Pontifícia Universidade Católica
do Rio de Janeiro

Henrique Monteiro de Abreu

Perceptron de múltiplas camadas para a classificação de polímeros a partir de dados de ensaios de tração

Dissertação de Mestrado

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós–graduação em Engenharia Química, de Materiais e Processos Ambientais, do Departamento de Engenharia Química e de Materiais da PUC-Rio.

Orientadora: Profª. Dra. Amanda Lemette Teixeira Brandão
Coorientador: Prof. Dr. José Roberto Moraes d'Almeida

Rio de Janeiro
fevereiro de 2024

Pontifícia Universidade Católica
do Rio de Janeiro

Henrique Monteiro de Abreu

Perceptron de múltiplas camadas para a classificação de polímeros a partir de dados de ensaios de tração

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós–graduação em Engenharia Química, de Materiais e Processos Ambientais da PUC-Rio. Aprovada pela Comissão Examinadora abaixo:

Profª. Dra. Amanda Lemette Teixeira Brandão
Orientadora
Departamento de Engenharia Química e de Materiais – PUC-Rio

Prof. Dr. José Roberto Moraes d'Almeida
Coorientador
Departamento de Engenharia Química e de Materiais – PUC-Rio

Profª. Dra. Andréa Pereira Parente
Escola de Química - UFRJ

Profª. Dra. Laura Hecker de Carvalho
Unidade Acadêmica de Engenharia de Materiais - UFCG

Rio de Janeiro, 26 de fevereiro de 2024

Henrique Monteiro de Abreu

Técnico em Alimentos pelo Instituto Federal de Educação, Ciência e Tecnologia (IFRJ). Bacharel em Engenharia Química pela Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio). Durante a graduação estagiou no Laboratório de Estudos Marinhos e Ambientais (LabMAM) da PUC-Rio. Atua há mais de 10 anos como técnico de laboratório no Departamento de Ciência dos Alimentos da Universidade Federal do Estado do Rio de Janeiro (UNIRIO). Atualmente também exerce o cargo de Analista de Pesquisa e Desenvolvimento na iniciativa Experimentação Ágil, Cocriação e Transformação digital (ExACTa) na PUC-Rio.

Ficha Catalográfica

Abreu, H. M.

Perceptron de múltiplas camadas para a classificação de polímeros a partir de dados de ensaios de tração / Henrique Monteiro de Abreu; orientadora: Amanda Lemette Teixeira Brandão; coorientador: José Roberto Moraes d'Almeida. – 2024.

62 f: il. color. ; 30 cm

Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Química e de Materiais, 2024.

Inclui bibliografia

1. Engenharia Química – Teses. 2. Engenharia de Materiais – Teses. 3. Polímeros. 4. Aprendizagem de máquina. 5. Redes neurais. 6. Perceptron de múltiplas camadas. 7. Classificação. 8. Ensaios de tração. I. Brandão, A. L. T.. II. d'Almeida, J. R. M.. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Química e de Materiais. IV. Título.

CDD: 620.11

Às minhas irmãs Carol, Isa e Sofia,
e à minha sobrinha Clara.

Agradecimentos

Gostaria de agradecer à minha orientadora, profª Amanda Brandão. Seu apoio, confiança, paciência e otimismo foram fundamentais tanto para a realização do mestrado quanto para a oportunidade que tive de seguir trabalhando com pesquisa.

Gostaria de agradecer também ao meu coorientador, profº José d'Almeida, pela sua disponibilidade em acompanhar pessoalmente os ensaios de tração quando o retorno presencial às atividades ainda era parcial, e também pela paciência e disponbilidade de sempre em ensinar.

Aos colegas do Laboratório de Modelagem, Automação e Controle (LaMAC) agradeço por manterem sempre um clima colaborativo de trabalho, e particularmente ao Francisco Strunck pelo companheirismo nas disciplinas que fizemos juntos.

