UNIVERSIDADE ESTADUAL PAULISTA "JÚLIO DE MESQUITA FILHO"

INSTITUTO DE QUÍMICA – CAMPUS ARARAQUARA

DEPARTAMENTO DE ENGENHARIA, FÍSICA E MATEMÁTICA

CLASSIFICAÇÃO DE VIDROS COM BASE NO TEOR DE ÓXIDOS UTILIZANDO REDES NEURAIS ARTIFICIAIS

LARA BORGES MACIEL RIBEIRO

ARARAQUARA - SP

2024

LARA BORGES MACIEL RIBEIRO

CLASSIFICAÇÃO DE VIDROS COM BASE NO TEOR DE ÓXIDOS UTILIZANDO REDES NEURAIS ARTIFICIAIS

Trabalho de Conclusão de Curso apresentado ao Instituto de Química, da Universidade Estadual Paulista “Júlio de Mesquita Filho” como parte das exigências para a obtenção do título de Bacharel em Engenharia Química.

Orientadora: Profa. Dra. Érica Regina Filletti Nascimento

ARARAQUARA - SP

2024

R484c

Ribeiro, Lara Borges Maciel

Classificação de vidros com base no teor de óxidos utilizando redes neurais artificiais / Lara Borges Maciel Ribeiro. -- Araraquara, 2024

44 p. : il.

Trabalho de conclusão de curso (Bacharelado - Engenharia Química) - Universidade Estadual Paulista (UNESP), Instituto de Química, Araraquara

Orientadora: Érica Regina Filletti Nascimento

1. Redes neurais (Computação). 2. Inteligência artificial. 3. Capacidade de aprendizagem. 4. Vidro. 5. MATLAB (Programa de computador). I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Dados fornecidos pelo autor(a).

LARA BORGES MACIEL RIBEIRO

CLASSIFICAÇÃO DE VIDROS COM BASE NO TEOR DE ÓXIDOS UTILIZANDO REDES NEURAIS ARTIFICIAIS

Trabalho de Conclusão de Curso apresentado à Universidade Estadual Paulista (UNESP) Instituto de Química, Araraquara, para obtenção do título de Bacharel em Engenharia Química.

Data da defesa: 10/12/2024

Banca Examinadora:

Documento assinado digitalmente
gov.br
ERICA REGINA FILLETTI NASCIMENTO
Data: 06/01/2025 17:04:05-0300
Verifique em https://validar.iti.gov.br

_____

Profa. Dra. Érica Regina Filletti Nascimento

Instituto de Química – UNESP, Araraquara

Documento assinado digitalmente
gov.br
FABIO ROBERTO CHAVARETTE
Data: 06/01/2025 17:09:01-0300
Verifique em https://validar.iti.gov.br

_____

Prof. Dr. Fábio Roberto Chavarette

Instituto de Química – UNESP, Araraquara

Documento assinado digitalmente
gov.br
MARCELO NALIN
Data: 07/01/2025 10:50:31-0300
Verifique em https://validar.iti.gov.br

_____

Prof. Dr. Marcelo Nalin

Instituto de Química – UNESP, Araraquara

RESUMO

Este trabalho visa a classificação de amostras de vidro com base em suas composições de óxidos, utilizando Redes Neurais Artificiais (RNAs) como metodologia de análise. A relevância deste estudo está na aplicação de técnicas de inteligência artificial no contexto forense, permitindo a identificação rápida e precisa de fragmentos de vidro encontrados em cenas de crime. Para isso, foi utilizada a base de dados Glass Identification, contendo 214 amostras de vidro classificadas em seis categorias, considerando a composição de óxidos de cada uma das amostras. O modelo de RNA foi desenvolvido e treinado no software MATLAB, onde foram testados três algoritmos de aprendizado — Retropropagação de Erros, Gradiente Conjugado Escalonado e Levenberg-Marquardt — com o último apresentando o melhor desempenho entre eles. O resultado obtido foi uma taxa de acerto de 90,19%, demonstrando a eficácia das RNAs na classificação de diferentes tipos de vidro. Concluiu-se que as redes neurais são ferramentas promissoras para a análise forense de vidros, pois aprimoram a precisão e agilidade na identificação de amostras, contribuindo para a investigação criminal. Contudo, limitações na distribuição de amostras na base de dados e o potencial de *overfitting* destacam a necessidade de estudos adicionais com bases de dados mais balanceadas.

Palavras-chave: Redes Neurais Artificiais. Vidros. Classificação. Óxidos. Análise forense. Inteligência Artificial. Levenberg-Marquardt. MATLAB.

ABSTRACT

This study aims to classify glass samples based on their oxide compositions, using Artificial Neural Networks (ANNs) as an analytical methodology. The relevance of this research lies in the application of artificial intelligence techniques within forensic science, enabling the rapid and accurate identification of glass fragments found at crime scenes. For this purpose, the Glass Identification dataset was used, containing 214 glass samples classified into six categories, considering oxide composition of each sample. The ANN model was developed and trained in MATLAB, where three learning algorithms were tested — Error Backpropagation, Scaled Conjugate Gradient, and Levenberg-Marquardt — with the latter yielding the best performance. The model achieved an accuracy rate of 90,19%, demonstrating the effectiveness of ANNs in classifying different types of glass. It was concluded that neural networks are promising tools for forensic glass analysis, enhancing both precision and efficiency in sample identification, thus supporting criminal investigations. However, limitations in the data sample distribution and the potential for overfitting highlight the need for further studies using more balanced datasets.

Keywords: Artificial Neural Networks. Glass. Classification. Oxides. Forensic analysis. Artificial Intelligence. Levenberg-Marquardt. MATLAB.

LISTA DE FIGURAS

Figura 1: Exemplo do funcionamento de uma Rede Neural Feedforward. 13
Figura 2: Código de programação para obtenção da RNA com melhor desempenho.27
Figura 3: Gráfico de Desempenho da Rede Neural Artificial 28
Figura 4: Curvas ROC para o conjunto treinamento. 30
Figura 5: Curvas ROC para o conjunto validação. 31
Figura 6: Curvas ROC para o conjunto teste.....32
Figura 7: Curvas ROC para o total dos dados.....33
Figura 8: Matriz de confusão para o conjunto treinamento. 34
Figura 9: Matriz de confusão para o conjunto validação. 35
Figura 10: Matriz de confusão para o conjunto teste.....36
Figura 11: Matriz de confusão para o total dos dados. 37

LISTA DE TABELAS

Tabela 1: Classificação do tipo de vidro de acordo com o número da classe. 19
Tabela 2: Quantidade de amostra em cada uma das classes. 20
Tabela 3: Valores iniciais e alvo de diferentes critérios de parada. 20
Tabela 4: Comparação de dados e resultado do conjunto treinamento entre o obtido e esperado pela RNA. 39

SUMÁRIO

1 Introdução 9
2 Referencial Teórico 12
2.1 Histórico das Redes Neurais Artificiais 15
2.2 Trabalhos Recentes Relacionados 16
3 Metodologia 19
3.1 Algoritmos Testados 21
3.1.1 Retropropagação de Erros 21
3.1.2 Gradiente Conjugado Escalonado 22
3.1.3 Levenberg-Marquardt 23
4 Resultados e Discussão 27
4.1 Desempenho da Rede Neural Artificial 28
4.2 Curva ROC 30
4.3 Matriz de Confusão 34
5 Considerações Finais 40
6 Referências Bibliográficas 42

1 INTRODUÇÃO

Conforme definição da ASTM (Sociedade Americana para Testes e Materiais), o vidro é um material derivado da fusão de compostos que, ao serem resfriados, alcançam uma rigidez sem cristalização, assumindo uma estrutura amorfa. Essa característica faz com que o vidro possa ser descrito como um “líquido super-resfriado”, conferindo-lhe propriedades como transparência, dureza e a capacidade de ser moldado em variadas formas. No mercado, os vidros possuem grande versatilidade, sendo amplamente utilizados em setores como construção civil, tecnologia, setor automobilístico e embalagens, demonstrando a importância deste material em diferentes indústrias.[1] [2]

A composição química dos vidros pode variar significativamente, dependendo de sua aplicação específica. Em geral, são compostos por ingredientes como sílica (SiO2), que fornece a estrutura básica do material; óxido de sódio (Na2O) e óxido de cálcio (CaO), que estabilizam a estrutura; e outros óxidos que afetam propriedades específicas, como resistência térmica, cor e durabilidade (como óxido de magnésio, ferro, alumínio e potássio). A variação no teor desses óxidos é fundamental para personalizar as características do vidro, permitindo sua adequação a diferentes demandas de mercado.[3]

A investigação criminal é uma ciência complexa e meticulosa, onde cada detalhe pode ser crucial para a resolução de um caso. Em muitos cenários, a análise forense de evidências físicas desempenha um papel fundamental, e a identificação precisa de materiais, como vidro, pode ser determinante para a conclusão bem-sucedida de uma investigação. Fragmentos de vidro encontrados em uma cena de crime podem fornecer insights valiosos, pois cada tipo de vidro possui características únicas, desde sua composição química até a maneira como quebra e se dispersa. A análise desses fragmentos pode ajudar a reconstruir eventos, determinar pontos de entrada ou saída, e até identificar o tipo específico de objeto usado para quebrar o vidro, estabelecendo conexões entre diferentes locais e fornecendo pistas cruciais sobre a dinâmica do incidente. Para realizar essa identificação, os peritos utilizam técnicas como análise microscópica e espectroscopia, além de um conhecimento profundo sobre as propriedades dos diferentes tipos de vidro. A identificação correta

de diferentes tipos de vidro desempenha, assim, um papel essencial na investigação criminal, fornecendo informações sólidas para a resolução de casos e evidências confiáveis para serem apresentadas em tribunal.[4] [5]

Nesse contexto, as Redes Neurais Artificiais (RNAs) surgem como uma abordagem inovadora e promissora para a análise de composições químicas, devido à sua capacidade de identificar padrões e correlacionar dados complexos de maneira precisa e ágil. Com o treinamento adequado, as RNAs podem atuar como uma ferramenta de apoio na classificação de vidros, auxiliando peritos a identificar e caracterizar amostras com maior rapidez e precisão. Assim, este trabalho busca explorar e avaliar a eficácia das RNAs na classificação de vidros, destacando seu potencial como ferramenta para a área forense. Esse estudo pode, portanto, contribuir para tornar os processos de identificação e análise de amostras de vidro mais eficientes e assertivos, fortalecendo as metodologias aplicadas nas investigações criminais.

Este estudo busca responder ao seguinte problema de pesquisa: como uma rede neural artificial, treinada com dados de composições químicas de vidros disponíveis na literatura, pode ser utilizada para classificar amostras de vidro em diferentes classes, possibilitando a identificação e compreensão da origem de fragmentos de vidro em contextos forenses?

