). O sistema de válvulas é uma parte importante de uma unidade LMS, realizando as trocas de posições das correntes de entrada e saída (ANDRADE NETO, 2015).

Figura 2-3 - Estrutura de uma unidade LMS.



Fonte: Adaptado de CUNHA (2021).

As 4 zonas de separação descritas a seguir, em que: “A” e “B” representam os componentes da mistura de alimentação (CUNHA, 2021):

- Zona 1: Regenera a fase sólida;
- Zona 2: Enriquece a fase sólida com A e a fase líquida com B;
- Zona 3: Enriquece a fase sólida com B e a fase líquida com A;
- Zona 4: Regenera a fase líquida.

Assim, devido ao fato de o LMS realizar a separação de forma contínua da mistura racêmica (CUNHA, 2021) e devido à sua eficiência, ele é utilizado em muitos segmentos

industriais que necessitam separar compostos complexos como petroquímicos e farmacêuticos (RAJENDRAN; PAREDES; MAZZOTTI, 2009).

2.3. O DICROÍSMO CIRCULAR E A INTERPRETAÇÃO DE SEUS SINAIS

O dicroísmo circular é um método de espectroscopia baseado no fato de que algumas moléculas interagem de forma distinta com a luz circularmente polarizada para a esquerda ou direita (WOODY, 1995).

As moléculas opticamente ativas irão absorver preferencialmente a luz polarizada em uma determinada direção. Essa diferença de absorção pode ser quantificada e está diretamente associada à presença dos enantiômeros em uma mistura racêmica, por exemplo (HURLBURT, 2022).

Como o dicroísmo circular fornece a diferença de absorção entre a luz nas duas direções, Equação (2-1), é necessária a utilização de uma técnica de absorção complementar como a UV-vis que fornece a absorção total da luz polarizada, Equação (2-2) (DRAKE et al, 1980).

$$S_{CD} = A_L - A_D \quad (2-1)$$

$$S_{UV} = \overline{A}_L + \overline{A}_D \quad (2-2)$$

Em que:

- S_{CD} : Sinal de dicroísmo circular obtido do detector quiral;
- S_{UV} : Sinal de ultravioleta visível obtido a partir do detector quiral
- A_L : Luz polarizada circularmente para a esquerda (dicroísmo circular);
- A_D : Luz polarizada circularmente para a direita (dicroísmo circular);
- \overline{A}_L : Luz polarizada circularmente para a esquerda (ultravioleta-visível);
- \overline{A}_D : Luz polarizada circularmente para a direita (ultravioleta-visível).

O cálculo desses dois sinais pode ser feito com um equipamento denominado “Detector Quiral”. Este equipamento emite uma luz circular polarizada para esquerda e direita periodicamente, detectando a diferença de absorção (medida associada ao dicroísmo circular) e a absorção total (medida associada à absorção UV-vis) (CUNHA, 2021).

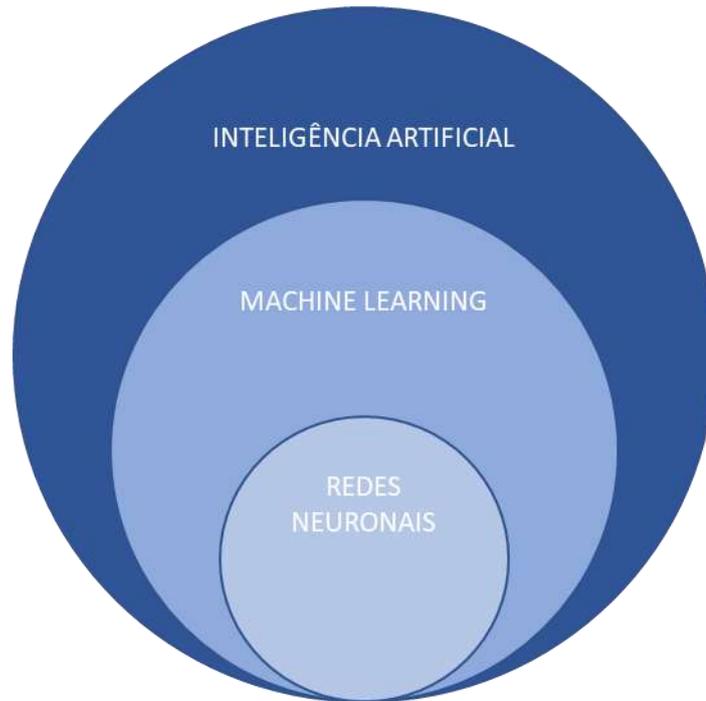
Assim, os sinais obtidos podem ser correlacionados com as concentrações individuais dos enantiômeros. No caso do praziquantel, para baixas concentrações, existe um comportamento não linear entre os sinais de saída e a concentração de cada enantiômero (CUNHA, 2021).

2.4. FUNDAMENTOS DE APRENDIZADO DE MÁQUINA E REDES NEURONAIS

2.4.1. APRENDIZADO DE MÁQUINA

Aprendizado de máquina é um subcampo da inteligência artificial que pode ser definido como a capacidade de uma máquina em imitar o comportamento humano inteligente. Assim, o aprendizado de máquina pode ser considerado como uma forma de aplicação da inteligência artificial, permitindo que os computadores desenvolvam a habilidade de aprender sem que isso seja programado diretamente (SAMUEL, 1950 apud BROWN, 2021). Na Figura 2-4 está representada a divisão dos subcampos da inteligência artificial.

Figura 2-4 - Posição das redes neurais e do aprendizado de máquina dentro da inteligência artificial.



Fonte: Adaptado de NOGUEIRA PEREIRA (2021) e ALOM. et al (2019).

Existem três subcategorias de aprendizado de máquina (BROWN, 2021):

- **Aprendizado supervisionado:** Nesta abordagem, os modelos de aprendizado de máquina são treinados com conjuntos de dados rotulados, permitindo que os modelos aprendam e se tornem mais acurados ao longo do tempo. É a abordagem mais utilizada atualmente;
- **Aprendizado não supervisionado:** O programa procura por padrões em dados não rotulados. O aprendizado de máquina não supervisionado pode encontrar padrões ou tendências que não estão evidentes;
- **Aprendizado por reforço:** O computador é treinado por meio de tentativa e erro para realizar a melhor ação, estabelecendo um sistema de recompensa.

2.4.2. REDES NEURONAIS

2.4.2.1. DESCRIÇÃO

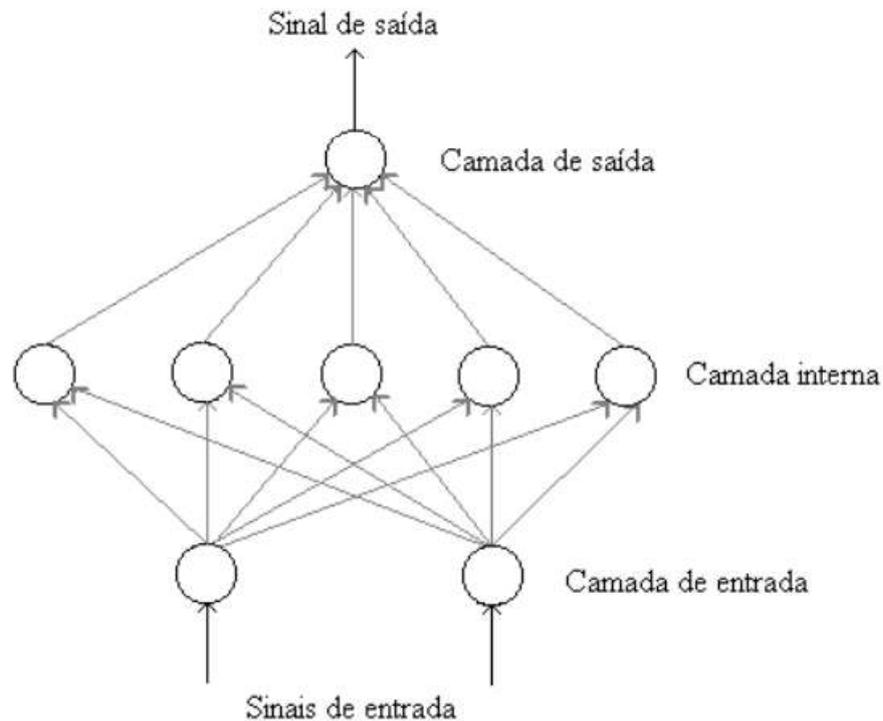
O termo “Rede neuronal” é resultado de pesquisas no campo da inteligência artificial e funciona como um subcampo do aprendizado de máquina, que busca entender e modelar o funcionamento do cérebro e aplicá-lo na computação (BAUGHMAN; LIU, 2014). A inteligência artificial pode ser definida com o ramo da ciência da computação que lida com processamento simbólico para resolução de problemas, ou seja, utiliza técnicas não algorítmicas (BUCHANAN; SHORTLIFFE, 1983 apud BAUGHMAN; LIU, 2014). No processamento simbólico, o conjunto de informações disponíveis sobre um domínio particular é extraído e representado simbolicamente por regras (STOCK, 1989 apud DE SOUZA JR., 1993).

O principal objetivo das redes neurais é mapear um conjunto de padrões de entrada e associá-los a uma saída. Para isso, a rede começa aprendendo a partir de um conjunto de exemplos anteriores com entradas e saídas já definidas. Em seguida, a rede aplica esse aprendizado em um novo dado de entrada para prever a saída apropriada (DE SOUZA, JR., 1993).

2.4.2.2. ESTRUTURA

Uma rede neuronal é composta por processadores denominados “neurônios”. Estes neurônios ficam dispostos em camadas e interconectados por canais unidirecionais, análogos às sinapses de um neurônio biológico (DE SOUZA JR., 1993). Estes processadores semelhantes aos neurônios recebem o nome de “perceptron”. Na Figura 2-5 está descrita uma rede neuronal do tipo “feedforward” com três camadas. O termo “feedforwad” indica que a informação processada por cada neurônio é passada apenas para frente (DE SOUZA JR.,1993):

Figura 2-5 - Exemplo de organização de uma rede neuronal do tipo feedforward.



Fonte: DE SOUZA JR (1993).

A estrutura acima é denominada “Multi-layer perceptron” (MLP) e pode ser considerada como um suplemento da rede neuronal “feedforward” (ABIRAMI; CHITRA,2020). Ela consiste em três tipos de camadas (BAUGHMAN; LIU, 2014):

- Camada interna: recebe a informação da fonte externa de dados. Transmite a informação para o resto da rede para que seja processada;
- Camada interna (“hidden layer”): recebe a informação da camada de entrada e processa toda a informação. O termo “Hidden” se origina do fato desse processamento ser feito de forma escondida;
- Camada de saída: recebe a informação da camada interna da rede e envia o resultado para o receptor externo.

Quando um “perceptron” recebe um conjunto de informações/entradas provenientes de outra camada ela passa pelos seguintes procedimentos (BAUGHMAN; LIU, 2014):

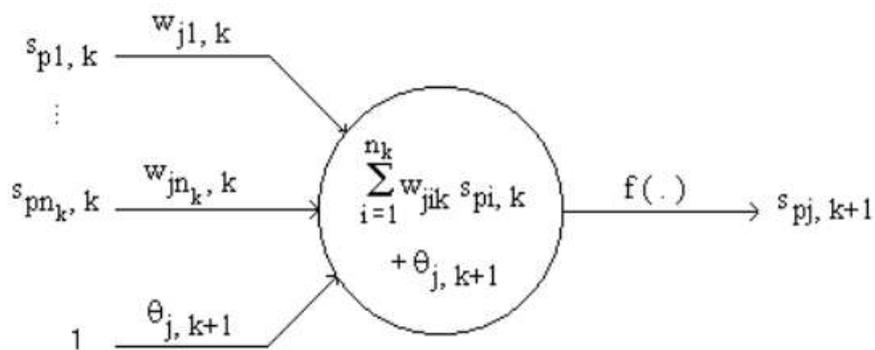
- Cada entrada será multiplicada por um “peso” correspondente a ela;

- Na sequência, todas as entradas que foram multiplicadas serão somadas e um termo denominado “bias”, que funciona como um limite interno de ativação do “perceptron”, também é computado nessa parcela, mas sem passar pela multiplicação dos pesos;
- Por último, o resultado dessa soma será inserido em uma função de ativação. A saída da função de ativação segue para os outros neurônios e o processo se repete.

A utilização de redes com múltiplas camadas internas é denominada “deep learning”. Essa abordagem permite que a rede aprenda sobre tendências ou padrões mais complexos, uma vez que a informação é processada em várias camadas com o aprendizado sendo transferido para a camada subsequente (Goodfellow et al, 2016).

A Figura 2-6 mostra a generalização do processo descrito acima para um neurônio “j” qualquer da camada (k+1). Ele recebe um conjunto de informações $s_{pi,k}$ ($i=1,\dots,n_k$), correspondentes às saídas dos n_k neurônios da camada anterior, ponderadas, cada uma, pelo peso $w_{j,i,k}$ correspondente a sua conexão. As entradas são somadas ponderadas pelo peso e o valor resultante é somado ao bias. O resultado passa por uma função de ativação e segue para o próximo neurônio (DE SOUZA JR,1993):

Figura 2-6 - Ativação do neurônio artificial.



Fonte: DE SOUZA JR (1993).

O resultado antes de ser inserido na função de ativação tem a forma:

$$(2-3)$$

$$\lambda_{pj,k+1} = \left[\sum_{i=1}^{n_k} w_{jik} * s_{pi,k} \right] + \theta_{j,k+1}$$

Com base nas bibliotecas de redes neuronais (Scikit-Learn), as principais funções de ativação são:

- Tangente hiperbólica:

$$f(\lambda_{pj,k+1}) = \tanh(\lambda_{pj,k+1}) \quad (2-4)$$

- Identidade:

$$f(\lambda_{pj,k+1}) = \lambda_{pj,k+1} \quad (2-5)$$

- Logística:

$$f(\lambda_{pj,k+1}) = \frac{1}{1 + e^{-\lambda_{pj,k+1}}} \quad (2-6)$$

- ReLu:

$$f(\lambda_{pj,k+1}) = \max(0, \lambda_{pj,k+1}) \quad (2-7)$$

Esse procedimento de ativação se repete desde a camada de entrada até a camada de saída (DE SOUZA JR, 1993).