Agradeço à minha mãe pelo apoio e motivação para a escrita deste trabalho, e à toda a minha família por ter sempre me apoiado e incentivado aos estudos. Aos meus amigos e em especial à minha namorada, Naiara Amorim, agradeço por essa caminhada juntos.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Resumo

Abreu, H. M.; Brandão, A. L. T.; d'Almeida, J. R. M.. **Perceptron de múltiplas camadas para a classificação de polímeros a partir de dados de ensaios de tração**. Rio de Janeiro, 2024. 62p. Dissertação de Mestrado – Departamento de Engenharia Química e de Materiais, Pontifícia Universidade Católica do Rio de Janeiro.

O ensaio de tração é o ensaio mecânico mais aplicado para a obtenção das propriedades mecânicas de polímeros. Por meio de um ensaio de tração é obtida a curva tensão-deformação, e é a partir desta curva que são obtidas propriedades mecânicas tais como o módulo de elasticidade, a tenacidade e a resiliência do material, as quais podem ser utilizadas na identificação de comportamentos mecânicos equivalentes em materiais poliméricos, seja para a diferenciação de resíduos plásticos para a reciclagem ou para a classificação de um material plástico reciclado quanto ao teor de um determinado polímero em sua composição. Porém, a obtenção das propriedades mecânicas a partir da curva tensão-deformação envolve cálculos e ajustes nos intervalos da curva em que essas propriedades são determinadas, tornando a obtenção das propriedades mecânicas um processo complexo sem a utilização de programas computacionais especializados. A partir da compreensão do padrão de comportamento da curva tensão-deformação de um material, algoritmos de aprendizagem de máquina (AM) podem ser ferramentas eficientes para automatizar a classificação de diferentes tipos de materiais poliméricos. Com o objetivo de verificar a acurácia de um algoritmo de AM na classificação de três tipos de polímeros, foram realizados ensaios de tração em corpos de prova de polietileno de alta densidade (PEAD), polipropileno (PP) e policloreto de vinila (PVC). O conjunto de dados obtido a partir das curvas tensão-deformação foi utilizado no treinamento de uma rede neural artificial perceptron de múltiplas camadas (PMC). Com uma acurácia de 0,9261 para o conjunto de teste, o modelo obtido a partir da rede PMC foi capaz de classificar os polímeros com base nos dados da curva tensão-deformação, indicando a possibilidade do uso de modelos de AM para automatizar a classificação de materiais poliméricos a partir de dados de ensaios de tração.

Palavras-chave

Polímeros; Aprendizagem de máquina; Redes neurais; Perceptron de múltiplas camadas; Classificação; Ensaios de tração.

Abstract

Abreu, H. M.; Brandão, A. L. T. (Advisor); d'Almeida, J. R. M. (Co-Advisor). **Multilayer perceptron for classifying polymers from tensile test data**. Rio de Janeiro, 2024. 62p. Dissertação de Mestrado – Departamento de Engenharia Química e de Materiais, Pontifícia Universidade Católica do Rio de Janeiro.

The tensile test is the most applied mechanical test to obtain the mechanical properties of polymers, which can be used in polymeric materials classification. Through a tensile test is obtained the stress-strain curve, is from which mechanical properties such as the modulus of elasticity, tenacity, and resilience of the material are obtained, which can be used to identify equivalent mechanical behaviors in polymeric materials, whether for the distinguishing plastic waste for recycling or for classifying recycled plastic material according to the content of a polymer type in its composition. However, obtaining mechanical properties from the stress-strain curve involves calculations and adjustments in the intervals of the curve in which these properties are determined, turning it into a complex process without the use of specialized software. By understanding the behavior pattern of a material's stress-strain curve, machine learning (ML) algorithms can be efficient tools to automate the classification of different types of polymeric materials. To verify the accuracy of an ML algorithm in classifying three types of polymers, tensile tests were performed on specimens made of high-density polyethylene (HDPE), polypropylene (PP), and polyvinyl chloride (PVC). The dataset obtained from the stress-strain curves was used in the training of a multilayer perceptron (MLP) neural network. With an accuracy of 0.9261 for the test set, the model obtained from the MLP neural network was able to classify the polymers based on the stress-strain curve data, thus indicating the possibility of using an ML algorithm to automate the classification of polymeric materials based on tensile test data.