Para atingir esse propósito, o objetivo geral é classificar amostras de vidro com base em suas composições de óxidos, utilizando redes neurais artificiais. Os objetivos específicos incluem desenvolver e treinar uma rede neural artificial com dados de composições de vidros obtidos na literatura e avaliar a precisão e eficiência da rede neural na classificação de diferentes classes de vidro.

Para atingir os objetivos propostos, este estudo utiliza uma base de dados de composições químicas de vidros do repositório UCI Machine Learning intitulada *Glass Identification* [15], com 214 amostras, cada uma caracterizada pelo teor de óxidos. Essas amostras foram classificadas em seis classes, e os dados foram divididos em subconjuntos de treinamento, validação e teste. Para avaliar a performance da rede neural artificial, foram testados três diferentes algoritmos de aprendizado. O modelo com a maior taxa de acerto foi selecionado para análise, validando o potencial das redes neurais na classificação precisa de vidros.

A estrutura deste trabalho é dividida em cinco capítulos. No primeiro capítulo, a introdução, foram apresentados o tema, os objetivos, a justificativa e a metodologia empregada. O segundo capítulo é dedicado à revisão bibliográfica, abordando conceitos fundamentais, o histórico e o desenvolvimento das redes neurais, seus avanços e aplicações específicas na classificação de materiais. O terceiro capítulo descreve a metodologia em detalhes, cobrindo o desenvolvimento e treinamento da rede neural. No quarto capítulo, resultados e discussão, são apresentados o código em MATLAB que gerou a RNA com melhor desempenho e uma análise aprofundada dos resultados obtidos. Por fim, no quinto capítulo, considerações finais, são sintetizadas as principais descobertas e conclusões do trabalho, abordando também as limitações do estudo e possíveis direções para pesquisas futuras.

2 REFERENCIAL TEÓRICO

O aprendizado profundo, ou *Deep Learning*, é uma subárea da Inteligência Artificial (IA) que se concentra em redes neurais com múltiplas camadas, conhecidas como redes neurais profundas. O termo "profundo" refere-se ao uso de Redes Neurais Artificiais (RNAs) com várias camadas para modelar padrões complexos em grandes conjuntos de dados. Estas redes neurais profundas possuem a capacidade de aprender representações de características cada vez mais abstratas e hierárquicas dos dados à medida que as informações são processadas através das camadas.[6]

Um fator essencial para o sucesso do *Deep Learning* nos últimos anos é a disponibilidade de grandes volumes de dados e o avanço de hardware, especialmente das GPUs (Unidades de Processamento Gráfico), que permitem treinar modelos complexos de maneira mais eficaz.

O aprendizado profundo geralmente requer grandes conjuntos de dados para treinamento eficaz, pois a quantidade de dados disponíveis impacta diretamente na habilidade do modelo de generalizar padrões para dados não vistos. Algoritmos de otimização avançados, como o Gradiente Descendente Estocástico, são fundamentais no treinamento de redes neurais profundas, ajustando os pesos da rede para minimizar a função de custo, indicando quão bem as previsões da rede correspondem aos rótulos reais. [6]

Esse cenário proporciona condições ideais para aplicar *Deep Learning* em tarefas que antes eram difíceis para métodos tradicionais de aprendizado de máquina, como a classificação de materiais pela composição química ou outras características, o que é altamente relevante para a análise de vidros no contexto deste trabalho. [6]

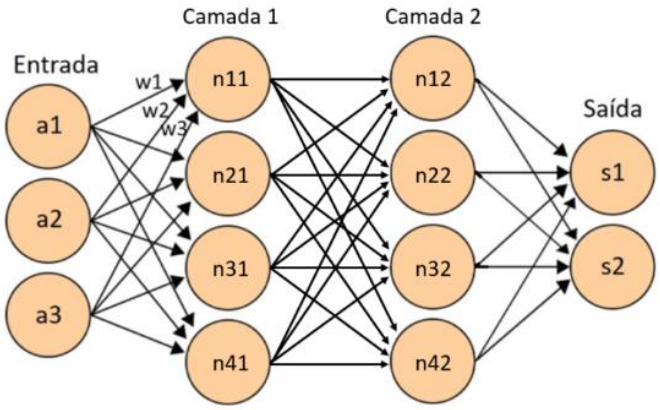
As Redes Neurais Artificiais (RNAs), que formam a base do *Deep Learning*, são sistemas computacionais inspirados na estrutura e no funcionamento do cérebro humano. Esses modelos matemáticos tentam imitar o comportamento dos neurônios biológicos por meio de unidades interconectadas, chamadas neurônios artificiais ou *perceptrons*. Cada conexão entre esses neurônios possui um peso associado, que é ajustado durante o treinamento da rede para que ela possa discernir padrões nos dados de entrada.[7]

Essas redes consistem em várias camadas:

- Camada de Entrada: Recebe os dados de entrada e os transmite para a camada oculta. Cada nó nesta camada representa uma característica ou atributo dos dados iniciais.
- Camadas Ocultas: Podem ser uma ou várias, e realizam transformações nos dados recebidos da camada de entrada. Cada nó em uma camada oculta processa uma combinação linear dos dados de entrada, ponderados pelos pesos, e aplica uma função de ativação não-linear para gerar uma saída.
- Camada de Saída: Produz a resposta final da rede neural. O número de nós nesta camada depende do tipo de problema em questão. [7]

As Redes Neurais *Feedforward* (RNF), também conhecidas como redes neurais de alimentação direta, representam um tipo comum de Redes Neurais Artificiais, onde a informação flui em apenas uma direção: para frente, através das camadas da rede. Em outras palavras, os dados de entrada são transmitidos adiante pela rede, camada por camada, até alcançar a camada de saída, sem qualquer retroalimentação dos neurônios de saída para os neurônios de entrada, conforme representado na Figura 1. [8]

Figura 1: Exemplo do funcionamento de uma Rede Neural Feedforward.



Fonte: Didática Tech, 2022.

O aprendizado das Redes Neurais Artificiais (RNAs) ocorre ajustando os pesos das conexões entre neurônios, buscando minimizar o erro entre as previsões da rede e os valores reais (ou alvos). No início, os pesos das conexões são geralmente definidos de maneira aleatória ou com valores iniciais específicos. Durante o treinamento, ocorre a propagação para frente (ou *forward propagation*): os dados de

entrada são alimentados na rede, e cada neurônio na camada de entrada representa uma característica desses dados. As informações são transmitidas pelas camadas ocultas, onde são aplicadas transformações lineares e não lineares baseadas nos pesos das conexões, resultando na previsão da rede para os dados de entrada.[7]

A previsão gerada pela rede é então comparada com o valor real correspondente, e a diferença entre eles, o erro, é calculada usando uma função de custo. Esse erro é essencial para o processo de aprendizado. Na etapa de retropropagação (ou *backpropagation*), o erro é distribuído de trás para frente na rede, da camada de saída para as camadas anteriores, permitindo identificar quanto cada peso contribui para o erro total. O objetivo é ajustar esses pesos de forma a reduzir o erro nas próximas previsões.[7]

Para realizar esse ajuste, um algoritmo de otimização, como o Levenberg-Marquardt ou o Gradiente Descendente Estocástico, é empregado para modificar os pesos gradualmente, em passos proporcionais ao erro calculado, até que a função de custo seja minimizada. Esse processo de propagação e ajuste é repetido por várias iterações ou épocas, aplicando o mesmo procedimento a múltiplos exemplos de treinamento. O ciclo é mantido até que o modelo atinja um nível aceitável de precisão ou que o erro pare de diminuir significativamente.[7]

Após o treinamento, espera-se que a RNA seja capaz de generalizar, ou seja, fazer previsões precisas para novos dados, não vistos durante o treinamento. Para assegurar essa capacidade, as amostras do problema em estudo são divididas em três conjuntos:

1. **Conjunto de treinamento** – usado para ajustar os pesos da rede com base no erro calculado.
2. **Conjunto de validação** – utilizado para avaliar a capacidade de generalização da rede e determinar o momento adequado para interromper o treinamento, prevenindo o *overfitting*.
3. **Conjunto de teste** – empregado para uma avaliação independente do desempenho da rede, permitindo verificar sua precisão fora do processo de treinamento. [9]

Esse processo iterativo de ajustar pesos conforme o erro calculado permite que a rede aprenda padrões complexos e faça previsões cada vez mais precisas. Essa capacidade de aprendizado possibilita que as RNAs abordem problemas de classificação, reconhecimento de padrões, visão computacional, entre outros, sendo particularmente valiosas para resolver problemas complexos nos quais os padrões nos dados são difíceis de capturar por métodos tradicionais de programação.

2.1 Histórico das Redes Neurais Artificiais

A criação das RNAs está intimamente ligada a estudos sobre o funcionamento do cérebro humano. O cérebro, com cerca de 86 bilhões de neurônios conectados por sinapses, processa informações de maneira hierárquica, identificando primeiro padrões simples e, em seguida, combinações mais complexas. Inspiradas nessa hierarquia biológica, as RNAs organizam neurônios em camadas, onde cada camada processa uma representação dos dados, permitindo que a rede extraia padrões a partir de dados brutos. Assim como no cérebro, onde a plasticidade sináptica permite a adaptação das conexões sinápticas, as RNAs ajustam os pesos das conexões entre os neurônios durante o processo de aprendizado. [10] [11]

Essa semelhança entre o funcionamento do cérebro e das RNAs reforça a aplicabilidade dessas redes em tarefas de reconhecimento de padrões e aprendizado hierárquico, imitando, ainda que de forma simplificada, o processamento adaptativo e flexível das redes biológicas. Contudo, vale destacar que, apesar das semelhanças estruturais, as RNAs ainda representam uma simplificação do cérebro humano e têm limitações significativas quando comparadas à complexidade do sistema nervoso biológico.[10] [11]

O desenvolvimento das RNAs tem uma longa trajetória, iniciada com o modelo de *Perceptron* de McCulloch-Pitts (MCP) na década de 1940, que foi o primeiro modelo computacional inspirado em neurônios biológicos. Esse modelo, criado por Warren McCulloch e Walter Pitts, consistia em entradas ponderadas que eram somadas e passavam por um limiar de ativação para produzir uma saída. O *Perceptron*, no entanto, limitava-se a resolver problemas linearmente separáveis, o que restringia sua aplicabilidade. Em 1957, Frank Rosenblatt expandiu o modelo com o *Perceptron* adaptativo, que introduziu o conceito de aprendizado supervisionado,

ajustando os pesos com base nos erros durante a classificação dos dados de treinamento. Modelos mais avançados, como o ADALINE (*Adaptive Linear Neuron*) e o MADALINE (*Multiple ADALINE*), desenvolvidos por Bernard Widrow e Marcian Hoff, trouxeram avanços como o aprendizado adaptativo e a capacidade de lidar com múltiplas saídas. Esses modelos representaram etapas fundamentais para o desenvolvimento das redes neurais multicamadas, que hoje são amplamente utilizadas para resolver problemas complexos de aprendizado de máquina. [11] [12]