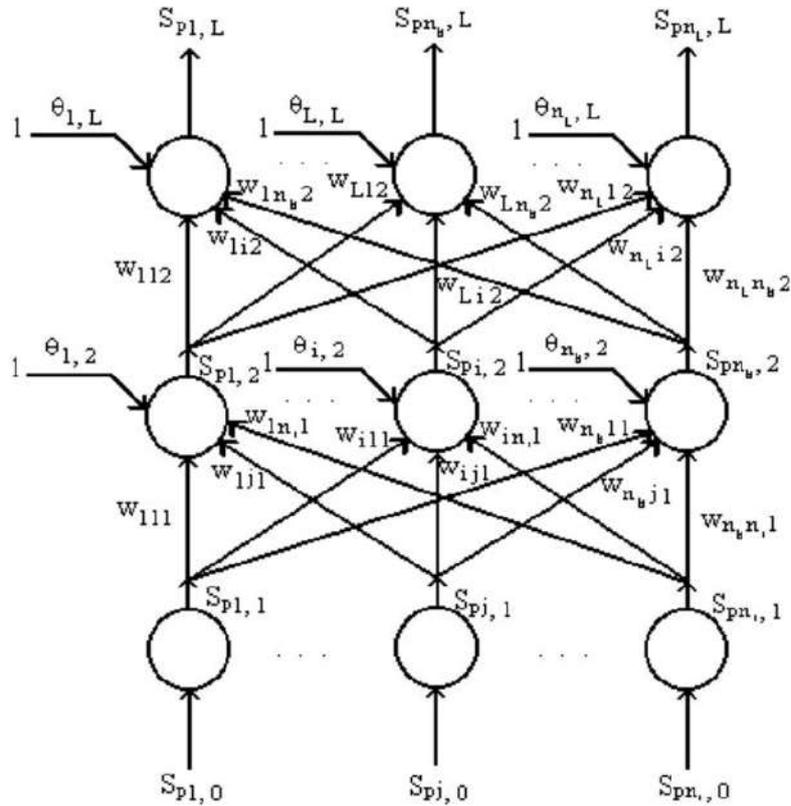
2.4.2.3. BACKPROPAGATION

Consiste em um procedimento de aprendizagem utilizado em redes neuronais. É um ajuste repetitivo dos pesos e dos “biases” das conexões de modo a minimizar o erro entre o vetor de saída real e o vetor de saída da rede (RUMELHART; HINTON; WILLIAMS, 1986).

As redes treinadas pelo método “backpropagation” são multicamadas, não apresentam conexões laterais e os seus sinais são do tipo “feedforward” (DE SOUZA JR, 1993), ilustrado na Figura 2-7. No geral, o “backpropagation” é o algoritmo de treinamento mais utilizado nas redes neuronais do tipo MLP, utilizando o gradiente descendente do erro (GUPTA, 2018).

$$\nabla E(W) = 0 \quad (2-8)$$

Figura 2-7 - Exemplo de rede feedforward com três camadas.



Fonte: De Souza Jr. (1993).

A Equação (2-10) representa o erro quadrático entre as saídas e o valor de referência, enquanto a Equação (2-9) seria o somatório dos erros de cada saída, ou seja, para cada N_p padrões presentes na rede. O termo “s” é referente às saídas e é obtido após as entradas passarem por todas as camadas da rede (DE SOUZA JR,1993):

$$E = \sum_{p=1}^{N_p} E_p \quad (2-9)$$

$$E_p = \frac{1}{2} * \sum_{l=1}^{n_L} (s_{pl,L} - t_{pl})^2 \quad (2-10)$$

A atualização dos pesos e dos “biases” é feita com base em um fator η denominado “taxa de aprendizagem”. Assim, os valores dos pesos e dos “biases” serão modificados gradualmente para minimizarem o erro (SAXE; MCCLELLAND; GANGULI, 2013). Os valores utilizados para a taxa de aprendizagem são geralmente bem menores que 1 (DE SOUZAR JR., 1993):

$$\mathbf{W}_{k+1} = \mathbf{W}_k - \eta \nabla E(\mathbf{W}_k) \quad (2-11)$$

$$\mathbf{B}_{k+1} = \mathbf{B}_k - \eta \nabla E(\mathbf{B}_k) \quad (2-12)$$

Dessa forma, a etapa de treinamento/”backpropagation” pode ser dividida em duas etapas principais (RUMELHART; MCCLELLAND 1986):

- “Forward pass”: é a etapa na qual os dados são processados pela rede, ou seja, desde a entrada até a camada de saída;
- “Backward pass”: é a etapa na qual a informação correspondente ao erro é passada para a rede novamente, ou seja, atualização dos pesos e biases.

Um dos principais problemas associados ao “backpropagation” é a convergência. Valores muito pequenos para η tornam a convergência lenta, enquanto valores muito grandes tornam a solução oscilatória (WALTROUS, 1988 apud DE SOUZA JR,1993). Dessa forma foram desenvolvidas variações na determinação dos valores de η bem como a utilização de um novo parâmetro ε denominado momento (RUMELHART; MCCLELLAND 1986). A Equação (2-13) representa implementação dessa modificação (DE SOUZA JR., 1993):

$$\mathbf{W}_{k+1} = \mathbf{W}_k - \eta \nabla E(\mathbf{W}_k) + \varepsilon \Delta \mathbf{W}_k \quad (2-13)$$

Apesar da adição do termo de momento, essa modificação pode não ser suficiente. Assim, uma alternativa é a utilização de uma taxa de aprendizagem e momento dinâmicos, sendo atualizados também com base no erro (LEONARD; KRAMER, 1989).

O algoritmo “Adam” também implementa modificações no método do gradiente descendente, buscando melhorar a eficiência computacional e reduzir os requisitos de memória. Esse método consiste em uma otimização estocástica baseada no gradiente de primeira ordem com a utilização de taxas de aprendizagem adaptativas. Uma das vantagens deste método é a sua adequação a problemas com uma grande quantidade de dados e de parâmetros (KINGMA; BA, 2015).

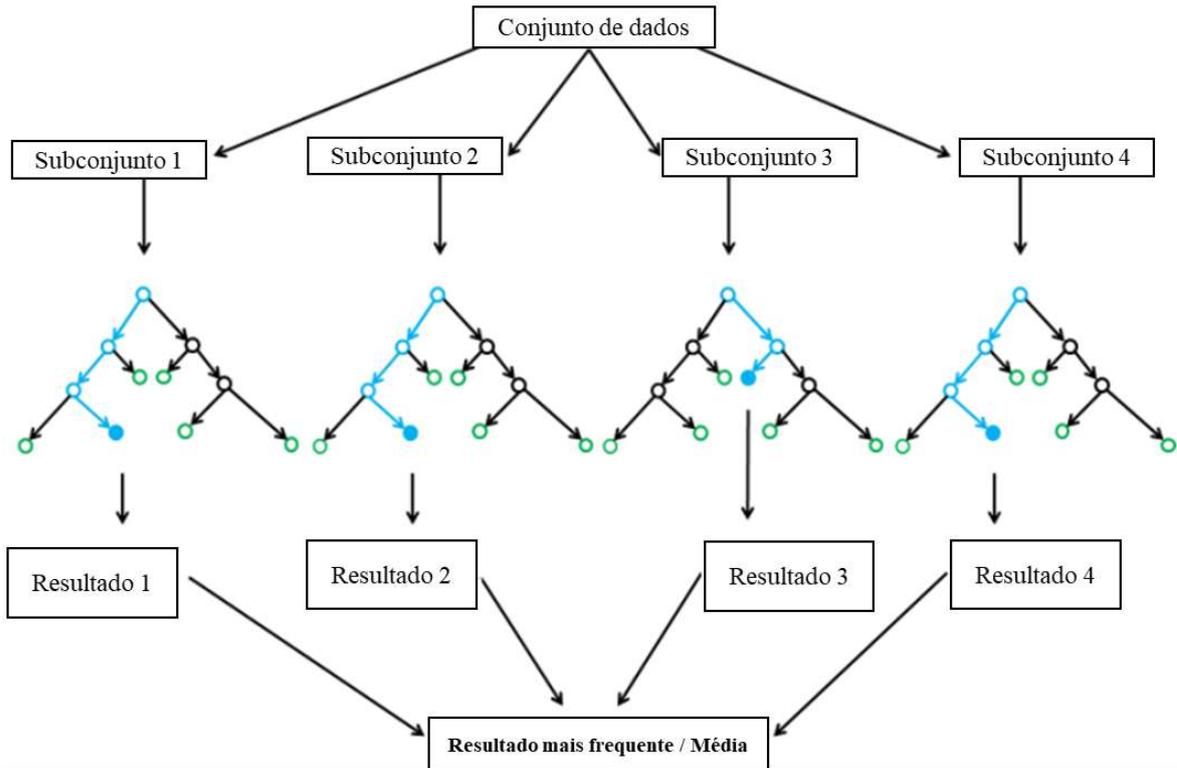
O processo de treinamento da rede neuronal termina ao atingir o número máximo de iterações ou com base na convergência dos valores. Após esta etapa de treinamento, os pesos e “biases” obtidos são armazenados e podem ser utilizados para a predição de saídas a partir de novas entradas. Na prática, os pesos e “biases” do treinamento são validados com outros pares de entradas e saídas conhecidos, antes de serem implementados diretamente para predição (DE SOUZA JR., 1993).

2.4.3. RANDOM FOREST

Este algoritmo consiste na utilização de árvores de decisão para classificação e predição (SPEISER et al., 2019). No “RandomForest” são geradas amostras a partir dos dados iniciais, com reposição, para a construção das árvores, esta etapa é denominada “bootstrap” (BELGIU e DRĂGUT, 2016). Na sequência, cada amostra passa pelas árvores, resultando em uma saída. No final, no caso de um problema de classificação, a resposta que aparecer em maior quantidade será considerada a predição da rede. No caso de um problema de regressão, as saídas são utilizadas para o cálculo de um valor médio que será o resultado (NOGUEIRA PEREIRA, 2021).

O procedimento de combinar as respostas de cada árvore é chamado de “Aggregating”. O procedimento de “Bootstrap” combinado com o “Aggregating” é denominado “Bagging” (NOGUEIRA PEREIRA, 2021).

Figura 2-8 - Algoritmo Random Forest.



Fonte: Adaptado de NOGUEIRA, PEREIRA (2021)

2.5. REVISÃO DA LITERATURA

Um trabalho mais diretamente relacionado à presente contribuição é o de CUNHA et al, 2020, no qual, foi realizado um estudo a partir dos sinais obtidos de um detector quiral em uma unidade de Leito Móvel Simulado. O detector mediu os sinais de dicroísmo circular e UV-vis de uma corrente com diferentes composições dos enantiômeros de praziquantel. A relação entre as concentrações dos enantiômeros e dos sinais do detector foi não linear. Assim, foi construído um modelo matemático empírico para relacionar os sinais de UV-vis e dicroísmo circular com as concentrações. A presença dessa não linearidade é um indicativo do potencial uso das redes neurais.

A Equação (2-14) e a Equação (2-15) representam os modelos obtidos a partir dos dados experimentais (CUNHA et al, 2020).

$$S_{UV,j} = \sum_{k=0}^2 \sum_{q=0}^1 (\alpha_{kq} \hat{C}_{L,j}^k * \hat{C}_{D,j}^q) \quad R^2 = 0,98 \quad (2-14)$$

$$S_{CD,j} = \sum_{k=0}^2 \sum_{q=0}^1 (\beta_{kq} \hat{C}_{L,j}^k * \hat{C}_{D,j}^q) \quad R^2 = 0,99 \quad (2-15)$$

Em que:

- \hat{C}_L : Concentração do L-PZQ;
- \hat{C}_D : Concentração do D-PZQ;
- α e β : Parâmetros determinados a partir do ajuste dos dados experimentais.

Para a determinação do modelo foram realizados ensaios em diferentes concentrações de praziquantel, incluindo a mistura racêmica e os enantiômeros puros. Utilizou-se etanol como solvente com uma vazão de 1mL/min. Na sequência, a solução foi alimentada no detector quiral a partir de uma bomba “HPLC”. Assim, obtiveram-se os sinais correspondentes a cada concentração (CUNHA et al, 2020).

Em seguida os dados foram separados em um conjunto de calibração e validação, utilizando o método “Leave-One-Out Cross Validation” que consiste em separar a base de dados repetidamente em uma seção de validação e outra de calibração. A seção de validação possui apenas um ponto enquanto o restante de dados fica seção de calibração (CUNHA et al, 2020).

O objetivo da etapa de calibração é determinar o conjunto de parâmetros que serão utilizados no modelo empírico. Assim, realizou-se a minimização da função objetivo G que depende das concentrações dos enantiômeros. A função e suas variáveis estão representadas nas Equações (2-16) , (2-17) e (2-18) (CUNHA et al, 2020):

$$G(\alpha_{kq}, \beta_{kq}) = (S_{UV} - \hat{S}_{UV})^2 + (S_{CD} - \hat{S}_{CD})^2 \quad (2-16)$$

$$\hat{S}_{UV} = \sum_{k=0}^m \sum_{q=0}^p (\alpha_{kq} C_L^k C_D^q) \quad (2-17)$$

$$\hat{S}_{CD} = \sum_{k=0}^m \sum_{q=0}^p (\beta_{kq} C_L^k C_D^q) \quad (2-18)$$

Em que:

- \hat{S}_{UV} : Sinal UV-vis predito;
- \hat{S}_{CD} : Sinal CD predito.

Com o valor dos parâmetros obtidos, repete-se o procedimento anterior, mas desta vez com o objetivo de determinar as concentrações. Durante a determinação do modelo, concluiu-se que os parâmetros α_{11} e α_{20} não possuem impacto no desempenho do modelo matemático. Assim, as equações para determinação da concentração dos enantiômeros passam a ser a Equação (2-19) e a Equação (2-20) (CUNHA et al., 2020):

$$\hat{S}_{UV,j} = \alpha_{00} + \alpha_{01} \hat{C}_D + \alpha_{10} \hat{C}_L + \alpha_{21} \hat{C}_L^2 \hat{C}_D \quad R^2 = 0,98 \quad (2-19)$$

$$\hat{S}_{CD,j} = \sum_{k=0}^2 \sum_{q=0}^1 (\beta_{kq} \hat{C}_{L,j}^k * \hat{C}_{D,j}^q) \quad R^2 = 0,99 \quad (2-20)$$

A limitação da quantidade de dados empregados (catorze padrões) e o fato do modelo adotado ser polinomial, sujeito a erros de extrapolação, motivam a verificação da qualidade preditiva desse modelo na presença de novos dados e o desenvolvimento de novas abordagens.