Keywords

Polymers; Machine learning; Neural networks; Multilayer perceptron; Classification; Tensile test.

Sumário

Lista de figuras

Lista de tabelas

Lista de Siglas

ACER Árvore de classificação e regressão

AD Árvores de decisão

AM Aprendizagem de máquina

COR Curva característica de operação do receptor

EA Estímulo adaptativo

FA Floresta aleatória

GDE Gradiente descendente estocástico

kVP k-vizinhos mais próximos

IA Inteligência artificial

MVS Máquina de vetores de suporte

PMC Perceptron de múltiplas camadas

PE Polietileno

PEAD Polietileno de alta densidade

PEBD Polietileno de baixa densidade

PP Polipropileno

PVC Policloreto de vinila

RL Regressão linear

RLo Regressão logística

RNA Redes neurais artificiais

RPMG Resíduos plásticos mal geridos

TSMS Técnica de sobreamostragem minoritária sintética

VC Vapnik-Chervonenkis

Olhem novamente para aquele ponto. É aqui,
é a nossa casa, somos nós.

Carl Sagan, *Pálido ponto azul.*

1. Introdução

Os resíduos plásticos representam um desafio global devido à falta de estratégias de reciclagem e de infraestruturas de gestão de resíduos, principalmente nas regiões menos desenvolvidas [1]. A produção anual global de plásticos foi reportada em cerca de 400 milhões de toneladas no ano de 2021 [2] e entre 60 e 99 milhões de toneladas de resíduos plásticos mal geridos (RPMG) foram reportados para o ano de 2015 [3], o que inclui plásticos jogados nas ruas ou descartados inadequadamente em lixões ou aterros não controlados, chegando ao oceano por meio dos cursos de águas interiores e escoamento de águas residuais, conforme apresentado por Jambeck *et al.* (2015). A Figura 1.1 mostra que apenas 5 países concentraram mais de 50 % de todos os RPMG gerados no mundo em 2019, todos países em desenvolvimento. Mais de 2 bilhões de toneladas de resíduos sólidos urbanos são gerados anualmente, com um aumento estimado para 2,2 bilhões de toneladas e 3,4 mil bilhões de toneladas anualmente até 2025 e 2050, respetivamente. Atualmente, os plásticos são responsáveis por cerca de 7 % a 12 % do peso total de resíduos sólidos urbanos gerados [2]. Portanto, é necessário melhorar as técnicas de gestão de resíduos plásticos para incentivar políticas que possam diminuir os RPMG.