2.2 Trabalhos Recentes Relacionados

No artigo de BERTO (2020) as autoras investigaram o uso de redes neurais artificiais (RNAs) para classificar amostras de cabelo humano quanto ao pigmento (cabelo natural ou descolorido) e ao gênero do doador. As análises foram realizadas com técnicas não invasivas de fluorescência de raios X de dispersão de comprimento de onda (WDXRF) e espectroscopia de emissão induzida por laser (LIBS). O objetivo principal foi verificar se as RNAs poderiam proporcionar uma alternativa eficiente às técnicas quimiométricas tradicionais, as quais apresentaram limitações em estudos anteriores, especialmente para classificar características complexas, como gênero e pigmento.[13]

Os resultados mostraram que a RNA desenvolvida para dados WDXRF foi capaz de classificar o pigmento das amostras com 100% de acurácia e o gênero com 89,5% de acurácia. Para a LIBS, a classificação do pigmento também obteve 100% de acurácia, enquanto a classificação do gênero alcançou 78,9%. Estes resultados indicam que a WDXRF oferece uma performance ligeiramente superior, especialmente para a determinação do gênero, quando comparada à LIBS. [13]

Na conclusão, os autores destacaram o potencial das RNAs, particularmente com os dados obtidos pela WDXRF, para a classificação multivariada de amostras biológicas como o cabelo. A aplicação da RNA permitiu resultados robustos e precisos, sugerindo sua utilidade em estudos de classificação onde os métodos convencionais falham. [13]

Este estudo se alinha ao tema do presente Trabalho de Conclusão de Curso, que envolve a classificação de materiais com base em suas características físico-químicas usando redes neurais artificiais. Assim como na classificação de amostras de cabelo, onde as RNAs conseguem lidar com as complexidades de dados espectrais e distinguir entre características distintas, o uso de RNAs para classificar vidros conforme o teor de óxidos representa uma aplicação similar. Ambos os estudos compartilham a abordagem de utilizar redes neurais para superar as limitações das técnicas tradicionais, promovendo a precisão e confiabilidade na análise de dados complexos. [13]

O artigo de RODRIGUES (2021) explora o uso de técnicas de análise multivariada para diferenciar e classificar fragmentos de para-brisas automotivos em contextos forenses. O objetivo do estudo foi avaliar a possibilidade de distinguir entre as faces interna e externa dos para-brisas de diferentes marcas de veículos usando análise multivariada, com dados de fluorescência de raios X de energia dispersiva (EDXRF) sobre as concentrações de óxidos em 56 amostras de vidro.[5]

A análise foi conduzida com duas abordagens: uma não supervisionada, por meio da Análise de Componentes Principais (PCA), e outra supervisionada, usando o método de Classificação de Analogias de Modelagem Independente Suave (SIMCA). A PCA identificou dois grupos principais de vidros, associados às faces interna e externa dos para-brisas, com variáveis-chave como óxidos de ferro e potássio. O modelo SIMCA foi então desenvolvido para classificar corretamente futuras amostras, e a validação desse modelo, por meio da Análise de Características Operacionais do Receptor, demonstrou alta acurácia com sensibilidade de 79,3% e especificidade de 81,5%.[5]

A conclusão do estudo destaca que a metodologia empregada pode auxiliar na diferenciação de fragmentos de vidros em investigações de crimes de trânsito, permitindo inferências sobre a origem dos fragmentos e contribuindo para a reconstituição da dinâmica do evento.[5]

Assim como no estudo, onde óxidos são utilizados para discriminar entre as faces de vidros automotivos, o presente trabalho se propõe a utilizar redes neurais artificiais para classificar vidros conforme o teor de óxidos, evidenciando a importância

de métodos analíticos e algoritmos de classificação para identificar características de materiais complexos.[5]

O trabalho de ERLER (2023) teve como objetivo principal criar algoritmos de redes neurais artificiais (RNAs) e florestas aleatórias para a classificação de diferentes grãos, utilizando imagens digitais e dados complementares de massa, volume e densidade. Para o desenvolvimento do modelo, foram testados três algoritmos diferentes para as RNAs (retropropagação, Levenberg-Marquardt e gradiente conjugado escalonado) com o intuito de avaliar qual fornecia o melhor desempenho para a classificação correta dos grãos. A classificação foi baseada em descritores de cor extraídos das imagens, e um modelo convolucional também foi desenvolvido para verificar a eficácia de uma abordagem de rede neural convolucional (CNN) com transferência de aprendizagem. [14]

Os resultados mostraram que o modelo de RNA treinado com o algoritmo de gradiente conjugado escalonado conseguiu 100% de precisão na classificação das quatro classes de grãos. O uso de florestas aleatórias também mostrou bom desempenho, confirmando sua utilidade para esse tipo de classificação. Além disso, a técnica de transferência de aprendizagem aplicada no modelo CNN permitiu que a classificação fosse realizada com alta precisão e em um tempo de processamento mais rápido. [14]

Na conclusão, o estudo demonstrou que tanto as RNAs quanto as florestas aleatórias são ferramentas eficazes para a classificação de grãos, e que a CNN com transferência de aprendizagem pode ser usada para agilizar o processo. Assim como o atual projeto visa a classificação de vidros com base no teor de óxidos usando RNAs, este estudo fornece uma base metodológica sobre o uso de redes neurais para distinguir entre diferentes materiais, demonstrando o potencial das RNAs em contextos de classificação complexa. [14]

3 METODOLOGIA

A pesquisa foi planejada com o objetivo de caracterizar amostras de vidros com base no teor de óxidos presentes, utilizando Redes Neurais Artificiais (RNA). A estratégia incluiu a utilização de uma base de dados disponível, a divisão dos dados em subconjuntos para treinamento, validação e testes, e a aplicação de diferentes algoritmos de aprendizado para verificar a performance da RNA.

A pesquisa foi conduzida utilizando dados do repositório de aprendizado de máquina UCI, especificamente a base de dados fornecida por German, B. (1987) intitulada *Glass Identification*.[15] O estudo focou-se na análise de dados referentes à composição percentual de óxidos em cada amostra.

A base de dados utilizada é composta por 214 diferentes amostras de vidro, cada uma contendo informações detalhadas sobre a composição em massa de oito diferentes óxidos (sódio, magnésio, alumina, sílica, potássio, cálcio, bário e ferro). As amostras foram previamente classificadas em seis classes distintas, representadas nas Tabelas 1 e 2:

Tabela 1: Classificação do tipo de vidro de acordo com o número da classe.

Número da Classe	Tipo de Vidro
1	Janelas de edifícios processadas por flutuação
2	Janelas de edifícios processadas sem flutuação
3	Janelas de veículos processadas por flutuação
4	Recipientes
5	Louças
6	Faróis

Fonte: Elaborado pela autora, 2024.

Tabela 2: Quantidade de amostra em cada uma das classes.

Número da Classe	Quantidade de Amostras
1	70
2	76
3	17
4	13
5	9
6	29

Fonte: Elaborado pela autora, 2024.

Os dados foram organizados em duas matrizes: uma matriz de entrada com 214 colunas, representando as amostras, e 8 linhas, representando a composição dos óxidos; e uma matriz de saída com 214 colunas e 6 linhas, nas quais o valor "1" indica que a amostra pertence a uma classe específica, e "0" indica que ela não pertence. Essas matrizes foram carregadas no software MATLAB, e a ferramenta *Neural Network Getting Started* foi utilizada para programar, treinar e visualizar a RNA.

O processo de treinamento da RNA utilizou 70% dos dados disponíveis, enquanto 15% foram destinados à validação e 15% aos testes. Foram testados três algoritmos de aprendizado distintos: Retropropagação de Erros, Levenberg-Marquardt e Gradiente Conjugado Escalonado, que serão descritos na próxima sessão.

Para garantir que o treinamento fosse eficiente, foram definidos critérios de parada que assegurassem que a RNA não fosse sobrecarregada ou treinada excessivamente, cujos valores alvo estão descritos na Tabela 3:

Tabela 3: Valores iniciais e alvo de diferentes critérios de parada.

Critério de Parada	Valor Inicial	Valor Alvo
Época	0	1000
Desempenho	0,261	0
Gradiente	0,0936	1,00E-07
Fator de ajuste do método	0,001	1,00E+10
Verificações de validação	0	6

Fonte: Elaborado pela autora, 2024.

O procedimento para determinar a RNA com maior taxa de acerto foi dividido em duas etapas principais. Inicialmente, o código em MATLAB foi automatizado para

variar o número de neurônios na camada oculta de 1 a 100, interrompendo o treinamento assim que a acurácia ultrapassasse 88%. Nessa primeira fase, identificou-se que a melhor arquitetura correspondia a uma rede com 47 neurônios na camada oculta.

Na etapa seguinte, um novo código foi desenvolvido para realizar múltiplos treinamentos em looping, mantendo os 47 neurônios fixos, até que a acurácia superasse o valor estabelecido de 88%. Esse procedimento foi repetido diversas vezes, gerando diferentes configurações de RNA. Por fim, a configuração com maior acurácia obtida foi selecionada.

Os resultados finais e os parâmetros da RNA escolhida foram salvos em um arquivo .mat, garantindo a reprodutibilidade dos resultados e possibilitando a utilização da rede neural na classificação de novos conjuntos de dados.

3.1 Algoritmos Testados

3.1.1 Retropropagação de Erros

A retropropagação de erros (também conhecida como *backpropagation*) é um algoritmo utilizado para treinar redes neurais, especialmente redes neurais profundas (*Deep Neural Networks*). Ele é utilizado para ajustar os pesos das conexões entre os neurônios, permitindo que a rede aprenda a mapear entradas para saídas desejadas de forma mais precisa e funciona da seguinte forma:

1) Inicialização: Primeiro, os pesos da rede neural são inicializados com valores aleatórios ou próximos de zero.
2) Propagação para Frente: Os dados de entrada são passados através da rede neural, camada por camada, para gerar uma previsão na camada de saída. Cada neurônio realiza uma combinação linear das entradas ponderadas pelos pesos e aplica uma função de ativação para produzir um resultado.
3) Cálculo do Erro: A diferença entre a previsão da rede e o valor real é calculada. Isso é chamado de erro ou função de custo. O objetivo é minimizar esse erro.

4) Retropropagação do Erro: O erro é então retropropagado pela rede, do final para o início. Em outras palavras, o algoritmo calcula quanto cada peso contribuiu para o erro.

5) Ajuste dos Pesos: Utilizando a informação sobre como cada peso contribuiu para o erro, o algoritmo de otimização (Gradiente Descendente) ajusta os pesos para reduzir o erro na próxima iteração. A ideia é ajustar os pesos na direção em que o erro diminui.

6) Iteração: Os passos 2 a 5 são repetidos por várias iterações (ou épocas) até que o erro da rede seja suficientemente baixo ou não melhore significativamente.