Há outros trabalhos na literatura que usam uma metodologia semelhante a utilizada neste trabalho, baseados em dados experimentais, incluindo dados de espectroscopia, e predição, mas para outras aplicações. A seguir serão abordados aqueles que mais se aproximaram da abordagem adotada neste trabalho.

No trabalho desenvolvido por RODRIGUES et al. (2022) utilizou-se uma rede neuronal do tipo “Multi-Layer Perceptron” para a predição das condições de operação de uma prensa de papel industrial. O principal objetivo é conseguir prever os problemas operacionais do equipamento com antecedência, evitando gastos associados com paradas não planejadas, por exemplo. Para o treinamento, foram utilizados dados de três anos dos sensores industriais, coletando informações como corrente elétrica, pressão, temperatura, nível de óleo, torque e velocidade de rotação (RODRIGUES et al., 2022).

Com relação à topologia da rede, foram necessárias apenas duas camadas escondidas com 150 e 75 neurônios respectivamente, utilizando o solver “Adam” e a função logística sigmoide como ativação. Como métrica de avaliação do desempenho da rede, optou-se pelo erro médio absoluto percentual. Em relação aos resultados obtidos por RODRIGUES et al. (2022), observaram-se erros inferiores a 10% para predições de três meses, um mês ou 15 dias, destacando a capacidade das redes neuronais de trabalharem com grandes quantidades de dados (RODRIGUES et al., 2022).

No trabalho desenvolvido por HONGXIANG; YUNTAO; XIANGJUN, 2011 as redes neuronais foram utilizadas para identificar, a partir de dados de espectroscopia de emissão atômica, as condições de um motor a diesel e diferentes tipos de óleos lubrificantes de motor. As informações contidas no óleo de equipamentos mecânicos, como por exemplo a concentração de elementos e partículas, podem revelar a condição de desgaste do equipamento (HONGXIANG; YUNTAO; XIANGJUN, 2011).

Uma das etapas consistiu na análise de trinta e três amostras provenientes de três tipos de óleo utilizando uma rede neuronal baseada em “backpropagation”. Ao realizar o treinamento com todos os dados disponíveis de espectroscopia, a rede foi capaz de prever corretamente todos os tipos de óleo presentes nas amostras (HONGXIANG; YUNTAO; XIANGJUN, 2011).

DAVIES; PATTISON; HIRST, 2022 desenvolveram modelos de aprendizado de máquina para predição da formação de produtos das reações acopladas de “Buchwald-Hartwig” entre hidrazina de benzofenona e 4-clorotolueno (DAVIES; PATTISON; HIRST, 2022).

Os dados foram disponibilizados pela “DeepMatter” e coletados na sua plataforma em nuvem “DigitalGlassware”. Estes dados incluem informações como temperatura, pressão, informações de cromatografia líquida e espectroscopia de massa, cor e umidade. Foram utilizados diversos modelos como regressão linear, regressão cúbica polinomial, “Random

Forest” e redes neuronais do tipo “Long short-term memory”. Dentre os modelos utilizados, as redes neuronais apresentaram o menor erro médio absoluto (1,2%), realizando predições de 120 minutos com erro de 6,3% (DAVIES; PATTISON; HIRST, 2022).

2.6. CONCLUSÕES DA REVISÃO DA LITERATURA

Com base nos trabalhos citados, observou-se que as redes neuronais são as ferramentas mais utilizadas para predição, apesar de existirem outras. As redes são capazes de lidarem com uma grande quantidade de dados, sejam eles variáveis de processo ou de espectroscopia. Essa capacidade de predição também pode ser utilizada com o objetivo de realizar um monitoramento em linha. Com relação ao LMS, na Escola de Química existem trabalhos que já foram desenvolvidos nesta linha. O trabalho de CUNHA et al, 2020 relaciona as concentrações dos enantiômeros de PZQ com os sinais obtidos de um detector quiral acoplado à unidade LMS, a partir de um modelo polinomial. No entanto, estes trabalhos não empregaram o modelo de aprendizado de máquina, motivando o desenvolvimento deste trabalho. Além disso, a disponibilização de mais dados potencializará o uso das redes neuronais, bem como outras ferramentas de aprendizado de máquina.

3. MATERIAIS E MÉTODOS

Neste capítulo serão abordados os dados utilizados para o treinamento, explicando a procedência e o que eles contêm. Na sequência, a seção referente às redes neurais, detalha o procedimento de treinamento e validação da rede e onde ele foi feito, bem como o tratamento inicial dos dados experimentais. Também é abordada a metodologia utilizada para otimização dos hiperparâmetros da rede, ou seja, uma descrição de como foi feita a otimização do seu projeto. Por último, é detalhada a comparação de redes com outras abordagens, evidenciando alternativas para normalização dos dados experimentais e diferentes topologias de rede.

3.1. DADOS

Os dados experimentais foram obtidos a partir de CUNHA (2021) da unidade LMS, totalizando cinquenta e sete pontos com informações referentes aos sinais de UV, dicroísmo circular, concentração de D-PZQ e L-PZQ. Como essa quantidade de dados é relativamente pequena, utilizou-se o modelo obtido em CUNHA et al (2020) para gerar mais pontos para o treinamento da rede. Apesar do treinamento ter sido feito inicialmente com base em um modelo, ele foi utilizado apenas para uma estimativa inicial da melhor rede, como é mostrado ao longo deste capítulo.

Na Figura 3-1 está representada a unidade LMS utilizada para a obtenção dos dados experimentais.

Figura 3-1 - Unidade LMS.



Fonte: Adaptado de CUNHA (2021).

A unidade está dividida em oito seções:

- Seção 1: Detector quiral. Equipamento utilizado para a obtenção dos sinais de dicroísmo circular e UV-vis. Ele está deslocado da sua posição para facilitar a visualização da unidade;
- Seção 2: Colunas de cromatografia;
- Seção 3: Alimentação da mistura de enantiômeros;
- Seções 4 e 5: Bombas utilizadas para manter a circulação dentro do sistema;
- Seção 6: Válvulas utilizadas para a troca periódica da posição das correntes;
- Seção 7: Computador utilizado no sistema supervisorio;
- Seção 8: Saída para obtenção do extrato e refinado.

As colunas cromatográficas estão distribuídas nas quatro zonas presentes em uma unidade LMS, totalizando seis colunas com a seguinte configuração [1 2 2 1], ou seja, uma coluna na zona 1, duas colunas na zona 2, duas colunas na zona 3 e uma coluna na zona 4. Uma bomba “HPLC” está alocada na zona 2 (CUNHA, 2021).

Buscou-se minimizar o tamanho dos tubos que conectam as colunas, válvulas e bombas, uma vez que não há separação da mistura neste trecho. O volume útil correspondente às zonas da unidade LMS está representado na Tabela 3-1 (CUNHA, 2021).

Tabela 3-1 - Volume útil das zonas da unidade LMS.

Zona	Volume útil (cm³)
Zona 1	5,82
Zona 2	11,66
Zona 3	11,66
Zona 4	5,83

Fonte: Adaptado de CUNHA (2021).

3.2. REDES NEURONAIS

Neste trabalho, foram utilizados os recursos fornecidos pela biblioteca Scikit-Learn (Python) e do framework de otimização Optuna. A biblioteca Scikit-Learn possui muitos algoritmos de aprendizado supervisionado e não supervisionado. Esses recursos incluem algoritmos para classificação, regressão e pré-processamento dos dados de entrada. Com relação ao Optuna, ele possui algoritmos eficientes de otimização de hiperparâmetros, além de funções que garantem a visualização do avanço da otimização (OPTUNA, 2022).

No geral, os algoritmos de aprendizagem se beneficiam da normalização dos dados de entrada. Assim, durante a etapa de pré-processamento, foram avaliadas duas abordagens de normalização: “StandardScaler” e “MinMaxScaler”. É importante ressaltar que antes da aplicação dos normalizadores, os dados foram embaralhados e separados para treino e validação, utilizando a função “train_test_split” que pertence à biblioteca sklearn (SCIKIT-LEARN, 2022).

O “StandardScaler” realiza a normalização com base no desvio padrão populacional e na média, enquanto o “MinMaxScaler” se baseia no valor máximo e mínimo dos dados de entrada. As transformações para cada normalizador estão representadas na Equação (3-1) e na Equação (3-2) (SCIKIT-LEARN, 2022).

$$z = \frac{x - \mu}{\sigma} \tag{3-1}$$

$$z = \frac{x - x_{Mínimo}}{x_{Máximo} - x_{Mínimo}} \tag{3-2}$$

Em que:

- z : Representa o valor normalizado;
- x : Representa o dado sem normalização;
- μ : Representa a média dos dados sem normalização;
- σ : Representa o desvio padrão populacional dos dados sem normalização;
- $x_{\text{Mínimo}}$: Representa o valor mínimo dos dados sem normalização;
- $x_{\text{Máximo}}$: Representa o valor máximo dos dados sem normalização.

Com a metodologia estabelecida para o pré-processamento, a transformação foi aplicada aos sinais de UV, Dicroísmo Circular, Concentração do Dextrógiro e Concentração do Levógiro. É importante ressaltar que as concentrações foram normalizadas, pois, apesar de serem saídas do sistema, elas também participam da etapa de treinamento da rede.

Após o pré-processamento, utilizou-se a ferramenta MLPRegressor. Essa classe implementa um perceptron multicamada (MLP) que treina usando “backpropagation” com a função de ativação identidade na camada de saída. A sua saída é um conjunto de valores contínuos e como Função perda, utiliza o erro quadrático. Vale ressaltar que o MLPRegressor aceita saídas múltiplas, o que é essencial para o trabalho, pois o objetivo é a obtenção da concentração dos enantiômeros (SCIKIT-LEARN, 2022).

O MLPRegressor aceita uma quantidade relativamente extensa de parâmetros de configuração (cerca de 23 parâmetros), o que torna uma tarefa difícil a determinação da melhor rede alterando-os manualmente. Dessa forma, foram analisados apenas os parâmetros que possuem maior impacto no desempenho da rede, estando listados na Tabela 3-2.

Tabela 3-2 - Descrição dos parâmetros de maior impacto na rede.

Parâmetro	Descrição
Hidden_layer_sizes	Tamanho da camada escondida
Activation	Função de ativação da camada escondida
Solver	Solver utilizado
Learning_rate	Taxa de aprendizagem
Max_iter	Número máximo de iterações

Fonte: Adaptado de SCIKIT-LEARN (2022).

Apesar da redução de 23 parâmetros para apenas cinco, a tarefa de alterá-los ainda seria trabalhosa, uma vez que alguns desses parâmetros trazem uma lista de possibilidades. No caso do MLPRegressor, alguns deles foram exemplificados na Tabela 3-3.

Tabela 3-3 - Possibilidades para os parâmetros individuais.

Parâmetro	Possibilidades
Activation	Identity, logistic, tanh, relu
Solver	Lbfgs, sgd, adam
Learning_rate	Constant, invscaling, adaptative

Fonte: Adaptado de SCIKIT-LEARN (2022)

Com base na descrição disponibilizada na biblioteca SKLEARN (2022):

- “Solver lbfgs”: é um otimizador dentro da família dos métodos quasi-Newton;
- “Solver sgd”: refere-se ao gradiente descendente estocástico;
- “Solver adam”: refere-se ao otimizador gradiente descendente estocástico proposto por KINGMA; BA, 2015. Este solver trabalha melhor com uma base de dados de treinamento e validação grande (na faixa dos milhares);
- “Learning rate constant”: é um valor constante para a taxa de aprendizagem, sendo definido por uma função de inicialização;
- “Learning rate invscaling” (usado quando o solver for sgd): reduz gradualmente a taxa de aprendizagem com o passar do tempo. Essa redução é baseada na potência inversa de t ;
- “Learning rate adaptative” (usado quando o solver for sgd): mantém a taxa de aprendizagem constante enquanto a função perda continuar diminuindo. Se em duas tentativas consecutivas não houver redução da função perda, a taxa é dividida por 5.

Diante dessa dificuldade, utilizou-se a ferramenta Optuna para obter o conjunto ótimo de hiperparâmetros da rede com base em uma métrica estabelecida. Vale ressaltar que a otimização foi feita com base na validação cruzada que será explicada posteriormente. No caso da rede neuronal utilizada neste trabalho, foi construída uma lista com os parâmetros presentes

na Tabela 3-2. Os parâmetros são transmitidos ao Optuna que realiza diversas combinações, eliminando as redes com pior desempenho previamente, garantindo um bom uso dos recursos computacionais (OPTUNA, 2022).

A métrica estabelecida foi o coeficiente de determinação médio (R^2), pois está associado à etapa de validação cruzada. Essa etapa divide os dados de treinamento em seções de tamanho especificados e avalia o conjunto de parâmetros em cada uma delas. No final desse procedimento, é avaliado o coeficiente de determinação médio de todas as seções, que é obtido a partir do coeficiente de determinação proveniente de cada seção, utilizando esse valor como um critério de seleção para rede. Esse procedimento permite que a rede neuronal tenha contato com todos os dados que estão sendo trabalhados. Na Figura 3-2, foram realizadas dez divisões e em cada uma delas é obtida uma rede neuronal. Para a avaliar o desempenho das redes, é feita a média da métrica em questão. Assim, são treinadas dez redes, ou seja, uma para cada divisão e, a partir do coeficiente de determinação de cada divisão, é feita a média que funcionaria como critério de desempenho (SCIKIT-LEARN, 2022).

Figura 3-2 - Exemplo de validação cruzada.