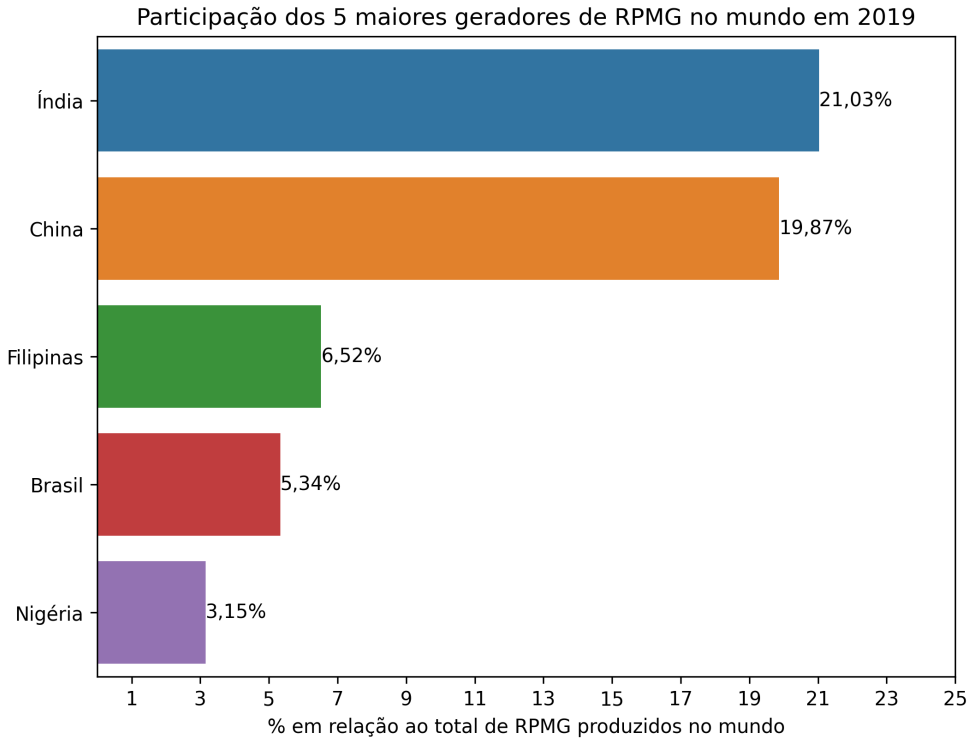


Figura 1.1: Participação dos 5 países que mais geraram RPMG no mundo em 2019. Adaptado de OurWorldInData.org/plastic-pollution.

Em uma economia circular, a reciclagem de plástico faz parte de uma gestão de resíduos plásticos que pode contribuir para a diminuição dos RPMG e ainda criar empregos, especialmente na indústria transformadora, e nas atividades de reutilização e remanufatura [5]. Apesar destes benefícios, a reciclagem de plástico também trouxe novos produtos plásticos que são cada vez mais difíceis de classificar [6]. De acordo com Turku *et al.* (2017), o PEAD, o PP e o PVC estão entre os 4 principais polímeros encontrados em resíduos plásticos. Nesse contexto, investigar métodos precisos e de fácil acesso para classificar materiais plásticos de acordo com a composição polimérica poderia colaborar para resolver a dificuldade de classificação de novos produtos plásticos.

O ensaio de tração é o ensaio mecânico mais aplicado para a obtenção de propriedades mecânicas de polímeros [8]. Por meio de um ensaio de tração é obtida a curva tensão-deformação, resultante da medida da força desenvolvida pela máquina de ensaio de tração à medida que o corpo de prova é alongado a uma taxa constante [9], e é a partir desta curva que são obtidas propriedades mecânicas tais como o módulo de elasticidade, a tenacidade e a resiliência do material. Estas propriedades mecânicas podem ser utilizadas na caracterização de materiais poliméricos, seja para a caracterização de material plástico reciclado de acordo com a sua composição [7] ou para a diferenciação de resíduos plásticos para reciclagem. Porém, a obtenção das propriedades mecânicas a partir da curva tensão-deformação envolve cálculos e ajustes nos intervalos da curva em que essas propriedades são determinadas, tornando a obtenção das propriedades mecânicas um processo complexo, caso não haja a utilização de programas computacionais especializados.

A partir da compreensão do padrão de comportamento da curva tensão-deformação de um material, algoritmos de aprendizagem de máquina (AM) podem ser ferramentas eficientes para automatizar a classificação de diferentes tipos de materiais poliméricos. Para verificar a acurácia de um algoritmo AM na classificação de três tipos de polímeros, foram realizados testes de tração em corpos de prova compostos exclusivamente de polietileno de alta densidade (PEAD), polipropileno (PP) e policloreto de vinila (PVC). O conjunto de dados obtido das curvas tensão-deformação foi aumentado e pré-processado para então ser utilizado no treinamento de uma rede neural perceptron multicamadas (PMC).

2. Objetivos gerais e específicos

Este trabalho tem como objetivo geral obter um modelo capaz de classificar três tipos de polímeros a partir de dados de curvas tensão-deformação, utilizando um algoritmo PMC programado em linguagem Python.

São objetivos específicos deste trabalho:

- realizar ensaios de tração para a obtenção de curvas tensão-deformação em corpos de prova de PEAD, PP e PVC;
- compor um conjunto de dados utilizando os dados das curvas tensão-deformação obtidas em laboratório;
- selecionar as arquiteturas e os hiperparâmetros de modelos de redes neurais;
- treinar cada modelo de rede neural com os dados de treino, e avaliar o desempenho dos modelos pela curva de aprendizagem e acurácia calculada com os dados de validação;
- colocar o melhor modelo em produção para que possa receber novos dados.