7) Convergência: Com o tempo, os pesos da rede neural convergem para valores que minimizam o erro, ou seja, a rede se torna melhor em fazer previsões precisas com base nos dados de entrada.

8) Validação e Teste: Após o treinamento, a rede é testada com dados que não foram usados no treinamento para verificar se ela generaliza bem para novos dados, ou seja, se ela pode fazer previsões precisas em situações não vistas durante o treinamento. [14] [16]

Este processo de retropropagação de erros foi utilizado para o treinamento da RNA através da ferramenta *Neural Network Getting Started* a partir do código *traingd*.

3.1.2 Gradiente Conjugado Escalonado

O Gradiente Conjugado Escalonado (SCG) é um algoritmo de otimização utilizado para treinar redes neurais, sendo particularmente eficiente para modelos com grandes conjuntos de dados, pois é mais rápido e menos computacionalmente exigente. Ele visa minimizar a função de erro ao ajustar iterativamente os pesos da rede neural em direções estrategicamente selecionadas e funciona da seguinte forma:

1) Inicialização: O algoritmo começa com uma direção inicial de busca definida pelo gradiente da função de erro, que indica o caminho de descida mais rápido.

2) Ajuste dos Pesos: Em cada iteração, o SCG ajusta os pesos da rede neural ao longo da direção de busca determinada. Este ajuste ocorre para minimizar a função de erro.
3) Determinação da Próxima Direção de Busca: Após o ajuste dos pesos, o algoritmo calcula uma nova direção de busca usando informações da direção de busca anterior. Essa nova direção é escolhida para manter a "conjugação" com a anterior, o que ajuda na eficiência do algoritmo.
4) Eliminação da Busca de Linha: Diferente de outros algoritmos de segunda ordem, o SCG elimina a necessidade de uma busca unidimensional a cada iteração. Isso significa que ele não precisa ajustar a "distância" do movimento a cada passo, tornando o processo mais rápido e eficiente.
5) Uso de Aproximação da Matriz Hessiana: O SCG usa uma técnica que aproxima a matriz Hessiana (que contém informações sobre a curvatura da função de erro) para ajudar na convergência sem sobrecarregar o sistema com cálculos complexos.
6) Parâmetros de Regularização: Dois parâmetros são usados para controlar a estabilidade do algoritmo e a qualidade da aproximação da matriz Hessiana. Eles ajudam a evitar que o algoritmo se desestabilize durante a otimização.
7) Função de Erro: O algoritmo utiliza a função de erro de entropia cruzada, que mede a diferença entre os valores reais e os valores previstos, ajustando os pesos para reduzir essa diferença. [14] [17]

O Gradiente Conjugado Escalonado também foi utilizado para o treinamento da RNA através da ferramenta *Neural Network Getting Started* a partir do código *trainscg*.

3.1.3 Levenberg-Marquardt

O algoritmo de Levenberg-Marquardt (LM) é um método de otimização usado para ajustar uma função matemática a um conjunto de dados experimentais e é particularmente útil em problemas de ajuste de curvas, como regressão não linear e treinamento de redes neurais, onde a função de erro é não linear em relação aos parâmetros do modelo. Em termos simples, ele ajuda a encontrar os parâmetros

certos em uma equação para que ela se ajuste melhor aos dados que temos e funciona desta forma:

1) Suposição Inicial: Começamos com uma suposição inicial para os parâmetros da nossa função. Esta suposição pode ser baseada em estimativas iniciais ou valores padrão.

2) Cálculo do Erro: Calculamos o erro entre os valores reais dos dados e os valores que a função com os parâmetros atuais produz. Quanto maior o erro, pior é o ajuste da função aos dados.

3) Ajuste dos Parâmetros: O algoritmo de Levenberg-Marquardt ajusta os parâmetros da função de uma maneira inteligente. Ele leva em consideração não apenas a magnitude do erro, mas também a sensibilidade dos parâmetros. Parâmetros que afetam mais a função recebem ajustes maiores.

4) Decisão de Ajuste: Dependendo de quão bem a função está se ajustando aos dados após os ajustes, o algoritmo decide se deve aumentar ou diminuir a magnitude dos ajustes nos parâmetros. Ele faz isso de uma maneira adaptativa, ou seja, aprendendo com os ajustes anteriores para tomar decisões melhores.

5) Iteração: O processo de ajuste, cálculo do erro, ajuste dos parâmetros e decisão de ajuste é repetido várias vezes. A cada iteração, a função se ajusta melhor aos dados, reduzindo gradualmente o erro.

6) Convergência: O algoritmo continua iterando até que o erro atinja um nível aceitável ou não melhore significativamente. Nesse ponto, os parâmetros da função são considerados como os melhores ajustes possíveis para os dados dados. [9] [18]

O algoritmo de Levenberg-Marquardt foi o que obteve os melhores resultados para o treinamento da RNA e é denominado no MATLAB como *trainlm*. O desenvolvimento do método LM aplicado às Redes Neurais Artificiais será descrito a seguir.

Dada uma rede neural com parâmetros (ou pesos) $\theta = [\theta_1, \theta_2, \dots, \theta_n]$, queremos ajustar θ para minimizar a função de erro quadrático $E(\theta)$, que é dada por:

$$E(\theta) = \frac{1}{2}\sum_{i=1}^{m}(y_i - f(x_i, \theta))^2$$

onde:

- m é o número total de amostras;
- y_i é o valor de saída desejado para a entrada x_i;
- $f(x_i, \theta)$ é a saída da rede para a entrada x_i com parâmetros θ.

O método LM ajusta os pesos iterativamente para minimizar $E(\theta)$ usando a seguinte atualização para θ:

$$\theta_{k+1} = \theta_k - (J^T J + \lambda I)^{-1} J^T e$$

onde:

- J é a matriz Jacobiana das derivadas de $f(x_i, \theta)$ em relação a cada parâmetro θ, ou seja, $J_{ij} = \frac{\partial f(x_i, \theta)}{\partial \theta_j}$;
- e é o vetor de erros, onde $e_i = y_i - f(x_i, \theta)$;
- λ é o parâmetro de ajuste que controla o equilíbrio entre o gradiente descendente e o método de Gauss-Newton;
- I é a matriz identidade.

A matriz Jacobiana J contém as derivadas da saída da rede $f(x_i, \theta)$ em relação a cada peso θ_J. Essa matriz fornece uma medida de como pequenas variações nos pesos afetam a saída da rede. Especificamente:

$$J = \begin{bmatrix} \frac{\partial f(x_1, \theta)}{\partial \theta_1} & \cdots & \frac{\partial f(x_1, \theta)}{\partial \theta_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f(x_m, \theta)}{\partial \theta_1} & \cdots & \frac{\partial f(x_m, \theta)}{\partial \theta_n} \end{bmatrix}$$

O vetor de erros e é definido como:

$$e = \begin{bmatrix} y_1 - f(x_1, \theta) \\ y_2 - f(x_2, \theta) \\ \vdots \\ y_m - f(x_m, \theta) \end{bmatrix}$$

Esse vetor representa o desvio entre a saída desejada e a saída real da rede para cada ponto de dados.

O parâmetro λ ajusta a influência do gradiente descendente versus o método de Gauss-Newton. Para valores grandes de λ, a atualização aproxima-se do gradiente descendente, garantindo que os passos de ajuste sejam pequenos e a convergência seja mais estável. Para valores menores de λ, a atualização aproxima-se do método de Gauss-Newton, que converge mais rapidamente perto do mínimo. [9] [18]

A atualização de λ é feita dinamicamente:

- Se uma iteração reduz $E(\theta)$, λ é reduzido (normalmente, λ é dividido por um fator $v > 1$);
- Se uma iteração aumenta $E(\theta)$, λ é aumentado (normalmente, multiplicado por v).

A redução em $E(\theta)$ para quando $E(\theta)$ for menor que um limite pré-definido ou se o número máximo de iterações for atingido. [9] [18]

4 RESULTADOS E DISCUSSÃO

O código utilizado no MATLAB, responsável por obter a configuração otimizada da RNA, é apresentado na Figura 2:

Figura 2: Código de programação para obtenção da RNA com melhor desempenho.

```
% Automatizar a Rede Neural para Reconhecimento de Padrões
% Loop até que a acurácia seja maior que 88%

% Este script assume que as seguintes variáveis estão definidas:
%   MatrizdeEntrada - dados de entrada.
%   MatrizdeSaida - dados de saída (alvos).

x = MatrizdeEntrada;
t = MatrizdeSaida;

% Escolher a Função de Treinamento
trainFcn = 'trainlm';  % Levenberg-Marquardt

% Número de neurônios na camada oculta
hiddenLayerSize = 47;

% Critério de parada
targetAccuracy = 0.88;
accuracy = 0;  % Inicialização

while accuracy <= targetAccuracy
    % Criar a Rede de Reconhecimento de Padrões
    net = patternnet(hiddenLayerSize, trainFcn);

    % Configurar a Divisão dos Dados para Treinamento, Validação e Teste
    net.divideParam.trainRatio = 70/100;
    net.divideParam.valRatio = 15/100;
    net.divideParam.testRatio = 15/100;

    % Treinar a Rede
    [net, tr] = train(net, x, t);

    % Testar a Rede
    y = net(x);
    tind = vec2ind(t);
    yind = vec2ind(y);

    % Calcular a acurácia
    accuracy = sum(tind == yind) / numel(tind);

    % Exibir resultados intermediários
    fprintf('Acurácia atual: %.2f%%\n', accuracy * 100);
end

% Exibir a Rede com Melhor Desempenho
fprintf('Acurácia de %.2f%% alcançada! Rede treinada com sucesso.\n',
accuracy * 100);
view(net);

% Plots
% Descomente as linhas abaixo para habilitar os diversos gráficos.
% figure, plotperform(tr)
% figure, plottrainstate(tr)
% figure, ploterrhist(gsubtract(t, y))
% figure, plotconfusion(t, y)
% figure, plotroc(t, y)
```

Fonte: Elaborado pela autora, 2024.

O modelo de Rede Neural Artificial mais eficaz, treinado a partir da base de dados *Glass Identification*, foi desenvolvido utilizando-se o algoritmo Levenberg-Marquardt. O treinamento do modelo foi interrompido ao alcançar o critério de parada "Verificações de Validação", com valor igual a 6. Esse critério indica que o treinamento

deve ser encerrado após o modelo errar a classificação de 6 amostras no conjunto de validação.