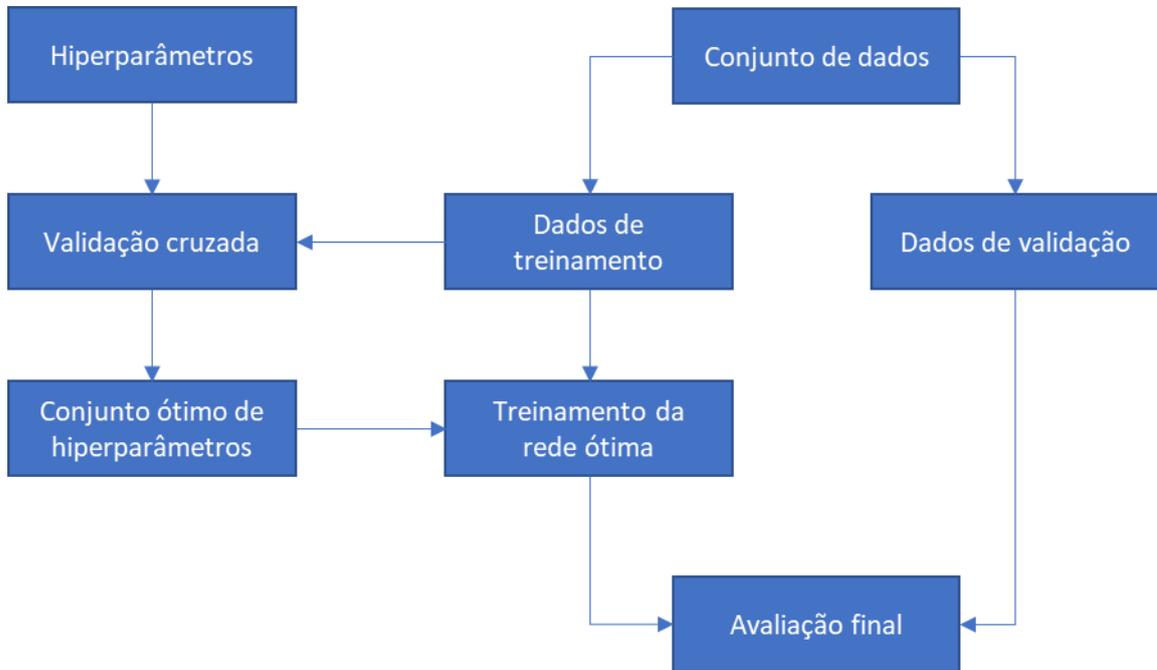


Fonte: Adaptado de Scikit-Learn Cross-Validation (2022).

A saída de interesse do Optuna é o conjunto de hiperparâmetros com melhor desempenho. Na sequência, esses hiperparâmetros são transferidos para outra rede a ser treinada com todos os dados de treinamento. O seu desempenho é avaliado com base na função “score”. A função “score” retorna o coeficiente de determinação, ou seja, o R^2 . Todos os dados de treinamento e validação são salvos no final da rotina como arquivos .csv. É importante ressaltar que ao utilizar a rede para prever a concentração dos enantiômeros, é necessário realizar a transformação inversa da normalização utilizada. Dentro da biblioteca sklearn já existe uma função que realiza esse procedimento, denominada “inverse_transform” (SCIKIT-LEARN, 2022).

Um resumo da metodologia aplicada pode ser descrito com base na Figura 3-3:

Figura 3-3 - Descrição da otimização utilizando validação cruzada e obtenção da rede ótima.



Fonte: Adaptado de Scikit-Learn “cross-validation” (2022).

3.3. COMPARAÇÃO DE REDES COM OUTRAS ABORDAGENS

Como abordado na Seção 3.2, a quantidade relativamente pequena de dados experimentais levou à utilização dos dados simulados com o modelo desenvolvido no trabalho CUNHA et al, 2020 como uma estimativa inicial para rede. Nesse caso, optou-se por realizar o treinamento com os dados fornecidos pelo modelo apenas para obter um conjunto ótimo de hiperparâmetros. Na sequência, eles foram utilizados em uma segunda rede neuronal que seria treinada apenas com os dados experimentais. Um ponto importante a ser ressaltado é que os pesos e os “biases” da rede otimizada inicialmente serviram como uma estimativa inicial para a rede secundária, evitando uma estimativa inicial aleatória. Essa etapa é caracterizada como “Transfer Learn”, ou seja, um tipo de transferência de aprendizado entre as redes neuronais. Sendo assim, foi possível comparar o desempenho da rede com “Transfer Learn” e a rede sem “Transfer Learn” (LI et al., 2020).

Também foi feita uma comparação entre o desempenho das redes neuronais citadas anteriormente com o algoritmo “RandomForest”. O procedimento de otimização utilizando o “RandomForestRegressor” é idêntico ao do “MLPRegressor”. Utilizou-se o Optuna para realizar a otimização dos hiperparâmetros disponíveis para o “RandomForest”. Assim, o conjunto ótimo foi obtido com os dados do modelo e posteriormente transferidos para o estimador para ser treinado com os dados experimentais.

Na Tabela 3-4 estão representados os parâmetros do “RandomForestRegressor” que foram transferidos ao Optuna.

Tabela 3-4 - Parâmetros do Random Forest.

Parâmetro	Descrição
Criterion	Função utilizada para medir a desempenho de cada amostra
Max_features	Número de recursos considerados ao procurar a melhor divisão do “dataset”
Oob_score	Permite o uso de amostras que estão fora da divisão inicial do “dataset” de cada árvore para avaliar a desempenho

Fonte: Adaptado de SCIKIT-LEARN (2022).

Outra forma de comparação foi realizar o treinamento da rede a partir dos mesmos dados de entrada, mas com apenas uma saída. Dessa maneira, foram construídas duas redes neuronais, ou seja, uma para cada enantiômero. A expectativa é que haja uma melhora no desempenho das redes, uma vez que o foco do treinamento será dado para as saídas individualmente. Apesar da relação entre as concentrações de enantiômeros, isto é, não podemos assumir que elas são variáveis independentes entre si, a separação dessas variáveis não comprometerá a análise, pois essa dependência fica implícita nas variáveis de entrada (Sinais UV e CD). O resumo da metodologia é apresentado na Tabela 3-5.

Uma última abordagem foi a obtenção de uma rede otimizada a partir dos dados experimentais diretamente. Como a quantidade de dados disponíveis é relativamente pequena, optou-se por reduzir as divisões de validação cruzada para três.

Após a obtenção da melhor rede diante das diferentes abordagens, realizou-se uma comparação da rede de melhor desempenho com o modelo matemático empírico de CUNHA et al. (2020).

Tabela 3-5 - Resumo da metodologia.

RESUMO
1) Pré-processamento e normalização dos dados
2) Otimização dos hiperparâmetros com validação cruzada via Optuna
3) Treinamento da rede com os melhores hiperparâmetros com todos os dados de treinamento (dados do modelo matemático)
4) Avaliação dos “scores” com dados de treino e os dados de validação (Dados do modelo matemático)
5) Treinamento com os dados experimentais com e sem “transfer learn” e comparação com outros algoritmos
6) Comparação com o modelo de CUNHA et al, 2020

Fonte: Adaptado de SCIKIT-LEARN (2022).

4. RESULTADOS

Este capítulo tem como objetivo descrever detalhadamente as etapas de treinamento, validação e comparação com outros algoritmos, apresentando os resultados obtidos em cada etapa, bem como a melhor rede obtida.

4.1. DADOS

Os dados experimentais (sinais e concentrações) estão disponíveis no APÊNDICE A. Os dados referentes ao modelo matemático para o treinamento e validação iniciais não estão descritos neste documento devido ao seu tamanho (736 linhas) e os dados experimentais estão descritos na Tabela A-1, na Figura 4-1 e na Figura 4-2.

Figura 4-1 - Conjunto completo de dados experimentais.

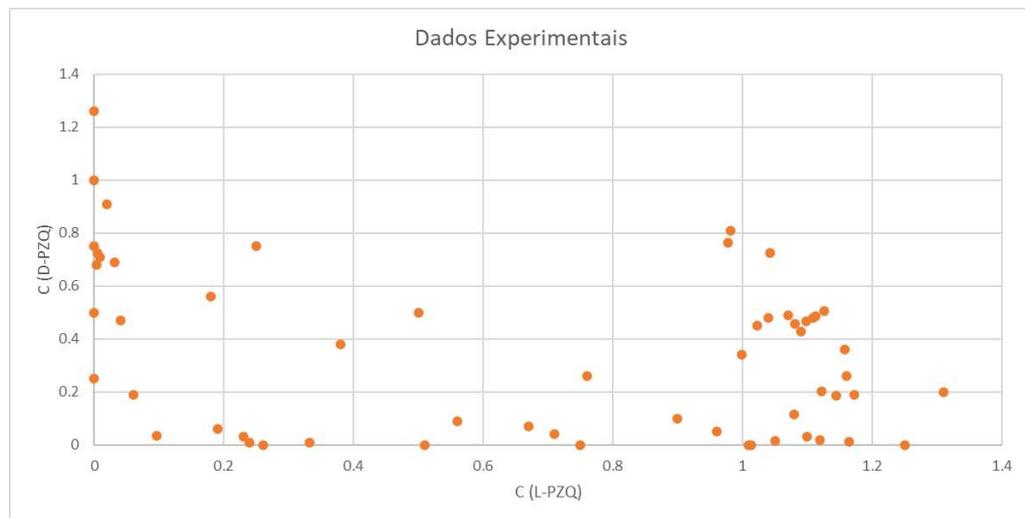
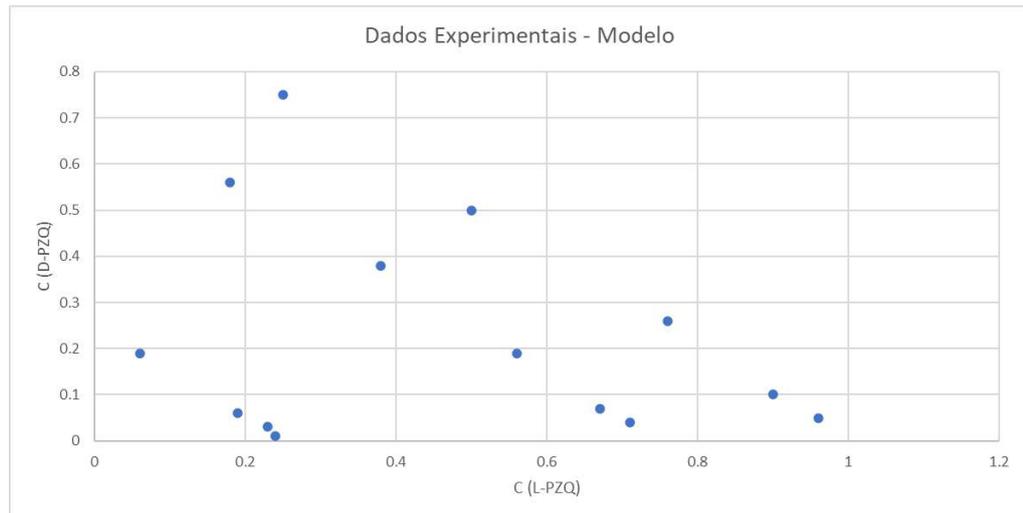


Figura 4-2 - Dados experimentais utilizados no modelo de CUNHA et al, 2020.



4.2. TREINAMENTO E VALIDAÇÃO

Esta etapa consistiu na divisão dos dados em duas categorias: Treinamento e Validação. Para realizar esta divisão foi utilizada uma ferramenta da biblioteca “Pandas”, permitindo a importação do arquivo .csv com os dados para treinamento e validação (PANDAS, 2022).

Para realizar a divisão dos dados em treinamento e validação, utilizou-se a função “train_test_split”, utilizando-se os seguintes argumentos (SCIKIT-LEARN, 2022):

- “Arrays”: argumento referente aos dados de entrada e saída. Neste caso, os dados de entrada seriam os sinais UV e CD e os sinais de saída são as concentrações de D-PZQ e L-PZQ. Vale ressaltar que os dados de entrada e saída podem ser armazenados em duas matrizes distintas, não havendo necessidade de utilizar quatro variáveis referentes a cada dado;
- “Test_size”: representa a porcentagem que será utilizada para o teste da rede, neste caso, é o conjunto de dados utilizados para a validação após o treinamento, utilizando-se um valor de 30%. Automaticamente, os outros 70% dos dados foram utilizados para treinar a rede;
- “Random_state”: controla o embaralhamento aplicado aos dados antes de dividi-los. Essa função permite dois tipos de valores: “none (default)” ou “integer”. No primeiro caso, se não for especificado um valor para o “random state”, sempre que a rotina for utilizada será gerado um novo conjunto de dados embaralhados. No segundo

caso, ao especificar o valor, a função garantirá que seja utilizado o mesmo conjunto de dados para diferentes execuções da rotina. Neste trabalho, optou-se por fixar um valor para o “random state”.

Após esta etapa de divisão entre os dados de treinamento e os dados de validação, estes foram normalizados utilizando abordagens “StandardScaler” e “MinMax”, como mencionado na Seção 3.2.

A partir dos dados de treinamento e validação normalizados, foi necessário estabelecer o valor que cada hiperparâmetro pode assumir para que a otimização da rede seja realizada. Assim, com base nos hiperparâmetros citados na Seção 3.2, foram estabelecidas as seguintes variáveis:

- “Numero_Maximo_Neuronios”: representa o número máximo de neurônios nas camadas escondidas (entrada manual);
- “Numero_Minimo_Neuronios”: representa o número mínimo de neurônios a ser utilizado nas camadas escondidas. Não é necessário que ele seja unitário, porém, deve ser maior que zero (entrada manual);
- “Numero_Camadas”: representa o número de camadas escondidas a serem consideradas na otimização (entrada manual);
- “Numero_Iteracoes”: número máximo de iterações que cada rede fará durante o processo de treinamento (entrada manual);
- “Funcao_ativacao”: lista contendo todas as funções de ativação disponíveis para o MLPRegressor;
- “Tipo_Solver”: lista contendo todos os solvers disponíveis para o MLPRegressor;
- “Aprendizado”: lista contendo todas as formas de “learning_rate” disponíveis para o MLPRegressor.

O principal objetivo dessa etapa é estabelecer as possibilidades de combinação que podem ser feitas pelo Optuna. No caso das variáveis referentes ao número máximo e mínimo de neurônios, elas são utilizadas como um limite superior e inferior para todas as topologias de redes possíveis nesse intervalo. O conjunto de todas as topologias é armazenado em uma variável denominada “Lista”. Neste trabalho, estabeleceram-se os valores da Tabela 4-1 para as variáveis de entrada manual:

Tabela 4-1 - Variáveis de entrada manual.