3. Fundamentação teórica

Este capítulo busca apresentar os principais conceitos sobre a rede neural PMC. Para isto, é feita uma abordagem de forma hierárquica desde a definição de inteligência artificial, passando pelos diferentes tipos de AM, e por fim o funcionamento de uma rede neural PMC, conforme apresentado no diagrama da Figura 3.1.

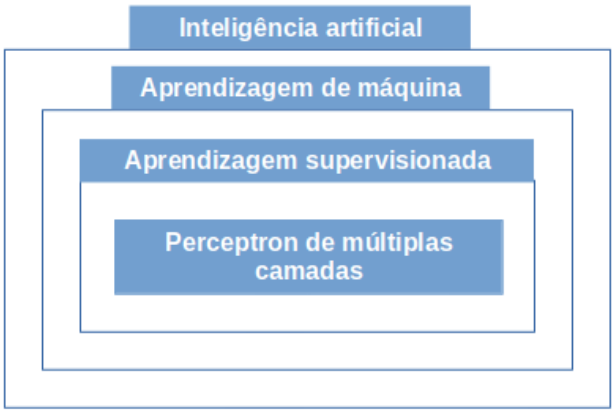


Figura 3.1: Diagrama para a apresentação da fundamentação teórica sobre a rede neural PMC.

3.1 Inteligência artificial

De acordo com Russel e Norvig (2016), inteligência artificial (IA) é, de uma forma geral, o estudo sobre agentes que desempenham ações baseadas em percepções de um ambiente. O Teste de Turing, proposto por Alan Turing (1950) apresenta uma definição prática de inteligência (Figura 3.2). Neste teste, uma pessoa faz perguntas a uma outra pessoa e a um computador, caso o indagador não consiga distinguir se a resposta é de uma pessoa ou uma máquina então o computador passa no teste. Para passar neste teste, é necessário que o computador tenha as seguintes capacidades:

- **processamento de linguagem natural** para permitir a comunicação bem sucedida com o interlocutor;
- **representação de conhecimento** para armazenar o conhecimento adquirido;
- **raciocínio automatizado** para usar as informações armazenadas e então responder à perguntas, e;

– **aprendizagem de máquina** para se adaptar a novas circunstâncias, e para detectar e extrapolar padrões.

Se adicionarmos a interação física entre o indagador e o computador, então se trataria do chamado Teste de Turing total. Para passar neste teste, o computador precisaria ainda de:

– **visão computacional** para distinguir objetos, e;

– **robótica** para manipular objetos e se movimentar.

Estas seis disciplinas fazem parte do campo da IA [10]. Ainda que o teste proposto por Turing permaneça relevante, os pesquisadores em IA não se dedicam a replicar a inteligência em sua totalidade de modo a passar no teste, e sim aos ganhos subjacentes desta inteligência. O advento recente do ChatGPT e outros *chatbots* são exemplos disto, existem muitas perguntas que estas ferramentas não são capazes de responder de forma satisfatória, o que torna suas respostas distinguíveis às de uma pessoa, porém ainda sim têm se mostrado em ferramentas importantes para diversas utilidades, já sendo utilizadas para agilizar serviços de atendimento a clientes de empresas ou mesmo para a geração de códigos em diferentes linguagens de programação.

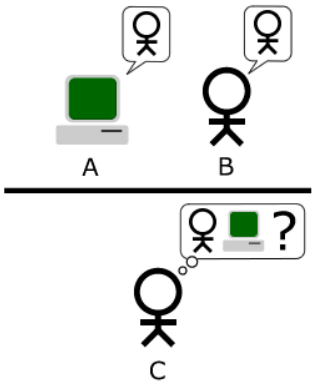


Figura 3.2: No Teste de Turing o indagador (C) tem a tarefa de tentar determinar se as respostas às suas perguntas foram dadas por uma máquina (A) ou uma pessoa (B). Ilustração de domínio público, disponível em `https://commons.wikimedia.org/wiki/File:Turing_Test_version_3.png`.