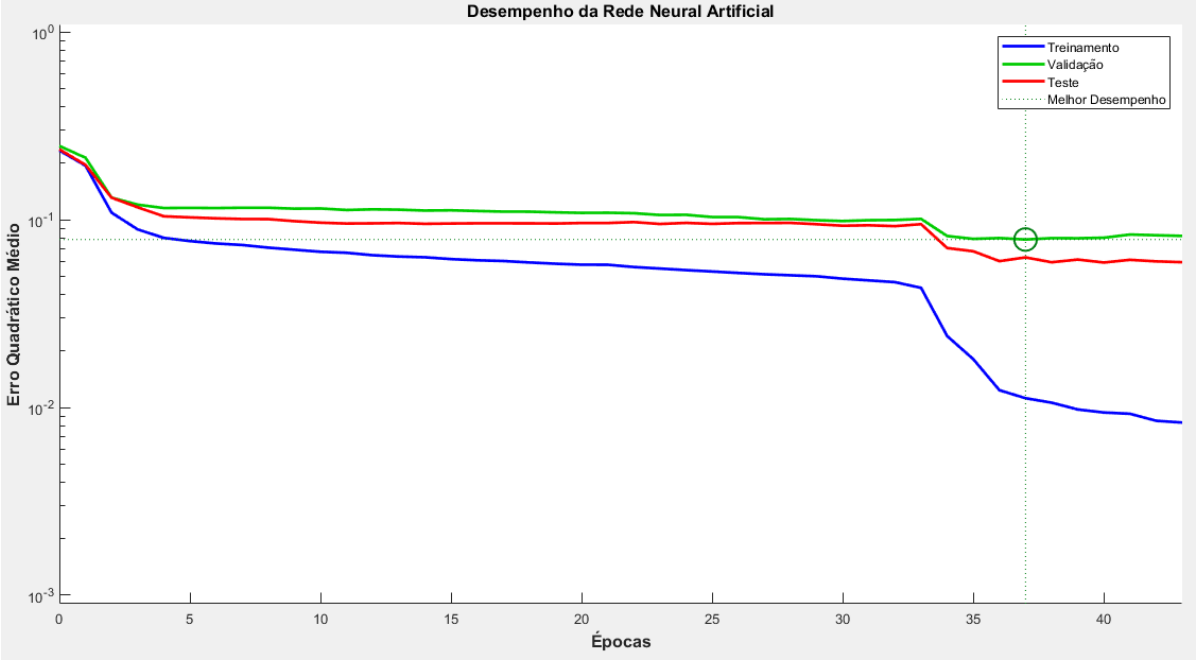
O modelo resultante apresentou uma arquitetura com 47 neurônios na camada oculta e alcançou uma acurácia de 90,19%.

A análise detalhada dos gráficos gerados pelo MATLAB, incluindo Curva de Desempenho, Curvas ROC e Matriz de Confusão, oferece insights significativos sobre o desempenho do modelo obtido. Esses resultados serão explorados e discutidos individualmente nos tópicos subsequentes, destacando a eficácia da RNA na classificação de vidros com base no teor de óxidos.

4.1 Desempenho da Rede Neural Artificial

Uma curva de desempenho da Rede Neural Artificial é uma representação gráfica que mostra a evolução do erro da rede neural ao longo das épocas de treinamento. O objetivo é observar como o erro quadrático médio (ou outra métrica de desempenho) diminui conforme o treinamento avança, indicando que a rede está aprendendo a realizar suas previsões de forma mais precisa.

Figura 3: Gráfico de Desempenho da Rede Neural Artificial



Fonte: Elaborado pela autora, 2024.

O gráfico de desempenho representado pela Figura 3 exibe como o erro quadrático médio (em escala logarítmica) varia conforme o modelo é treinado, validado e testado ao longo de várias épocas.

A linha azul mostra a evolução do erro quadrático médio no conjunto de treinamento. A curva apresenta uma redução contínua no erro ao longo das épocas, com uma queda acentuada após a época 30. Esse comportamento indica que a rede neural está aprendendo os padrões presentes nos dados de treinamento de forma eficaz. Contudo, após a época 37, o erro de treinamento continua a diminuir, enquanto os erros de validação e teste se estabilizam. Esse comportamento é um indicativo de *overfitting*, onde a rede começa a ajustar-se excessivamente aos dados de treinamento, sem ganhos adicionais em termos de generalização.

A linha verde corresponde ao erro no conjunto de validação, que é usado para monitorar o desempenho do modelo em dados não vistos durante o treinamento. O menor erro de validação é atingido na época 37, com um valor de 0,078554, que é destacado no gráfico. Após esse ponto, o erro de validação permanece praticamente estável, sugerindo que o modelo atingiu o melhor ponto de generalização e que treinar além desse limite não proporciona melhorias significativas no desempenho.

O erro no conjunto de teste, representado pela linha vermelha, segue um comportamento semelhante ao da validação. Há uma diminuição gradual do erro até a época 37, com estabilização após esse ponto. A proximidade entre os erros de validação e teste demonstra que a rede neural conseguiu generalizar bem para novos dados, validando a eficácia do treinamento.

O melhor desempenho da RNA foi alcançado na época 37, com um erro quadrático médio de 0,078554 no conjunto de validação. Esse ponto marca o equilíbrio entre aprendizado e generalização, evitando grandes discrepâncias entre os erros nos diferentes conjuntos. Contudo, o fato de o erro de treinamento continuar a diminuir enquanto os erros de validação e teste se mantêm estáveis sinaliza um leve *overfitting*, que poderia ser mitigado utilizando técnicas como o *early stopping*, encerrando o treinamento no momento de melhor desempenho.

Em resumo, a análise das curvas revela que a rede neural apresentou desempenho satisfatório, com capacidade de aprendizado dos dados de treinamento e boa generalização para dados desconhecidos. A proximidade entre os erros de

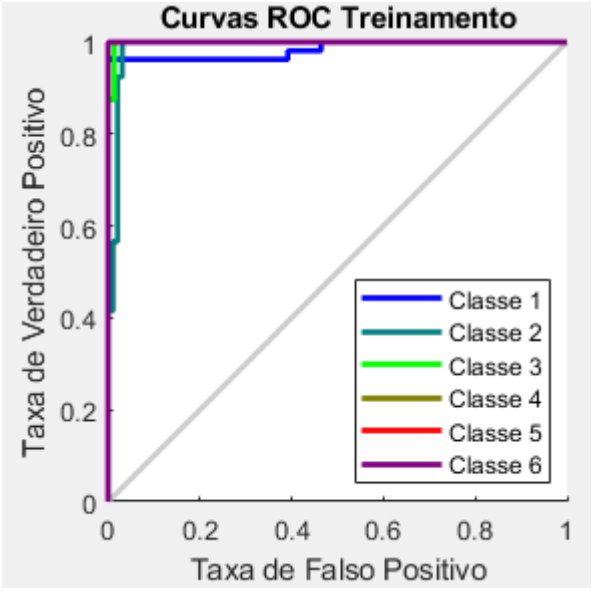
validação e teste reforça que a RNA foi eficaz em minimizar o erro de predição sem perder sua capacidade de extrapolar padrões para novos conjuntos de dados.

4.2 Curva ROC

A curva ROC (Receiver Operating Characteristic) é uma ferramenta amplamente utilizada para avaliar o desempenho de classificadores, particularmente em problemas de classificação binária ou multiclasses. Ela permite a visualização da capacidade de distinção do modelo entre as diferentes classes, traçando a relação entre a taxa de verdadeiros positivos e a taxa de falsos positivos.

No eixo Y da curva ROC, encontramos a taxa de verdadeiros positivos, que mede a proporção de exemplos corretamente classificados como positivos pelo modelo. Esta métrica também é conhecida como sensibilidade ou *recall*. No eixo X, por outro lado, temos a taxa de falsos positivos, que quantifica a proporção de exemplos negativos incorretamente classificados como positivos. A linha diagonal cinza presente nos gráficos representa um classificador aleatório, no qual a probabilidade de acerto é equivalente à de erro. Uma curva ROC ideal se aproxima do canto superior esquerdo do gráfico, indicando alta sensibilidade (verdadeiros positivos) e baixa taxa de falsos positivos.

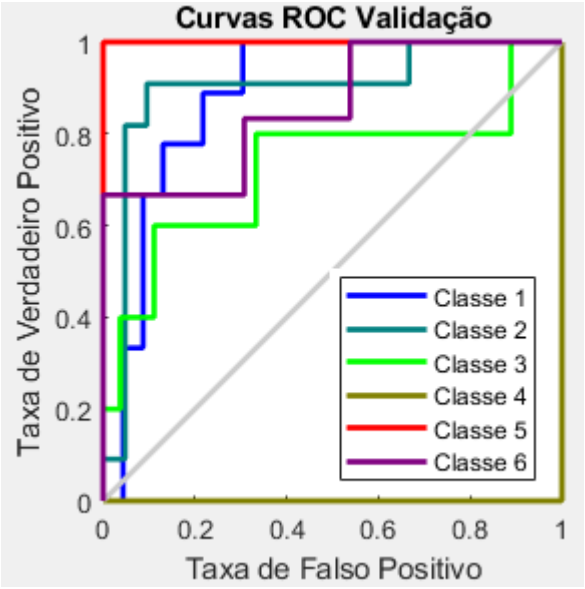
Figura 4: Curvas ROC para o conjunto treinamento.



Fonte: Elaborado pela autora, 2024.

Ao analisar o gráfico gerado para o conjunto treinamento, apresentado na Figura 4, observa-se que o modelo apresenta alto desempenho, com taxas de verdadeiros positivos (eixos verticais) muito próximas de 1 para todas as classes, o que indica que o modelo teve alto poder discriminativo durante o treinamento, separando bem as classes. Isso é esperado, pois o modelo geralmente apresenta o melhor desempenho no conjunto de treinamento, uma vez que foi ajustado diretamente nesses dados.

Figura 5: Curvas ROC para o conjunto validação.



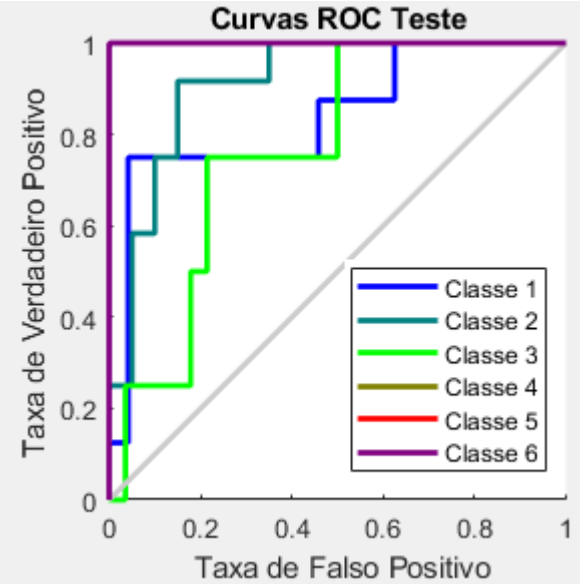
Fonte: Elaborado pela autora, 2024.

No conjunto de validação, mostrado na Figura 5, o desempenho ainda é alto, principalmente para algumas classes específicas (classe 5), mas as curvas começam a apresentar uma leve queda em comparação ao treinamento. Algumas curvas têm uma leve inclinação em direção à linha de não discriminação (linha cinza diagonal), indicando um pequeno aumento na taxa de falsos positivos para algumas classes.