Variável	Valor
Numero_Maximo_Neuronios	2
Numero_Minimo_Neuronios	1
Numero_Camadas	2
Numero_Iteracoes	5000

A escolha desses valores foi feita com base em uma análise inicial sem otimização para avaliar como esses parâmetros influenciariam nos resultados. Além disso, a análise inicial também mostrou que redes com uma quantidade maior de neurônios ou de camadas têm uma tendência em piorar a predição do estimador. Com relação ao Optuna, ele se baseia na criação de uma função objetivo na qual são especificadas as possibilidades de hiperparâmetros e a métrica utilizada para otimização. É importante ressaltar que não há garantia que todas as possibilidades de parâmetros serão testadas. No entanto, o Optuna possui um recurso de retirar as tentativas que não são promissoras na otimização, permitindo que o número de tentativas aumente e mais possibilidades sejam percorridas. Além disso, optou-se pelo método de otimização bayesiana. A seguir encontra-se um detalhamento dos termos e valores dos argumentos utilizados para a otimização (OPTUNA, 2022), resumidos na Tabela 4-2:

- “Trial”: número que indica a execução da função objetivo, ou seja, qual é a tentativa que está sendo feita no momento. Atribuiu-se o valor 200 para este termo, ou seja, serão feitas 200 tentativas para otimizar os parâmetros.
- “Step”: na rotina de otimização este termo representa a divisão da validação cruzada. Neste trabalho, foram feitas 10 divisões dos dados de treinamento
- “Intermediate”: na rotina de otimização o valor do coeficiente de determinação de cada “step” em uma “trial” fica armazenado em uma matriz. No final de cada step, é calculada a média de todos esses coeficientes e armazenados na variável “Intermediate”
- “Study”: é o nome dado à otimização baseada em uma função objetivo. Recebe como argumentos “Direction” e “Pruner”. O termo “Direction” representa em que direção a função objetivo será otimizada, neste trabalho objetivou-se maximizar o R^2 médio, assim o valor inserido neste termo foi “maximize”. Já o “Pruner” se refere a uma tentativa que será interrompida com base em um determinado critério. O critério

utilizado foi o “MedianPruner”. Assim, a “trial” será interrompida caso o melhor valor da variável “Intermediate” dentro da “trial” seja pior que a mediana dos valores intermediários de tentativas anteriores no mesmo “step”. Para evitar que uma possível boa solução fosse interrompida na primeira divisão, o critério de corte só foi aplicado a partir do quarto “step”.

Tabela 4-2 - Resumo da rotina de otimização via Optuna e validação cruzada.

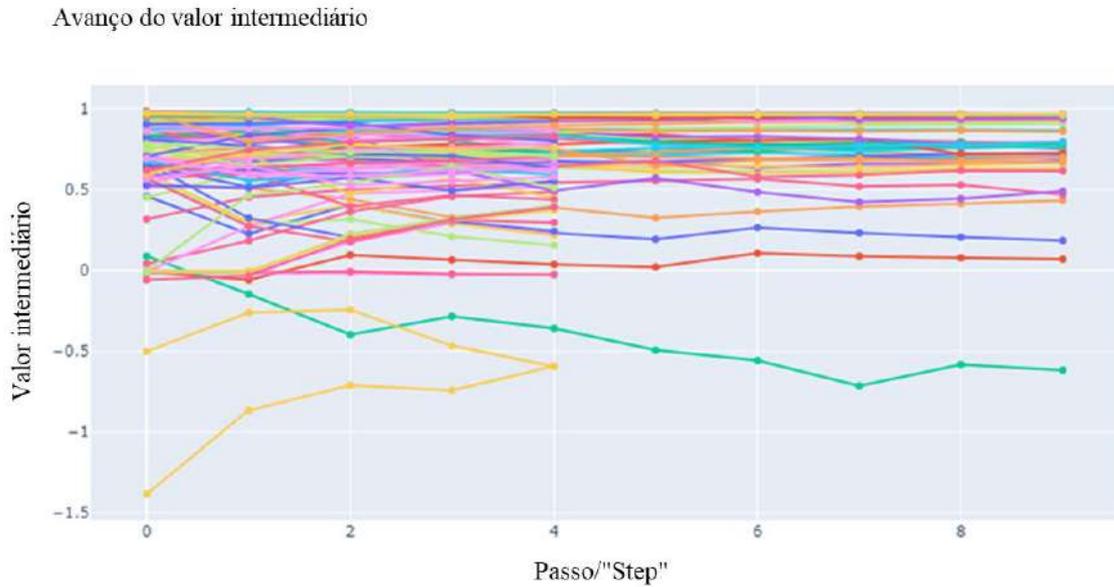
RESUMO DA ROTINA DE OTIMIZAÇÃO
1) A função objetivo recebe o número de tentativas e as informações para interromper tentativas não promissoras.
2) Escolha aleatória dos hiperparâmetros a partir das possibilidades possíveis
3) Construção do MLPRegressor a partir dos hiperparâmetros definidos da etapa 2
4) Determinação do R^2 para cada divisão da validação cruzada dos dados de treinamento.
5) Armazenamento dos R^2 individuais em uma matriz
6) Cálculo da média dos R^2 individuais de cada divisão
7) Armazenamento do valor da média na variável “Intermediate”
8) Avaliação da tentativa se será interrompida ou não usando o “MedianPruner”
9) Atualização da variável “step” e saída da função objetivo (Retorna o valor da média dos R^2)
10) Repetir até que o número de tentativas atinja o valor 200

Os resultados para a otimização da rede utilizando a normalização de média com desvio padrão estão representados em cinco tipos de gráficos descritos a seguir:

- Avanço do valor intermediário, Figura 4-3: representa a evolução da variável “Intermediate” em uma “trial” em cada “step”. Nesse caso, a evolução da média dos coeficientes de determinação dentro de cada tentativa. Cada linha no gráfico a seguir representa uma “trial” e cada ponto representa o valor da média em um “step”. As linhas

que foram interrompidas representam os resultados cortados com base no critério de interrupção estabelecido. Vale ressaltar que ele só passou a ser aplicado no step 4.

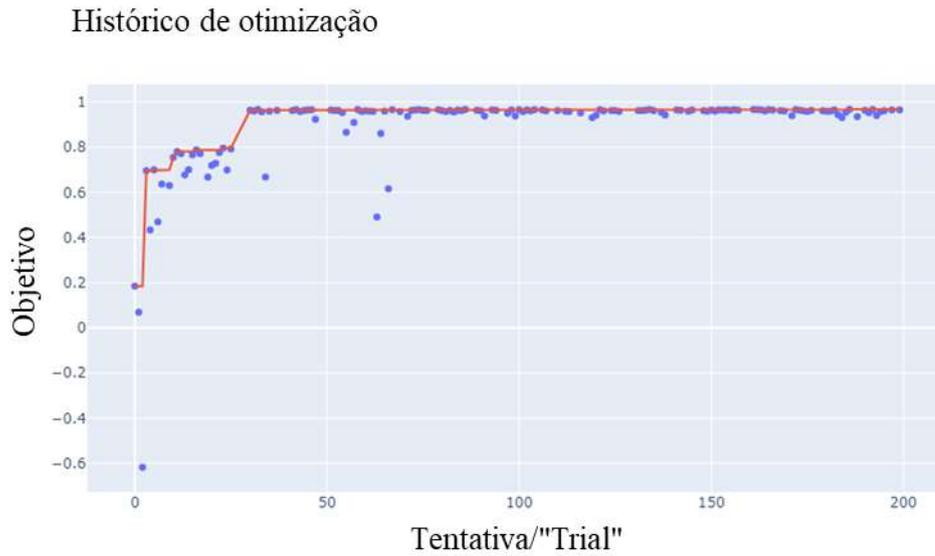
Figura 4-3 - Evolução da variável “Intermediate” em função do “step”.



Fonte: Adaptado de OPTUNA (2022).

- Histórico de otimização, Figura 4-4: neste gráfico estão representados os valores da função objetivo (Média) no final de cada tentativa, ou seja, ao completar a validação cruzada para um dado conjunto de hiperparâmetros. A linha vermelha representa a evolução do melhor valor obtido até uma dada tentativa e os pontos azuis representam o valor da média no final de cada tentativa.

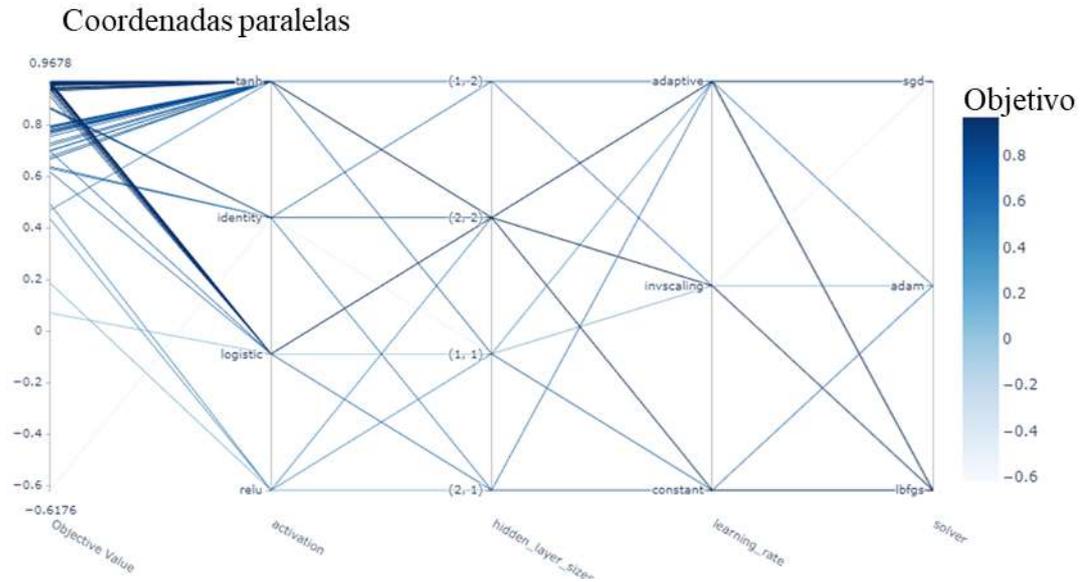
Figura 4-4 - Evolução do valor ótimo.



Fonte: Adaptado de OPTUNA (2022).

- Coordenadas paralelas, Figura 4-5: Este gráfico permite uma rápida visualização de qual foi o melhor conjunto de hiperparâmetros. No eixo horizontal estão representados os hiperparâmetros, enquanto nas linhas verticais estão representadas as possibilidades de escolha para cada um deles. A intensidade da cor azul de cada linha indica o quão próximo do objetivo cada conjunto ficou. A principal característica desse gráfico é evidenciar outros possíveis candidatos para o conjunto ótimo de hiperparâmetros. Vale ressaltar que a otimização convergiu em poucas tentativas.

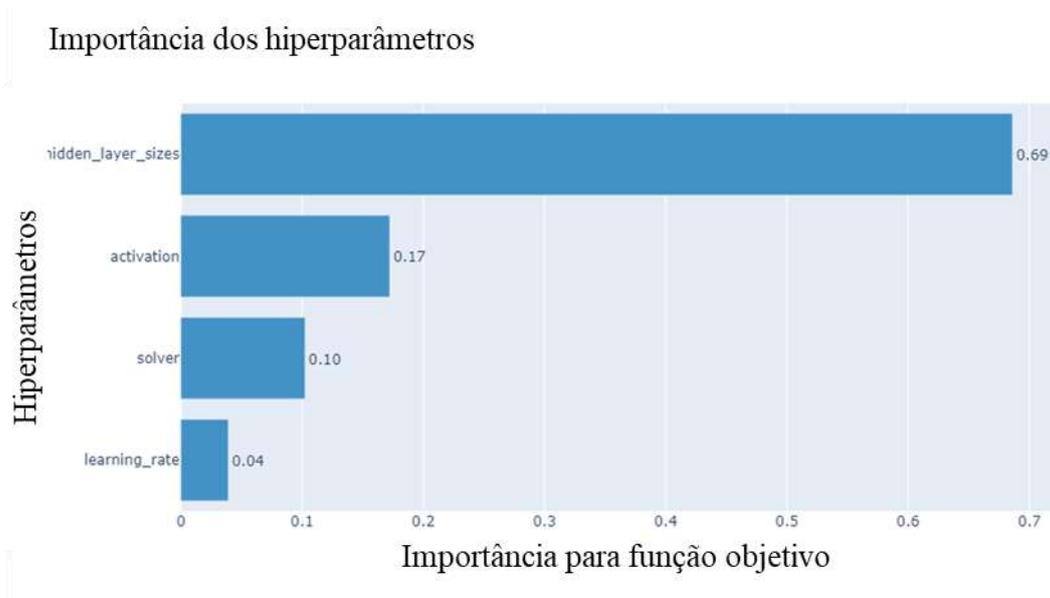
Figura 4-5 – Coordenadas paralelas.



Fonte: Adaptado de OPTUNA (2022).

- Importância dos hiperparâmetros, Figura 4-6: Este gráfico representa o impacto de cada hiperparâmetro no desempenho da rede.

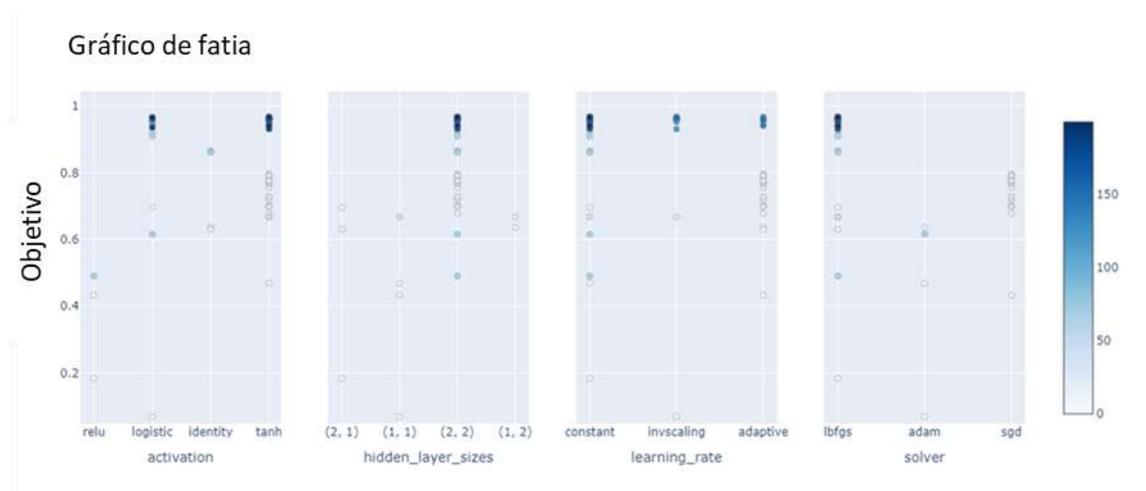
Figura 4-6 - Importância de cada hiperparâmetro para o valor da função objetivo.