3.2 Aprendizagem de máquina

Se um agente melhora o seu desempenho em determinadas tarefas após realizar observações sobre o ambiente à sua volta, então pode-se afirmar que houve aprendizado [10]. Conforme apresentado na Figura 3.3, em um modelo simples de aprendizagem de máquina o ambiente fornece uma informação para um *elemento de aprendizagem*. O *elemento de aprendizagem* por sua vez utiliza

a informação para aperfeiçoar a *base de conhecimento*, e por fim o *elemento de desempenho* executa a tarefa a partir da *base de conhecimento* [11].

É comum que a informação oriunda do *ambiente* seja insuficiente para que o *elemento de desempenho* consiga preencher todos os detalhes necessários desta informação ou mesmo que consiga distinguir o que deve ser ignorado. Desta forma, a máquina precisa operar inicialmente por hipóteses para definir parâmetros e depois receber uma *retroalimentação* do *elemento de desempenho*, permitindo então que a máquina avalie suas hipóteses e modifique parâmetros se for necessário.

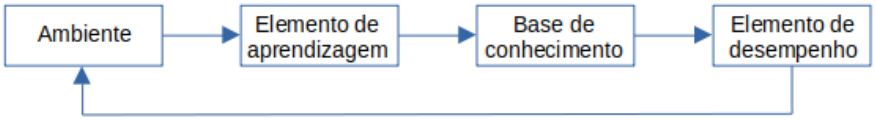


Figura 3.3: Diagrama do processo de aprendizagem de máquina. Adaptado de Haykin (2021).

A partir da forma como é realizada a avaliação das hipóteses e modificação de parâmetros, os algoritmos de AM podem ser divididos em quatro tipos principais: aprendizagem supervisionada, aprendizagem por reforço, aprendizagem não supervisionada e aprendizagem semi-supervisionada [10] [11].

3.3 Aprendizagem supervisionada

Neste tipo de AM, o algoritmo observa exemplos de entradas e saídas de dados e então gera uma função que mapeia as entradas e saídas correspondentes (Figura 3.4).

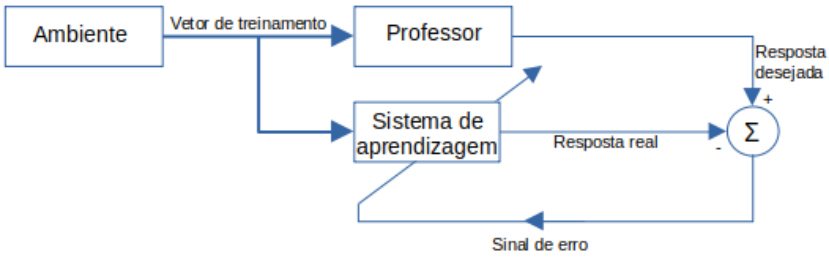


Figura 3.4: Diagrama do processo de aprendizagem supervisionada. Os parâmetros da rede são ajustados conforme a influência combinada do *sinal de erro* e do *vetor de treinamento*. O *sinal de erro* é a diferença entre a resposta desejada e a resposta real do agente. O ajuste é realizado de forma iterativa até que o agente consiga a emulação ótima do *professor*, de acordo com uma determinada métrica estatística. Adaptado de Haykin (2021).

O treinamento de um modelo para classificar se um *e-mail* é ou não um *e-mail* do tipo *spam* pode ser um exemplo de aprendizagem supervisionada [12]. Para treinar este modelo são necessários diversos *e-mails* que contêm a resposta à pergunta "É *spam*?". A partir desta resposta e das características dos e-mails, como palavras-chave, comprimento do texto e presença de *links*, o algoritmo aprende a rotular o conteúdo como um *e-mail* do tipo *spam*. Este é um modelo de classificação binária, uma vez que classifica os *e-mails* em duas categorias, se o *e-mail* é ou não *spam*. O modelo identifica padrões associados a cada a uma das duas categorias em novos *e-mails* recebidos, e os rotula de acordo com o treinamento recebido, permitindo que o gerenciador da caixa de entrada filtre os *e-mails* classificados como *spam*.