Esse comportamento sugere que o modelo está generalizando bem, mas já há um pequeno indicativo de perda de desempenho fora do conjunto de treinamento, o que pode indicar o início de *overfitting* para algumas classes. A presença de diferentes inclinações para cada classe também indica que algumas classes podem ser mais difíceis de distinguir corretamente no conjunto de validação (como por exemplo, a classe 3).

A linha marrom, que é a Curva ROC para a classe 4, está abaixo da diagonal, indicando que não houve exemplos da classe 4 no conjunto de validação. Sem dados da classe 4 para validar, o modelo não pôde calcular as taxas de verdadeiros positivos e falsos positivos para essa classe, resultando na ausência de uma curva ROC real para esta classe.

Figura 6: Curvas ROC para o conjunto teste.

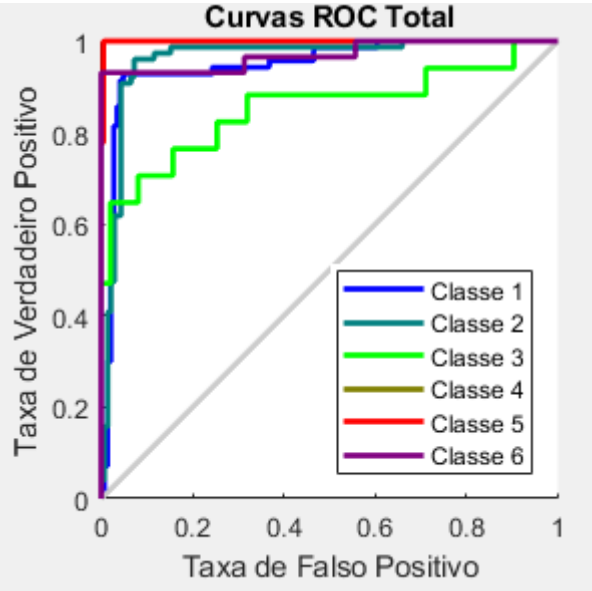


Fonte: Elaborado pela autora, 2024.

No conjunto de teste, apresentado na Figura 6, vemos que algumas curvas ainda têm alto desempenho, como as curvas 4, 5 e 6, que atingem a curva ideal, mas há classes que apresentam uma ROC menos pronunciada, com taxas de falso positivo

relativamente maiores. No geral, a RNA apresentou um desempenho satisfatório no conjunto de teste, com bom aprendizado e capacidade de generalização para a maioria das classes. Contudo, o desempenho mais baixo em algumas classes sugere a necessidade de ajustes adicionais, como balanceamento dos dados ou otimização dos hiperparâmetros, para melhorar a consistência da classificação entre todas as classes.

Figura 7: Curvas ROC para o total dos dados.



Fonte: Elaborado pela autora, 2024.

A Figura 7 exibe a curva ROC total, que consolida todos os conjuntos, mostra um alto desempenho global, com a maioria das classes apresentando áreas sob a curva (AUC) próximas de 1. No entanto, nota-se que a curva da classe 3 não está totalmente próxima do canto superior esquerdo, o que indica uma performance mais fraca em relação a esta classe em específico.

Esse gráfico de desempenho total confirma que o modelo é, em geral, bom para a maioria das classes, mas ainda há diferenças de precisão entre elas, o que é comum em problemas de classificação multiclasse.

No geral, a Curva ROC Total demonstra que a RNA obteve um desempenho satisfatório, com boa capacidade de classificação para a maioria das classes. A proximidade das curvas ao ideal para grande parte das classes evidencia a eficácia do modelo na generalização dos padrões. Contudo, a classe 3 aponta oportunidades

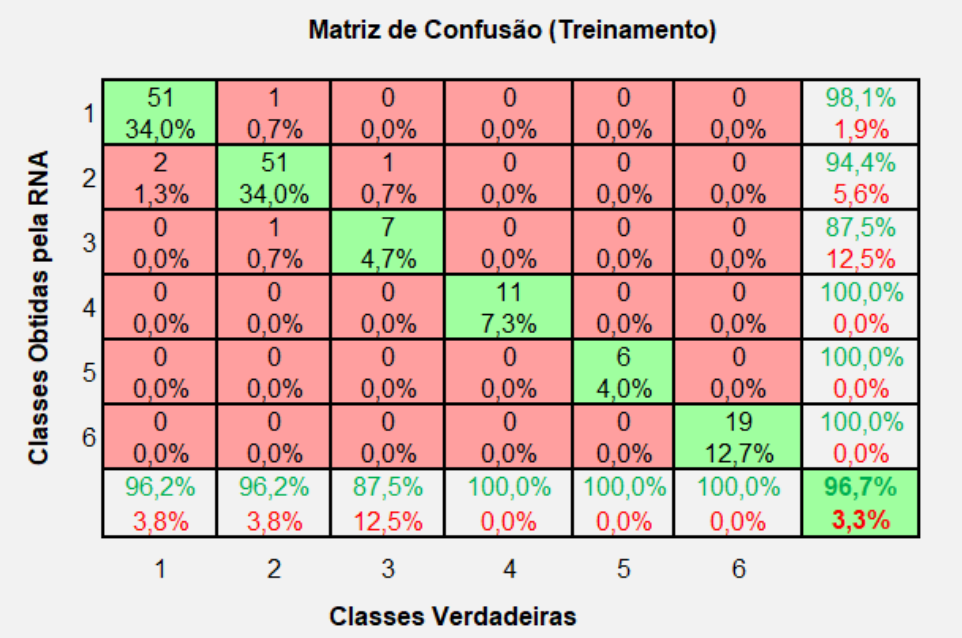
de melhoria, como ajustes na arquitetura da rede ou balanceamento dos dados, para garantir um desempenho mais homogêneo entre todas as classes.

4.3 Matriz de Confusão

Uma matriz de confusão é uma ferramenta usada para avaliar o desempenho de um modelo de classificação. Ela organiza as previsões do modelo em uma tabela que mostra quantas vezes o modelo acertou ou errou ao classificar exemplos em diferentes classes. Isso facilita a análise detalhada do desempenho do modelo para cada classe, permitindo entender onde ele está acertando e onde está cometendo erros.

Em uma matriz de confusão, as linhas representam as classes preditas pelo modelo, enquanto as colunas representam as classes verdadeiras (ou seja, a classe real do exemplo). As quatro matrizes de confusão apresentadas (Treinamento, Validação, Teste e Total) mostram o desempenho do modelo em classificar corretamente e incorretamente cada uma das classes.

Figura 8: Matriz de confusão para o conjunto treinamento.



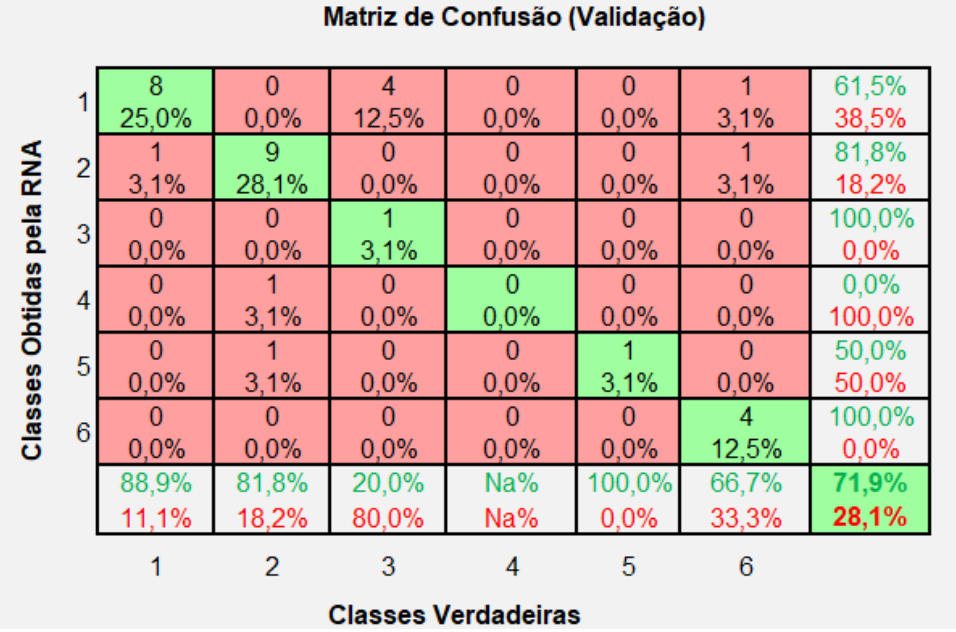
Fonte: Elaborado pela autora, 2024.

Na matriz de confusão do treinamento, representada pela Figura 8, vemos uma alta taxa de acertos para as classes mais comuns. Por exemplo:

- **Classes 1 e 2**: Desempenho alto com 51 acertos (96,2%) e 2 erros (3,8%).
- **Classes 4, 5 e 6**: Todos os exemplos foram classificados corretamente (100% de acurácia).
- **Classes 3**: Apresentou o desempenho mais baixo, com um índice de acerto de 87,5% e 12,5% de erro, o que pode indicar uma maior dificuldade da RNA em discriminar essa classe em relação às demais.

Em geral, a matriz indica que o modelo aprendeu bem as características das classes no conjunto de treinamento, o que é esperado, pois ele foi ajustado nesses dados. No entanto, o excelente desempenho pode sugerir um leve *overfitting*, pois o modelo pode ter se ajustado muito bem aos dados de treinamento.

Figura 9: Matriz de confusão para o conjunto validação.



Fonte: Elaborado pela autora, 2024.