Fonte: Adaptado de OPTUNA (2022).

- Gráfico de fatia, Figura 4-7: Este gráfico indica a contribuição de cada hiperparâmetro para um dado valor obtido da função objetivo. Também garante uma rápida visualização dos possíveis hiperparâmetros ótimos.

Figura 4-7 - Gráfico de fatia.



Fonte: Adaptado de OPTUNA (2022).

Ao final de cada “trial”, a rede obtida é salva, permitindo que a rede ótima seja utilizada posteriormente para a obtenção dos hiperparâmetros, que se encontram listados nas Tabela 4-3 a Tabela 4-5, bem como os “scores” da rede ótima com os dados completos de treino e validação:

Tabela 4-3 - Conjunto ótimo de hiperparâmetros.

Topologia (Hidden_Layer_Sizes)	Ativação	Solver	Aprendizado	Score treinamento	Score validação
(2,2)	tanh	lbfgs	constant	0,9692	0,9702

Tabela 4-4 - Pesos da rede otimizada.

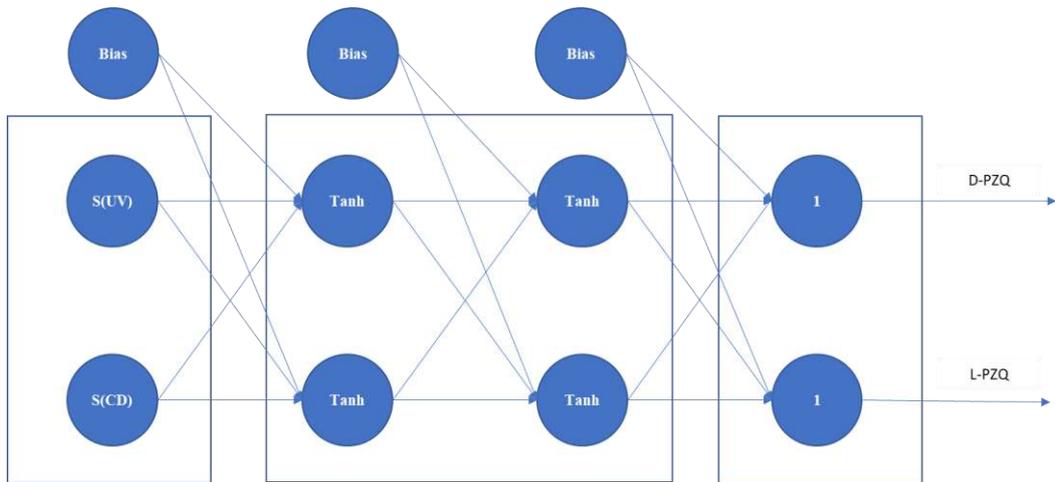
Região	Pesos
Entrada e primeira camada escondida	[0,0588; -0,4115; 0,1839; 0,1139]
Primeira e segunda camada escondida	[-1,2984; -5,4753; -0,15411; -10,6683]
Segunda camada escondida e saída	[-10,2112; 0,8381; 0,3993; -3,4110]

Tabela 4-5 - Bias da rede otimizada.

Região	Bias
Primeira camada escondida	[-0,8816; -1,9345]
Segunda camada escondida	[-1,2986; -14,7642]
Camada de saída	[-2,2638; 2,4209]

A Figura 4-8 ilustra a topologia da rede obtida:

Figura 4-8 - Ilustração da rede neuronal obtida.



Antes de iniciar o treinamento com os dados experimentais, é necessário repetir todo o procedimento de pré-processamento. Neste caso, os dados também são normalizados utilizando o desvio padrão e a média.

Para implementar o conceito de “Transfer Learn”, foi necessário alterar a função “MLPRegressor” utilizando o método “Override”. Este método permite que se altere um trecho específico de uma função previamente existente, mas sem afetar o resto dela (SCIKIT-LEARN, 2022). Assim, no trecho correspondente à inicialização dos pesos e bias do estimador, foram introduzidos os valores presentes Tabela 4-4 e na Tabela 4-5. Na sequência, os outros hiperparâmetros foram passados para a rede. Esse procedimento foi idêntico ao da rede sem “transfer learn”, mas nesse caso as funções de inicialização foram mantidas.

Após o treinamento das redes com os dados experimentais e de validação foram obtidos os resultados da Tabela 4-6:

Tabela 4-6 - Avaliação do desempenho do “Transfer Learn” com os dados experimentais.

Rede	Score Treinamento	Score Validação
Transfer Learn	0,8850	0,7786
Sem Transfer Learn	0,8129	0,8281

4.3. COMPARAÇÃO COM OUTROS ALGORITMOS

Nesta seção serão abordados outros algoritmos e formas distintas de realizar o treinamento e validação dos estimadores. É importante ressaltar que a metodologia é a mesma, mas com algumas modificações pontuais no pré-processamento e treinamento. Na Tabela 4-7 encontra-se uma descrição das diferentes abordagens feitas neste trabalho. A abordagem “Independente” faz referência à utilização de dois estimadores especializados em prever a concentração de cada enantiômero. A abordagem denominada como “Padrão” consiste na obtenção de um estimador para prever a concentração dos dois enantiômeros de forma simultânea. A combinação MLPRegressor x StandardScaler x Padrão não foi colocada na tabela, pois foi considerada como sendo a abordagem de referência.

Tabela 4-7 - Diferentes abordagens utilizadas para a obtenção do estimador.

Índice	Estimador	Normalização	Abordagem
1	MLPRegressor	StandardScaler	Independente
2	MLPRegressor	MinMaxScaler	Independente
3	MLPRegressor	MinMaxScaler	Padrão
4	RandomForest	StandardScaler	Padrão
5	RandomForest	StandardScaler	Independente
6	RandomForest	MinMaxScaler	Independente
7	RandomForest	MinMaxScaler	Padrão

Os resultados da otimização para cada abordagem estão no APÊNDICE B. As Tabela 4-8 e Tabela 4-9 apresentam os resultados correspondentes às diferentes abordagens. A escolha

em dividir os resultados em duas tabelas é para facilitar a comparação entre as redes independentes e as redes do tipo “padrão”. É importante ressaltar que a rede 1 será considerada como a referência entre as redes de abordagem independente.

A sigla “TL” indica “Transfer Learn” enquanto a sigla “STL” indica “Sem Transfer Learn”. No caso das redes independentes, utilizaram-se os índices A e B para indicarem as saídas individuais correspondentes ao D-PZQ e L-PZQ, respectivamente.

Tabela 4-8 - Resultados para as redes do tipo "Padrão" (duas saídas em conjunto).

Índice	Score Treinamento	Score Validação
Referência TL	0,8850	0,7786
Referência STL	0,8129	0,8281
3-TL	0,8628	0,5027
3-STL	0,8171	0,5353
4	0,8933	0,6611
7	0,8869	0,5982

Tabela 4-9 - Resultados para as redes do tipo "Independente".

Índice	Score Treinamento	Score Validação
1-A-TL	0,8656	0,2975
1-A-STL	0,7678	0,4414
1-B-TL	0,9497	0,9104
1-B-STL	0,9529	0,8933
2-A-TL	0,7407	0,2734
2-A-STL	0,7462	0,28755
2-B-TL	0,9229	0,9173
2-B-STL	0,9550	0,8992
5-A	0,8561	0,4064
5-B	0,9490	0,8074
6-A	0,8566	0,5240
6-B	0,9498	0,8012

A Tabela 4-10 apresenta os resultados obtidos a partir da rede otimizada diretamente com os dados experimentais:

Tabela 4-10 - Resultados da rede otimizada diretamente com os dados experimentais.

Topologia (Hidden_Layer_Sizes)	Ativação	Solver	Aprendizado	“Score” treinamento	“Score” validação
(2,2)	tanh	lbfgs	adaptative	0,7058	0,7712

A Tabela 4-11 apresenta uma comparação dos resultados obtidos, usando todos os dados experimentais, da melhor rede dentre todas e do modelo proposto por CUNHA et al, 2020.

Tabela 4-11 - Comparação com os resultados do modelo de CUNHA,2021.

Estimador	“Score”
Modelo CUNHA,2021	0,54
Rede	0,82

4.4. CONCLUSÕES DO CAPÍTULO

Com relação às redes do tipo “Independentes”, observou-se uma dificuldade em realizar o ajuste para o D-PZQ. Apesar dos bons resultados na etapa de treinamento houve uma queda significativa no desempenho da rede para os dados de validação. Abaixo estão listados alguns pontos que podem ter contribuído para a queda de desempenho da rede para o ajuste do D-PZQ:

- Possível sobreajuste com os dados de treinamento;
- Dificuldade intrínseca da rede em determinar a concentração do D-PZQ. É possível que o ajuste correspondente a ele seja naturalmente mais difícil;
- O uso dos dados de um modelo matemático para obtenção dos pesos da rede para ser utilizada no “Transfer Learn” pode ter prejudicado o desempenho desta técnica, uma vez que já existem erros associados ao próprio modelo matemático.

No caso do L-PZQ, todas as redes independentes conseguiram ajustes com bom desempenho.

Comparando o algoritmo “MLPRegressor” e o “RandomForest”, observou-se que o “MLPRegressor” realiza ajustes melhores para o L-PZQ em relação ao “RandomForest”. No entanto, o “RandomForest” acaba sendo melhor nas predições do D-PZQ, apesar de ambos não terem um bom desempenho na predição do D-PZQ.

Em relação às normalizações, a “StandardScaler” apresentou resultados melhores quando comparada à “MinMaxScaler” no “MLPRegressor”. No caso do “RandomForest”, a “MinMaxScaler” foi melhor

O “Transfer Learn”, no geral, não apresentou uma melhora significativa em relação à abordagem padrão (sem “Transfer Learn”). O uso do “Transfer Learn” piorou o desempenho dos estimadores na predição da concentração do D-PZQ, algo que corrobora a hipótese de sobreajuste da rede com os dados de treino. Ao utilizar a transferência de aprendizagem, a característica da rede obtida durante o treinamento torna-se ainda mais impactante no momento da validação, potencializando o problema de sobreajuste.

Dessa forma, para os estimadores com abordagens “Independentes”, conclui-se que a combinação mais adequada seria a seguinte:

- Estimador: “MLPRegressor” para o L-PZQ e “RandomForest” para o D-PZQ;
- Normalização: “StandardScaler” para o “MLPRegressor” e “MinMaxScaler” para o “RandomForest”;
- Não utilizar a abordagem de “Transfer Learn” para evitar a propagação dos erros associados ao ajuste do D-PZQ.

É importante ressaltar que a utilização da abordagem “Independente” se mostrou promissora para determinação da concentração do L-PZQ.

No caso dos estimadores do tipo “Padrão”, alguns resultados foram semelhantes ao da abordagem anterior. A normalização “StandardScaler” apresentou melhor desempenho em relação à normalização “MinMaxScaler”. Além disso, o uso do “Transfer Learn” também piorou o desempenho da rede, destacando a possibilidade de sobreajuste devido ao uso do modelo matemático como base para treinamento inicial. Em relação ao “MLPRegressor”, ele teve um desempenho melhor que o “RandomForest”. Dessa forma a composição mais adequada para utilização neste caso seria:

- Estimador: “MLPRegressor”;

- Normalização: “StandardScaler”;
- Sem “Transfer Learn”.

É importante destacar que a quantidade relativamente pequena de dados experimentais pode ter influenciado o desempenho dos estimadores, o que não exclui as outras abordagens em relação a considerada ótima citada acima. Além disso, as redes neuronais apresentaram um melhor desempenho em relação ao modelo proposto por CUNHA et al (2020). Assim, justifica-se a piora no desempenho observada ao utilizar o “Transfer Learn”, uma vez que o modelo matemático empírico possui mais influência na rede que foi inicializada com os pesos obtidos dos dados do modelo.

Vale destacar que a rede otimizada diretamente com os dados experimentais também apresentou um bom desempenho, destacando que a menor influência do modelo empírico garante um aumento no desempenho da rede. Ao comparar esta rede com a melhor, os resultados são relativamente próximos, mas a diferença pode ser justificada pela diferença no tipo de taxa de aprendizagem de cada rede.

A Tabela 4-12 resume os resultados obtidos da melhor rede, do “Random Forest” e do modelo polinomial com os dados experimentais.

Tabela 4-12 - Resumo dos resultados obtidos.

Estimador	“Score” - dados completos	“Score” - validação
Modelo CUNHA,2021	0,54	0,59
Rede	0,82	0,83
“Random Forest”	0,82	0,66

É importante ressaltar que, apesar do “Random Forest” possuir um desempenho próximo ao da rede neuronal utilizando todos os dados experimentais, a rede apresentou resultados mais consistentes ao longo do treinamento e validação. Assim, optou-se por adotar a rede neuronal como sendo o estimador de melhor desempenho.

Para comparar o desempenho da rede com o modelo polinomial, foram feitas a Figura 4-9 e a Figura 4-10.

Figura 4-9 - Gráficos comparativos com todos os dados experimentais.