Conforme a Figura 9, a matriz de validação mostra um desempenho inferior ao do conjunto de treinamento, como esperado, pois os dados de validação não foram diretamente utilizados para ajuste do modelo. Observações importantes incluem:

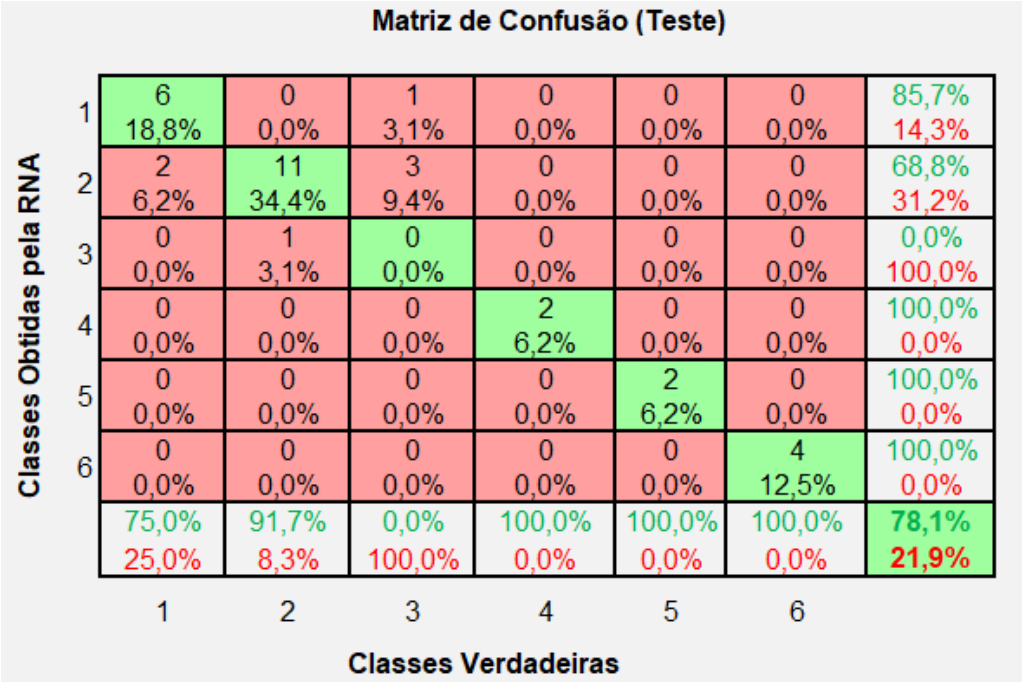
- **Classes 1 e 2**: Apresentam alguns erros de classificação, com taxas de acerto de 88,9% e 81,8%, respectivamente. Ainda assim, são relativamente altas,

indicando que o modelo está generalizando razoavelmente bem para essas classes.

- **Classe 3**: Possui 5 exemplos, os quais apenas 1 foi classificado corretamente (acurácia de 20%), reforçando o fato de que a RNA tem dificuldade em discriminar esta classe em específico.
- **Classe 4**: Não possui exemplos no conjunto de validação, refletindo-se nos valores "NA" (não aplicável).
- **Classe 5**: Apesar de possuir apenas 1 exemplo, teve uma taxa de acerto de 100%.
- **Classe 6**: Apresenta taxa de acerto razoável, de 66,7%.

A presença de erros nas classes principais (1, 2 e 6) sugere que o modelo já enfrenta uma certa dificuldade para generalizar, confirmando a hipótese de *overfitting* observada no conjunto de treinamento.

Figura 10: Matriz de confusão para o conjunto teste.



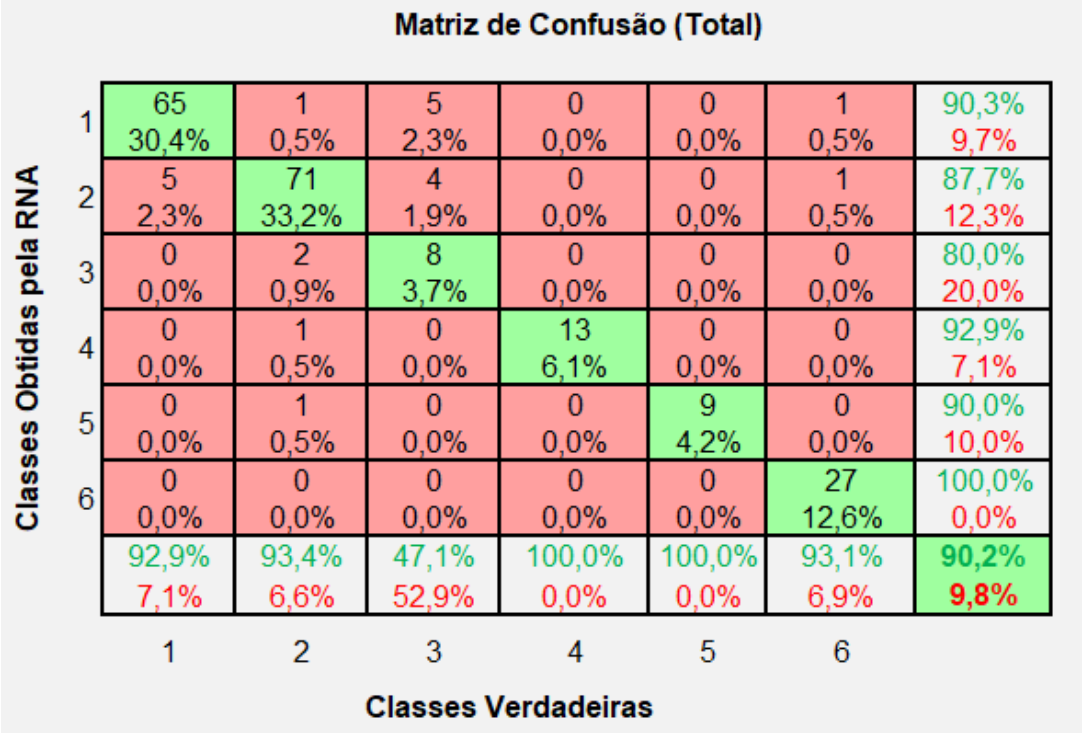
Fonte: Elaborado pela autora, 2024.

A matriz do conjunto de teste, apresentada na Figura 10, fornece uma visão mais clara sobre a capacidade de generalização do modelo:

- **Classe 1**: Desempenho ainda razoável, mas agora com mais erros (taxa de acerto reduzida para 75%). Isso sugere que o modelo tem uma queda significativa de desempenho em dados novos, especialmente em comparação com o treinamento.
- **Classe 2**: Bom desempenho, com taxa de acerto de 91,7%.
- **Classe 3**: Nenhum acerto (acurácia de 0%) indicando que a RNA não conseguiu aprender os padrões específicos desta classe.
- **Classes 4, 5 e 6**: Todos os exemplos foram classificados corretamente (100% de acurácia).

O desempenho inconsistente nas classes, especialmente nas classes 1 e 3, destaca problemas de generalização. Isso indica que o modelo poderia se beneficiar de ajustes para evitar *overfitting* e melhorar a robustez.

Figura 11: Matriz de confusão para o total dos dados.



Fonte: Elaborado pela autora, 2024.

A matriz de confusão total, mostrada na Figura 11, consolida os resultados dos três conjuntos (Treinamento, Validação e Teste):

- **Classe 1**: Apresentou 92,9% de acerto total, com a maioria das amostras corretamente classificadas (65 acertos). Houve 7,1% de erro, com as principais confusões ocorrendo com a classe 2. Isso indica que, embora a rede tenha boa capacidade de identificar essa classe, ainda existem casos de confusão com classes vizinhas.
- **Classe 2:** Alcançou uma acurácia total de 93,4%, com 71 amostras corretas. No entanto, 6,6% das amostras foram classificadas incorretamente. Esse resultado reflete um bom desempenho, mas indica uma leve dificuldade na discriminação em relação a classes próximas.
- **Classe 3:** Apresentou um desempenho inferior, com 47,1% de acerto total (8 acertos). Houve 52,9% de erro, com amostras sendo frequentemente confundidas com as classes 1 e 2. Esse resultado revela uma limitação da RNA em capturar as características específicas dessa classe.
- **Classe 4**: Apresentou uma acurácia total de 100%, com 13 acertos em todas as amostras. Não houve erros, o que demonstra que a RNA tem excelente capacidade de discriminar os padrões dessa classe.
- **Classe 5**: Apresentou um desempenho excelente, com 100% de acurácia total e 9 amostras classificadas corretamente. Não houve nenhum erro, evidenciando uma excelente capacidade da RNA de identificar essa classe.
- **Classe 6**: Obteve 93,1% de acerto total, com 27 amostras classificadas corretamente e 6,9% de erro. Os erros foram mínimos e se restringiram a confusões com outras classes próximas, indicando um bom desempenho geral para essa classe.

A matriz consolidada indica um desempenho global satisfatório da RNA, com acurácia total de 90,19%. As classes 4 e 5 se destacam, com 100% de acurácia total, demonstrando que a rede possui alta capacidade de generalização para esses padrões. Em contrapartida, a Classe 3 apresentou o pior desempenho, com apenas 47,1% de acerto, evidenciando dificuldades significativas na identificação dessa classe e um alto índice de confusão com outras classes.

De forma geral, a rede neural demonstra um bom desempenho, mas as limitações na Classe 3 sugerem a necessidade de otimizações adicionais, como ajuste de

hiperparâmetros, balanceamento dos dados e refinamento da arquitetura da rede, para melhorar a capacidade de discriminação dessa classe. A matriz também reforça a consistência da RNA para classes bem representadas, mas destaca desafios na generalização para classes com menor número de exemplos ou características similares a outras.

A Tabela 4 apresenta os valores de classificação obtidos pela Rede Neural Artificial (RNA) para cada amostra do conjunto de teste, juntamente com o resultado de classificação previsto pelo modelo. Para efeito de comparação, são também exibidos os valores e as classes reais dessas amostras. As linhas em cinza representam as amostras que foram incorretamente classificadas pela RNA, enquanto as linhas em branco indicam as amostras que foram corretamente atribuídas à classe esperada.

Tabela 4: Comparação de dados e resultado do conjunto treinamento entre o obtido e esperado pela RNA.

Amostra	**Dados de Saída da RNA**						**Resultado Obtido pela**	**Dados Alvo**						
	Classe 1	Classe 2	Classe 3	Classe 4	Classe 5	Classe 6	**RNA**	Classe 1	Classe 2	Classe 3	Classe 4	Classe 5	Classe 6	**Resultado Esperado**
23	0,66	0,06	0,29	0,00	0,00	0,00	Classe 1	1	0	0	0	0	0	Classe 1
32	1,00	0,00	0,00	0,00	0,00	0,00	Classe 1	1	0	0	0	0	0	Classe 1
56	0,00	1,00	0,00	0,00	0,00	0,00	Classe 2	1	0	0	0	0	0	Classe 1
58	0,79	0,10	0,11	0,00	0,00	0,00	Classe 1	1	0	0	0	0	0	Classe 1
59	0,44	0,38	0,17	0,00	0,00	0,00	Classe 1	1	0	0	0	0	0	Classe 1
62	0,00	0,98	0,02	0,00	0,00	0,00	Classe 2	1	0	0	0	0	0	Classe 1
66	1,00	0,00	0,00	0,00	0,00	0,00	Classe 1	1	0	0	0	0	0	Classe 1
67	0,87	0,13	0,00	0,00	0,00	0,00	Classe 1	1	0	0	0	0	0	Classe 1
82	0,10	0,89	0,00	0,00	0,00	0,00	Classe 2	0	1	0	0	0	0	Classe 2
90	0,02	0,98	0,00	0,00	0,00	0,00	Classe 2	0	1	0	0	0	0	Classe 2
95	0,01	0,99	0,00	0,00	0,00	0,00	Classe 2	0	1	0	0	0	0	Classe 2
101	0,00	1,00	0,00	0,00	0,00	0,00	Classe 2	0	1	0	0	0	0	Classe 2
103	0,00	1,00	0,00	0,00	0,00	0,00	Classe 2	0	1	0	0	0	0	Classe 2
111	0,00	0,97	0,00	0,03	0,00	0,00	Classe 2	0	1	0	0	0	0	Classe 2
113	0,00	1,00	0,00	0,00	0,00	0,00	Classe 2	0	1	0	0	0	0	Classe 2
114	0,00	1,00	0,00	0,00	0,00	0,00	Classe 2	0	1	0	0	0	0	Classe 2
127	0,31	0,25	0,44	0,00	0,00	0,00	Classe 3	0	1	0	0	0	0	Classe 2
132	0,00	0,99	0,00	0,01	0,00	0,00	Classe 2	0	1	0	0	0	0	Classe 2
135	0,01	0,95	0,05	0,00	0,00	0,00	Classe 2	0	1	0	0	0	0	Classe 2
140	0,00	1,00	0,00	0,00	0,00	0,00	Classe 2	0	1	0	0	0	0	Classe 2
149	0,35	0,62	0,03	0,00	0,00	0,00	Classe 2	0	0	1	0	0	0	Classe 3
151	0,04	0,95	0,01	0,00	0,00	0,00	Classe 2	0	0	1	0	0	0	Classe 3
158	1,00	0,00	0,00	0,00	0,00	0,00	Classe 1	0	0	1	0	0	0	Classe 3
163	0,00	0,70	0,30	0,00	0,00	0,00	Classe 2	0	0	1	0	0	0	Classe 3
165	0,00	0,26	0,00	0,69	0,00	0,05	Classe 4	0	0	0	1	0	0	Classe 4
173	0,00	0,00	0,00	0,99	0,01	0,00	Classe 4	0	0	0	1	0	0	Classe 4
179	0,00	0,00	0,00	0,00	0,99	0,01	Classe 5	0	0	0	0	1	0	Classe 5
185	0,00	0,00	0,00	0,00	0,99	0,00	Classe 5	0	0	0	0	1	0	Classe 5
194	0,00	0,01	0,00	0,01	0,01	0,97	Classe 6	0	0	0	0	0	1	Classe 6
203	0,00	0,00	0,00	0,00	0,04	0,96	Classe 6	0	0	0	0	0	1	Classe 6
207	0,00	0,01	0,00	0,00	0,02	0,97	Classe 6	0	0	0	0	0	1	Classe 6
213	0,00	0,00	0,00	0,00	0,01	0,99	Classe 6	0	0	0	0	0	1	Classe 6

Fonte: Elaborado pela autora, 2024.

Na tabela acima, é possível identificar com precisão quais das 32 amostras do conjunto de teste foram corretamente classificadas, assim como as 7 amostras classificadas incorretamente, destacadas em cinza.

5 CONSIDERAÇÕES FINAIS

Este trabalho buscou desenvolver um modelo de rede neural artificial (RNA) para classificar amostras de vidro de acordo com suas composições de óxidos, contribuindo com um método de análise para a área forense. No decorrer do estudo, foram cumpridos os objetivos específicos e o objetivo geral proposto, pois a RNA apresentou uma acurácia significativa na classificação dos diferentes tipos de vidro, validando a viabilidade do uso dessa metodologia em contextos de identificação forense. Além disso, a pesquisa conseguiu responder de maneira satisfatória ao problema levantado, demonstrando que, com um treinamento adequado, a rede neural pode ser uma ferramenta valiosa para a classificação de vidros com base em composições químicas.

Os resultados obtidos evidenciaram que o modelo de RNA, treinado com o algoritmo de Levenberg-Marquardt, alcançou uma taxa de acerto de 90,19%, demonstrando uma excelente capacidade de generalização para as amostras de vidro disponíveis. No entanto, desafios significativos foram identificados na classificação da Classe 3, que apresentou o pior desempenho geral (47,1% de acurácia). Esses erros estão relacionados, em grande parte, à escassez de dados representativos e à proximidade composicional dessa classe com outras, dificultando a discriminação do modelo.

Para superar essas limitações e aprimorar o desempenho da RNA, especialmente na Classe 3, sugere-se aumentar a quantidade de amostras por meio de coleta de novos dados ou técnicas de aumento de dados (*data augmentation*). Além disso, a aplicação de métodos de seleção de características (*feature selection*) pode contribuir para melhorar a separabilidade das classes, possibilitando uma classificação mais precisa e homogênea. Esses ajustes são essenciais para fortalecer a capacidade do modelo em lidar com desafios de classes sub-representadas em estudos futuros.

Para trabalhos futuros, outra abordagem seria avaliar o desempenho da RNA sem considerar a Classe 3, verificando se a exclusão desta classe resulta em um aumento na acurácia geral do modelo. Tais ajustes poderão indicar a sensibilidade do modelo a variações nas entradas e contribuir para uma análise mais abrangente da eficácia do método em situações experimentais distintas.

Além disso, observou-se que o treinamento poderia ter sido interrompido próximo às 37 épocas, ponto no qual a RNA atingiu seu melhor equilíbrio entre aprendizado e generalização. Continuar o treinamento após esse ponto potencializou o risco de *overfitting*, evidenciado pela queda de desempenho nos dados de validação e teste, o que impactou na capacidade de generalização do modelo.

Do ponto de vista da formação acadêmica, este projeto ampliou significativamente meus conhecimentos e introduziu conceitos de inteligência artificial e redes neurais artificiais, um campo que até então não fazia parte da minha formação acadêmica. Este estudo representou um desafio e uma oportunidade para expandir minha visão sobre o uso de técnicas avançadas de análise de dados em problemas de engenharia.

6 REFERÊNCIAS BIBLIOGRÁFICAS

[1] Akerman, M.; **Natureza, Estrutura e Propriedades do Vidro**, 2000, pp. 1–37.

[2] Kolb, D.; Kolb, K. E.; J. **Chem. Educ.** 1979, 56, 604.

[3] VIDRO CERTO. **O que é o vidro?** Disponível em: https://vidrocerto.org.br/vc-novo/o-que-e-o-vidro/. Acesso em: 01 de novembro de 2024.

[4] EL-KHATIB, Mohmmad Jamal; ABU-NASSER, Bassem S.; ABU-NASER, Samy S. **Glass Classification Using Artificial Neural Network**. *International Journal of Academic Pedagogical Research (IJAPR)*, v. 3, n. 2, p. 25-31, fev. 2019.

[5] RODRIGUES, Caio H. P.; BRUNI, Aline T. **Vidros automobilísticos como vestígios de cena de crime: uma abordagem multivariada**. *Química Nova*, v. 44, n. 5, p. 553-560, 2021. Disponível em: http://dx.doi.org/10.21577/0100-4042.20170720. Acesso em 04 de novembro de 2024.

[6] O que são Redes Neurais e *Deep Learning*?. **Didática Tech**, 2022. Disponível em: https://didatica.tech/introducao-a-redes-neurais-e-deep-learning/. Acesso em: 03 de novembro de 2024.

[7] NASCIMENTO, Érica Regina Filletti. **Desenvolvimento de modelos neurais para o processamento de sinais acústicos visando a medição de propriedades topológicas em escoamentos multifásicos.** 2007. Tese (Doutorado em Engenharia Mecânica) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2007.

[8] DEEP LEARNING BOOK. **A arquitetura das redes neurais.** Disponível em: https://www.deeplearningbook.com.br/a-arquitetura-das-redes-neurais/. Acesso em: 03 de novembro de 2024.

[9] PRONI, C.; HANEDA, R. N.; FILLETTI, É. R**. Desenvolvimento de redes neurais artificiais para análise do fluxo de permeado de uma bebida à base de açaí no processo de microfiltração tangencial.** *C.Q.D.– Revista Eletrônica Paulista de Matemática*, Bauru, v. 17, p. 189-205, fev. 2020. Edição Ermac. DOI: 10.21167/cqdvol17ermac202023169664cprnherf189205. Disponível em: http://www.fc.unesp.br/departamentos/matematica/revista-cqd/. Acesso em: 01 de novembro de 2024.

[10] O que é o cérebro?. **Cognifit**, 2016. Disponível em: <https://www.cognifit.com/br/cerebro/>. Acesso em: 06 de agosto de 2023.

[11] ERLER, Vinícius Liberato. **Desenvolvimento de uma interface gráfica e de uma rede neural artificial para prever o fluxo de permeado de um sistema de microfiltração tangencial de gomas guar e xantana.** 2023. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Química) – Instituto de Química, Universidade Estadual Paulista "Júlio de Mesquita Filho", Araraquara, 2023. Orientadora: Profa. Dra. Érica Regina Filletti Nascimento.

[12] Capítulo 2 – Uma Breve História das Redes Neurais Artificiais. **DeepLearning Book**, 2018. Disponível em: <https://www.deeplearningbook.com.br/uma-breve-historia-das-redes-neurais-artificiais/>. Acesso em: 03 de novembro de 2024.

[13] BERTO, Tamires Messias; SANTOS, Mônica Cardoso; PEREIRA, Fabíola Manhas Verbi; FILLETTI, Érica Regina. **Artificial neural networks applied to the classification of hair samples according to pigment and sex using non-invasive analytical techniques.** X-Ray Spectrometry, v. 49, n. 6, p. 685-693, 2020. Disponível em: https://doi.org/10.1002/xrs.3163. Acesso em: 02 de novembro de 2024.

[14] ERLER, Vinícius Liberato. **Desenvolvimento de modelos neurais e de florestas aleatórias para classificar grãos de soja, feijão, grão de bico e milho a partir de imagens digitais**. Araraquara, 2023. Relatório final apresentado à FAPESP, como parte das exigências para o término da bolsa de iniciação científica. Orientadora: Profa. Dra. Érica Regina Filletti Nascimento.

[15] German, B.; **Glass Identification.** *UCI Machine Learning Repository*, 1987, https://doi.org/10.24432/C5WW2P. Acesso em 10 de fevereiro de 2023.

[16] FLORIAM, B. G.; PEREIRA, F. M. V.; FILLETTI, Érica R. **Estimating bulk density in leguminous grains with different traits using color parameters from digital images combined with artificial neural networks.** Eclética Química, *[S. l.]*, v. 45, n. 1, p. 11–17, 2020. Disponível em: https://revista.iq.unesp.br/ojs/index.php/ecletica/article/view/1009. Acesso em: 03 de novembro de 2024.

[17] ALMEIDA, Leandro Maciel. **Uma metodologia de busca por Redes Neurais Artificiais quase-ótimas.** 2007. 114 f. Dissertação (Mestrado) - Curso de Ciência da

Computação, Centro de Informática, Universidade Federal de Pernambuco, Recife, 2007.

[18] HAGAN, M. T.; MENHAJ, M. B. **Training feedforward networks with the Marquardt algo-rithm. IEEE Transactions on Neural Networks**, v. 5, n. 6, p. 989-993, 1994.