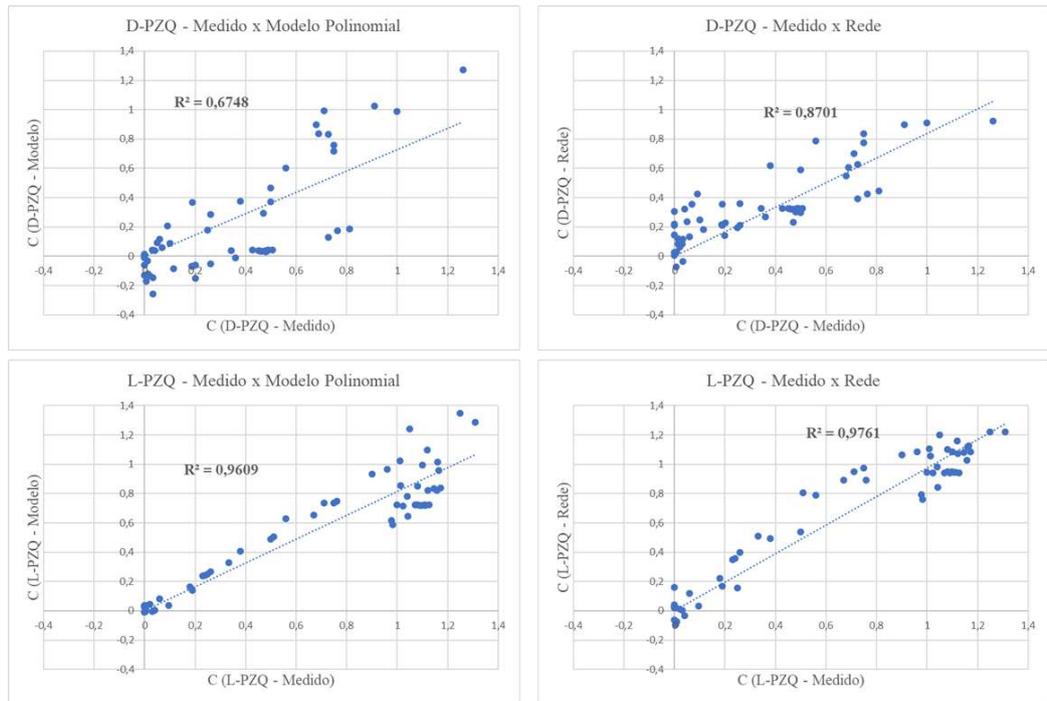
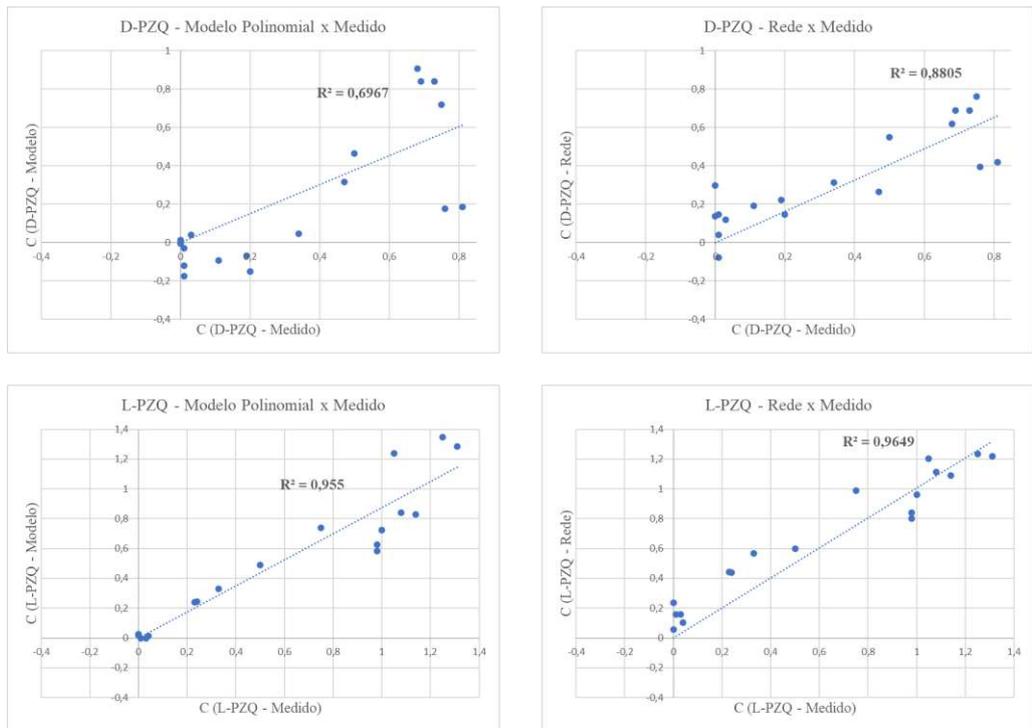


Figura 4-10 - Gráficos comparativos com os dados de validação.



5. CONCLUSÕES E SUGESTÕES

Neste trabalho foram apresentados conceitos associados ao uso de uma unidade de Leito Móvel Simulado (LMS) para separação dos enantiômeros de praziquantel, o principal medicamento utilizado para tratamento de esquistossomose. Como apenas um dos enantiômeros é considerado como composto ativo, enquanto o outro traz características negativas para a administração oral do medicamento (amargor), o monitoramento em linha dessas concentrações torna-se essencial para o desenvolvimento do controle em malha fechada, buscando atingir a condição ótima de operação.

O trabalho de CUNHA (2021) mostrou que a correlação entre os sinais obtidos de um detector quiral e as concentrações dos enantiômeros é não linear. Essa característica indica um grande potencial na utilização das redes neuronais para a predição da concentração a partir dos sinais de dicroísmo circular e UV-vis provenientes do detector quiral.

Como as bibliotecas atuais fornecem inúmeras combinações de hiperparâmetros, surge naturalmente a necessidade de encontrar os hiperparâmetros ótimos para a rede neuronal. Assim, neste trabalho, utilizou-se o software Optuna para otimização. A escolha dele se deu devido aos algoritmos eficientes de busca e os amplos recursos gráficos para estudo da otimização.

As redes neuronais se destacam por encontrar padrões e tendências em um espaço amostral grande de dados para treinamento. Assim, um dos principais desafios associados a este trabalho foi encontrar uma metodologia capaz de contornar a pequena quantidade de dados experimentais disponíveis. Dessa maneira, optou-se por treinar a rede inicialmente com os dados do modelo matemático de CUNHA (2021) que funcionou como a base da rotina de otimização.

Apesar da disponibilidade de rotinas de otimização do Optuna com validação cruzada, estas não atenderiam alguns pontos relevantes para a obtenção dos hiperparâmetros ótimos, como por exemplo a combinação de diversas topologias. Assim, foi necessário realizar algumas modificações dentro da rotina de otimização, o que destaca a versatilidade e adaptabilidade do software escolhido, permitindo que o usuário realize ajustes conforme a sua necessidade.

Para avaliar o desempenho das redes neuronais, também foram comparados os resultados obtidos por diferentes algoritmos e diferentes abordagens. Foram feitas combinações

entre os estimadores “MLPRegressor”, “RandomForest” e as normalizações “StandardScaler” e “MinMaxScaler”. Além disso, também foi utilizado o conceito de “Transfer Learn”, buscando aproveitar a rede resultante do treinamento com o modelo matemático, e a predição independente das concentrações do D-PZQ e L-PZQ, utilizando uma rede neuronal para cada enantiômero.

Com relação aos resultados obtidos, foi possível concluir que as redes neuronais não necessitaram de uma topologia muito complexa para obter resultados promissores. Dentre as diferentes comparações entre os algoritmos, a que mais se destacou foi aquela representada pelo estimador “MLPRegressor”, normalização “StandardScaler” e predição conjunta das duas concentrações de enantiômeros. É importante ressaltar que neste trabalho observou-se que a predição da concentração do D-PZQ é mais difícil em relação ao L-PZQ. Para este último, as redes neuronais, em alguns casos, obtiveram um R^2 superior a 0,9.

A otimização da rede diretamente com os dados experimentais apresentou um melhor desempenho em relação ao modelo de CUNHA (2021). Apesar de não ter sido a melhor rede, o seu resultado está relativamente próximo da melhor rede obtida.

A comparação de resultados entre o modelo de CUNHA (2021) e as redes neuronais evidencia o potencial de aplicação destas redes no monitoramento em linha da concentração de enantiômeros de PZQ. Além disso, no modelo de CUNHA (2021), é necessário obter as concentrações utilizando um solver a partir dos sinais. Este procedimento pode resultar em raízes sem significado físico. No caso das redes neuronais, as concentrações são obtidas diretamente a partir dos sinais de CD e UV-Vis. Dessa forma, as redes neuronais são adequadas para o monitoramento em linha, devido à capacidade preditiva verificada que ainda pode ser melhorada ao disponibilizar mais dados para treinamento da rede. O “RandomForest” também possuiu um desempenho muito próximo da rede neuronal quando a predição foi feita com todos os dados experimentais, mas apresentou resultados menos consistentes em relação ao treinamento e validação.

Como sugestões para o desenvolvimento de trabalhos futuros nesta linha:

- Utilização de pipelines, uma ferramenta que permite aplicar uma série de transformações e um estimador final. Com essa ferramenta, é possível implementar múltiplos métodos de normalização, por exemplo, possibilitando uma análise mais rápida do efeito de todos esses métodos;

- Ampliar o número de hiperparâmetros a serem utilizados na otimização, utilizando os recursos do Optuna para evitar um aumento excessivo no custo computacional;
- Estudar a dificuldade dos estimadores em prever a concentração do D-PZQ, buscando entender se isso é uma dificuldade intrínseca do estimador ou se é resultado de incertezas experimentais;
- Investigar o uso do “RandomForest” para a predição independente do D-PZQ, pois o seu desempenho foi melhor que o “MLPRegressor”, apesar de ainda estar longe do ideal.

6. REFERÊNCIAS

ABIRAMI, S.; CHITRA, P. **Advances in Computers**. Academic Press (2020).

ALOM, M.Z.; TAHA, T.M. et al. **A State-of-the-Art Survey on Deep Learning Theory and Architectures**. *Electronics*, 8, 292 (2019).

ANDRADE NETO, A.S. **MODELING, CONTROL AND OPTIMIZATION OF THE ENANTIOMERIC SEPARATION OF PRAZIQUANTEL IN SIMULATED MOVING BED**. Dissertação (Mestrado em Engenharia Química) - COPPE/PEQ/UFRJ, Rio de Janeiro, RJ (2015).

ANDRADE NETO, A.S.; SECCHI, A.R.; SOUZA JR., M.B.; BARRETO JR., A.G. **Nonlinear model predictive control applied to the separation of praziquantel in simulated moving bed chromatography**. *Journal of Chromatography A*, 1470, 42–49 (2016). DOI: 10.1016/j.chroma.2016.09.070.

BAUGHMAN, D.R.; LIU, Y.A. **Neural Networks in Bioprocessing and Chemical Engineering**. Academic Press (2014).

BELGIU, Mariana; DRĂGUȚ, Lucian. **Random forest in remote sensing: A review of applications and future directions**. *ISPRS Journal of Photogrammetry and Remote Sensing*, 114, 24–31 (2016). DOI: 0.1016/j.isprsjprs.2016.01.011.

BROWN, Sara. **Machine learning, explained**. 21 abr. 2021. Disponível em: <https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained>. Acesso em: 20 nov. 2022.

CUNHA, Felipe C. et al. **Separation of praziquantel enantiomers using simulated moving bed chromatographic unit with performance designed for semipreparative applications**. *Chirality*, 31, 583–591 (2019). DOI: 10.1002/chir.23084.

CUNHA, F.C. **Chromatographic separation of praziquantel racemate using simulated moving bed: from unit design to dynamic studies with online measurements**, Tese de Doutorado, COPPE/PEQ/UFRJ, Rio de Janeiro, RJ (2021).

CUNHA, F.C. et al. **Simultaneous absorption of UV-vis and circular dichroism to measure enantiomeric concentrations of praziquantel under nonlinear conditions.** *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, 241, 118645 (2020). DOI: 10.1016/j.saa.2020.118645.

DAVIES, Joseph C.; PATTISON, David; HIRST, Jonathan D. **Machine learning for yield prediction for chemical reactions using in situ sensors.** *Journal of Molecular Graphics and Modelling*, 118, 108356 (2023). DOI 10.1016/j.jmgm.2022.108356.

DE HOLANDA, R.C. et al. **Supervisory system for automated simulated moving bed (SMB) liquid chromatography (LC).** *Instrumentation Science & Technology - Chromatography*, 1-16 (2022). DOI: 10.1080/10739149.2022.2128371.

DOENHOFF, Michael J. et al. **Resistance of *Schistosoma mansoni* to praziquantel: is there a problem?** *Transactions of The Royal Society of Tropical Medicine and Hygiene*, 96 (5), 465-469 2002. DOI: 10.1016/S0035-9203(02)90405-0.

DRAKE, Alex F. et al. **Simultaneous monitoring of light-absorption and optical activity in the liquid chromatography of chiral substance.** *Journal of Chromatography A*, 202 (2), 239-245 (1980). DOI: 10.1016/S0021-9673(00)81732-X.

DRUGS for Helminths. In: BENNETT, J.E.; DOLIN, R.; BLASER, M.J. **Mandell, Douglas, and Bennett's Principles and Practice of Infectious Diseases.** 9. ed. Elsevier, v, 2, 544-555 (2020).

DE SOUZA JR, M.B., **Redes Neurais Multicamadas Aplicadas à Modelagem e Controle de Processos Químicos**, Tese de Doutorado, COPPE/PEQ/UFRJ, Rio de Janeiro, RJ (1993).

FERREIRA, L.S. et al. **Development of an alcohol fermentation control system based on biosensor measurements interpreted by neural networks.** *Sensors and Actuators B: Chemical*, 75 (3), 166-171 (2001). DOI: 10.1016/S0925-4005(01)00540-8.

FIOCRUZ. **Dia Mundial das Doenças Tropicais Negligenciadas.** Rio de Janeiro, RJ (2022). Disponível em: <https://www.bio.fiocruz.br/index.php/br/noticias/2794-dia-mundial-das-doencas-tropicais-negligenciadas>. Acesso em: 27 out. 2022.

GOODFELLOW, Ian et al. **Deep Learning.** MIT Press (2016). Disponível em: <http://www.deeplearningbook.org>. Acesso em: 8 dez. 2022.

GUPTA, A. **Introduction to Deep Learning** (2018). Disponível em: <https://www.iche.org/resources/publications/cep/2018/june/introduction-deep-learning-part-1>. Acesso em: 21 nov. 2022.

HONGXIANG, Tian; YUNTAO, Liu; XIANGJUN, Wu. **Application of Neural Network to Diesel Engine SOA**. Third International Conference on Measuring Technology and Mechatronics Automation (2011). DOI: 10.1109/ICMTMA.2011.141.

HURLBURT, Nick. **Circular Dichroism** (2022). Disponível em: [https://chem.libretexts.org/Bookshelves/Physical_and_Theoretical_Chemistry_Textbook_Maps/Supplemental_Modules_\(Physical_and_Theoretical_Chemistry\)/Spectroscopy/Electronic_Spectroscopy/Circular_Dichroism](https://chem.libretexts.org/Bookshelves/Physical_and_Theoretical_Chemistry_Textbook_Maps/Supplemental_Modules_(Physical_and_Theoretical_Chemistry)/Spectroscopy/Electronic_Spectroscopy/Circular_Dichroism). Acesso em: 27 nov. 2022.

KINGMA, Diederik P.; BA, Jimmy Lei. **ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION**. ICLR (2015). Disponível em: https://www.researchgate.net/publication/269935079_Adam_A_Method_for_Stochastic_Optimization. Acesso em: 7 dez. 2022.

LEONARD, J.; KRAMER, M.A. **Improvement of the backpropagation algorithm for training neural networks**. Computers & Chemical Engineering, 14 (3), 337-341 (1989). DOI: 10.1016/0098-1354(90)87070-6.

LI, Weijun et al. **Transfer learning for process fault diagnosis: Knowledge transfer from simulation to physical processes**. Computers & Chemical Engineering, 139, 106904 (2020). DOI: 10.1016/j.compchemeng.2020.106904.

MEYER, Thorsten et al. **Taste, a new incentive to switch to (R)-praziquantel in schistosomiasis treatment**. PLoS Neglected Tropical Diseases, 3 (1), e357 (2009). DOI: 10.1371/journal.pntd.0000357.

MINISTÉRIO DA SAÚDE. **Diretrizes Técnicas – Vigilância da Esquistossomose Mansoni** (2014). Disponível em: https://bvsm.sau.gov.br/bvs/publicacoes/vigilancia_esquistossome_mansoni_diretrizes_tecnicas.pdf. Acesso em: 27 out. 2022

MINISTÉRIO DA SAÚDE. **Educação em Saúde para o controle da esquistossomose** (2018). Disponível em:

https://bvsmms.saude.gov.br/bvs/publicacoes/educacao_saude_controle_esquistossomose.pdf.

Acesso em: 27 out. 2022

MINISTÉRIO DA SAÚDE. **Esquistossomose (2022)**. Disponível em: <https://www.gov.br/saude/pt-br/assuntos/saude-de-a-a-z/e/esquistossomose#:~:text=O%20que%20%C3%A9%20Esquistossomose%3F,pe%C3%A7as%20vermes%20causadores%20da%20esquistossomose>. Acesso em: 27 out. 2022.

NOGUEIRA PEREIRA, J.N, **Fault detection and diagnosis of a two-column sour water treatment unit based on artificial intelligence algorithms**, Tese de Mestrado, EPQB/EQ/UFRJ, Rio de Janeiro, RJ (2021).

OPTUNA. **Key Features (2022)** Disponível em: <https://optuna.readthedocs.io/en/stable/index.html>. Acesso em: 12 nov. 2022.

PANDAS. **PANDAS (2022)**. Disponível em: <https://pandas.pydata.org/>. Acesso em: 11 dez. 2022.

RAJENDRAN, Arvind; PAREDES, Galatea; MAZZOTTI, Marco. **Simulated moving bed chromatography for the separation of enantiomers**. The Journal of Chromatography A, 1216 (4), 709-738 (2009). DOI: 10.1016/j.chroma.2008.10.075.

Raymond L. Watrous, **Learning Algorithms for Connectionist Networks: Applied Gradient Methods of Nonlinear Optimization** (1988).

RODRIGUES, João Antunes et al. **Comparison of Different Features and Neural Networks for Predicting Industrial Paper Press Condition**. Energies, 15 (17), 6308 (2022). DOI: 10.3390/en15176308.

RUMELHART, D.E.; HINTON, G.E.; WILLIAMS, R.J. **Learning representations by back-propagating errors**. Nature, 323, 533–536 (1986). DOI: 10.1038/323533a0.

RUMELHART, D.E.; MCCLELLAND, J.L., **Parallel Distributed Processing: Explorations in the Microstructure of Cognition**, Volume 1: Foundations, The Mit Press, Cambridge (1986).

SAXE, Andrew M.; MCCLELLAND, James L.; GANGULI, Surya. **Learning hierarchical category structure in deep neural networks**. Cognitive Science Society (2013). Disponível em: <https://stanford.edu/~jlmcc/papers/SaxeMcCGanguli13CogSciProc.pdf>. Acesso em: 26 nov. 2022.

SCIKIT-LEARN. **Cross validation** (2022). Disponível em: https://scikit-learn.org/stable/modules/cross_validation.html. Acesso em: 01 dez. 2022.

SCIKIT-LEARN. **MLPRegressor** (2022). Disponível em: https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html. Acesso em: 11 dez. 2022.

SCIKIT-LEARN. **Preprocessing data** (2022). Disponível em: <https://scikit-learn.org/stable/modules/preprocessing.html#preprocessing>. Acesso em: 05 nov. 2022.

SCIKIT-LEARN. **Pipeline** (2022). Disponível em: <https://scikit-learn.org/stable/modules/generated/sklearn.pipeline.Pipeline.html>. Acesso em: 28 nov. 2022.;

SCIKIT-LEARN. **RandomForestRegressor** (2022). Disponível em: <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html>. Acesso em: 01 dez. 2022.;

SCIKIT-LEARN. **StandardScaler** (2022). Disponível em: <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html>. Acesso em: 11 dez. 2022.

SCIKIT-LEARN. **Train_test_split** (2022). Disponível em: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html. Acesso em: 11 dez. 2022.

SPEISER, Jaime Lynn et al. **A comparison of random forest variable selection methods for classification prediction modeling**. Expert Systems with Applications, [s. l.], 15 nov. 2019. DOI <https://doi.org/10.1016/j.eswa.2019.05.028>. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0957417419303574>. Acesso em: 11 dez. 2022.

SUPELANO, Reinaldo Calderón et al. **Evaluation of the optimal performance of ModiCon and ModiCon+VariCol simulated moving bed variants**. Journal of Chromatography A, 1675, 463182-463191 (2022). DOI: 10.1016/j.chroma.2022.463182.

SUPELANO, Reinaldo Calderón et al. **One-step optimization strategy in the simulated moving bed process with asynchronous movement of ports: A VariCol case study**. Journal of Chromatography A, 1634, 461672-461682 (2020). DOI: 10.1016/j.chroma.2020.461672.

SUPELANO, Reinaldo Calderón et al. **Optimal performance comparison of the simulated moving bed process variants based on the modulation of the length of zones and the feed concentration.** *Journal of Chromatography A*, 1651, 462280-462292 (2021). DOI: 10.1016/j.chroma.2021.462280.

WOODY, R. W. **Circular-Dichroim.** *Methods in Enzymology* **246**, 34-71 (1995).

APÊNDICE A

Dados experimentais utilizado para o treinamento e validação da rede. O índice “A” indica o D-PZQ e o índice “B” indica o L-PZQ:

Tabela A-1 - Dados experimentais utilizados para treinamento e validação da rede.

S_UV (abs)	S_CD (mdeg)	C_D-PZQ (g/L)	C_L-PZQ (g/L)
0,08	2,3	0,00	0,26
0,13	3,7	0,00	0,51
0,18	4,8	0,00	0,75
0,23	6,0	0,00	1,01
0,28	6,7	0,00	1,25
0,06	-1,1	0,25	0,00
0,09	-1,5	0,50	0,00
0,13	-2,7	0,75	0,00
0,16	-3,7	1,00	0,00
0,19	-4,2	1,26	0,00
0,08	1,9	0,01	0,24
0,09	1,4	0,03	0,23
0,08	0,2	0,06	0,19
0,10	-1,3	0,19	0,06
0,19	4,5	0,04	0,71
0,18	3,9	0,07	0,67
0,22	2,3	0,09	0,56
0,19	-0,2	0,38	0,38
0,15	-2,3	0,56	0,18
0,27	4,9	0,05	0,96
0,26	4,8	0,10	0,90
0,29	2,0	0,26	0,76
0,24	-0,7	0,50	0,50
0,20	-3,3	0,75	0,25
0,11	-3,7	0,71	0,01
0,15	7,6	0,02	1,12

0,20	4,8	0,48	1,04
0,08	-1,4	0,47	0,04
0,19	6,6	0,26	1,16
0,10	-3,3	0,68	0,00
0,13	7,5	0,03	1,10
0,13	-3,9	0,91	0,02
0,16	7,7	0,01	1,05
0,11	-3,0	0,73	0,01
0,18	8,0	0,20	1,31
0,11	-3,0	0,69	0,03
0,02	0,7	0,03	0,10
0,07	3,7	0,01	0,33
0,13	6,8	0,00	1,01
0,14	7,2	0,01	1,16
0,15	6,4	0,11	1,08
0,16	6,1	0,19	1,14
0,16	6,0	0,20	1,12
0,16	6,1	0,19	1,17
0,18	5,5	0,36	1,16
0,19	4,5	0,46	1,08
0,19	4,4	0,45	1,02
0,18	4,5	0,47	1,10
0,19	4,5	0,48	1,11
0,19	4,5	0,49	1,11
0,19	4,4	0,34	1,00
0,19	4,4	0,43	1,09
0,19	4,4	0,51	1,13
0,19	4,4	0,49	1,07
0,20	3,2	0,73	1,04
0,20	2,3	0,81	0,98
0,21	2,6	0,76	0,98

APÊNDICE B

Informações associadas à comparação dos algoritmos:

MLPREGRESSOR X STANDARDSCALER X INDEPENDENTE:

Tabela B-1 - Conjunto de hiperparâmetros da rede otimizada do estimador A.

Topologia (Hidden_Layer_Sizes)	Ativação	Solver	Aprendizado	Score treinamento	Score validação
(2,2)	tanh	lbfgs	constant	0,9935	0,9863

Tabela B-2 – “Scores” do estimador A com os dados experimentais.

Rede	Score Treinamento	Score Validação
Transfer Learn	0,8656	0,2975
Sem Transfer Learn	0,7678	0,4414

Para o estimador B:

Tabela B-3 - Conjunto de hiperparâmetros da otimização para o estimador B.

Topologia (Hidden_Layer_Sizes)	Ativação	Solver	Aprendizado	Score treinamento	Score validação
(2,2)	tanh	lbfgs	adptive	0,9488	0,9487

Tabela B-4 – “Scores” do estimador B com os dados experimentais.

Rede	Score Treinamento	Score Validação
Transfer Learn	0,9497	0,9104
Sem Transfer Learn	0,9529	0,8933

MLPREGRESSOR X MINMAX X IDEPENDENTE:

Alterando-se o método de normalização e utilizando a separação das saídas, foram obtidos os seguintes resultados:

Para o estimador A:

Tabela B-5 - Conjunto ótimo de parâmetros do estimador A.

Topologia (Hidden_Layer_Sizes)	Ativação	Solver	Aprendizado	Score treinamento	Score validação
(2,2)	tanh	lbfgs	invscaling	0,9919	0,9901

Tabela B-6 – Score do estimador A com os dados experimentais

Rede	Score Treinamento	Score Validação
Transfer Learn	0,7407	0,2734
Sem Transfer Learn	0,7462	0,28755

Para o estimador B:

Tabela B-7 - Conjunto ótimo de parâmetros para o estimador B.

Topologia (Hidden_Layer_Sizes)	Ativação	Solver	Aprendizado	Score treinamento	Score validação
(2,2)	tanh	lbfgs	constant	0,9358	0,9141

Tabela B-8 – “Scores” do estimador B para os dados experimentais.

Rede	Score Treinamento	Score Validação
Transfer Learn	0,9229	0,9173
Sem Transfer Learn	0,9550	0,8992

MLPREGRESSOR X MINMAX X PADRÃO:

Neste caso, só existirá uma rede. Apenas será alterado o método de normalização, passando a utilizar-se o MinMax.

Tabela B-9 - Conjunto ótimo de parâmetros do estimador.

Topologia (Hidden_Layer_Sizes)	Ativação	Solver	Aprendizado	Score treinamento	Score validação
(2,2)	tanh	lbfgs	constant	0,9663	0,9546

Tabela B-10 – “Scores” com os dados experimentais.

Rede	Score Treinamento	Score Validação
Transfer Learn	0,8628	0,5027
Sem Transfer Learn	0,8171	0,5353

RANDOM FOREST X STANDARDSCALER X PADRÃO:

No caso do estimador Random Forest, não foi utilizado o conceito de Transfer Learn. Assim, o estimador foi treinado após a otimização e apenas os seus hiperparâmetros foram utilizados no estimador treinado usando os dados experimentais.

Tabela B-11 - Conjunto de hiperparâmetros obtidos da otimização.

Criterion	Max Features	Oob Score	Score treinamento	Score validação
MSE	None	False	0,9931	0,9562

Tabela B-12 – “Scores” dados experimentais.

Score Treinamento	Score Validação
0,8933	0,6611

RANDOM FOREST X STANDARD SCALER X INDEPENDENTE:

O procedimento neste caso é o mesmo utilizado no MLPRegressor, separando os estimadores em 2.

Para o estimador A:

Tabela B-13 - Conjunto de hiperparâmetros obtidos da otimização estimador A.

Criterion	Max Features	Oob Score	Score treinamento	Score validação
MSE	None	True	0,9982	0,9831

Tabela B-14 – “Scores” dados experimentais estimador A.

Score Treinamento	Score Validação
0,8561	0,4064

Para o estimador B:

Tabela B-15 - Conjunto de hiperparâmetros obtidos da otimização estimador B.

Criterion	Max Features	Oob Score	Score treinamento	Score validação
MAE	None	True	0,9874	0,6512

Tabela B-16 – “Scores” dados experimentais estimador B.

Score Treinamento	Score Validação
0,9490	0,8074

RANDOM FOREST X MINMAX X INDEPENDENTE:

Para o estimador A:

Tabela B-17 - Conjunto de hiperparâmetros obtidos da otimização estimador B.

Criterion	Max Features	Oob Score	Score treinamento	Score validação
MSE	None	False	0,9982	0,9946

Tabela B-18 – “Scores” dados experimentais estimador A.

Score Treinamento	Score Validação
0,8566	0,5240

Para o estimador B:

Tabela B-19 - Conjunto de hiperparâmetros obtidos da otimização estimador B.

Criterion	Max Features	Oob Score	Score treinamento	Score validação
MAE	None	True	0,9859	0,9498

Tabela B-20 – “Scores” dados experimentais estimador B.

Score Treinamento	Score Validação
0,9498	0,8012

RANDOM FOREST X MINMAX X PADRÃO:

Tabela B-21 - Conjunto de hiperparâmetros obtidos da otimização.

Criterion	Max Features	Oob Score	Score treinamento	Score validação
mae	None	False	0,9927	0,9675

Tabela B-22 – “Scores” dados experimentais.

Score Treinamento	Score Validação
0.8869	0.5982