3.4 Redes neurais artificiais

O desenvolvimento das redes neurais artificiais (RNA) começou a partir do modelo matemático da atividade neuronal proposto por McCulloch e Pitts (1943), e do mecanismo de aprendizagem por reforço proposto por Hebb (1949) para explicar o aprendizado no cérebro humano [13]. A Figura 3.5 ilustra uma comparação entre um neurônio biológico e o modelo matemático simples da atividade neuronal proposto por McCulloch-Pitts. O modelo matemático é descrito por uma unidade ou nó a qual recebe um conjunto de entradas. Estas entradas são multiplicadas por pesos numéricos os quais determinam a força de ligação entre duas unidades, os resultados são somados e aplicados a uma função de ativação. Quando a função de ativação é também um limiar rígido, *i.e.*, uma função degrau, então o neurônio é chamado de *perceptron* [10] [14].

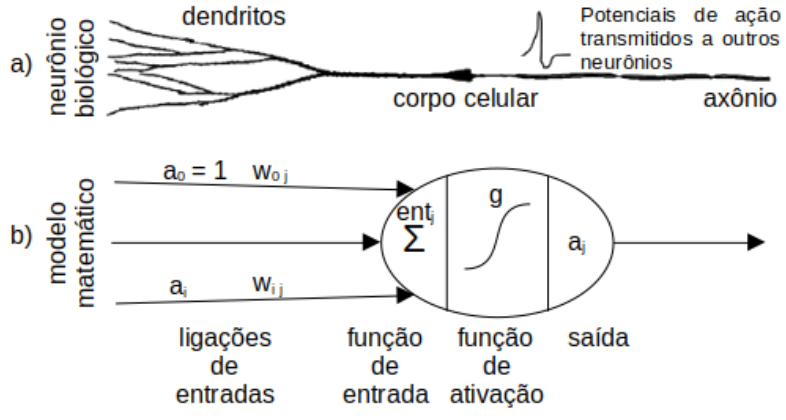


Figura 3.5: Comparação entre um neurônio biológico (a) e um modelo matemático simples para um neurônio (b) proposto por McCulloch-Pitts. A saída do neurônio j é dada por $a_j = g(ent_j) = g(\sum_{i=0}^{n} w_{i,j} a_i)$, onde a_i é a saída do neurônio i e $w_{i,j}$ é o peso numérico do neurônio i para o neurônio j. Adaptado de Aguiar *et al.* (2022).

Existem basicamente duas maneiras distintas de conectar os neurônios entre si em uma rede neural. Na rede neural de alimentação direta (*feed-forward*, os neurônios são conectados em somente uma direção, formando um grafo acíclico (Figura 3.6). Já em uma rede neural recorrente as saídas da rede retroalimentam os neurônios, formando um sistema dinâmico que pode chegar a um estado estacionário. Este tipo de rede neural é capaz de ter memória a curto prazo, tornando esta rede interessante como um modelo do cérebro, porém a sua compreensão é mais complexa comparada a uma rede de alimentação direta.

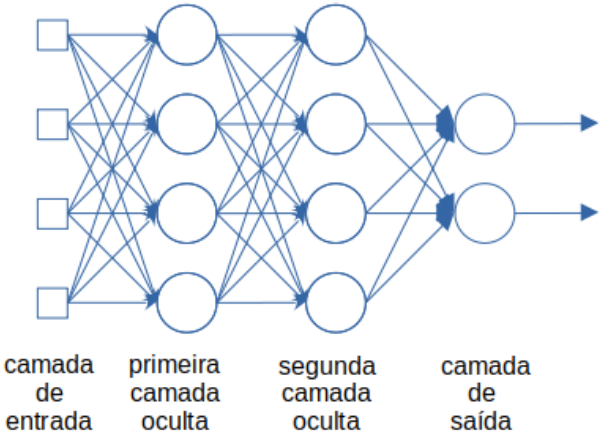


Figura 3.6: Grafo arquitetural de uma rede neural de alimentação direta e múltiplas camadas. Cada círculo corresponde a um neurônio, também chamado de *nó*. Já os quadrados são as variáveis de entrada, também chamadas de *atributos*.

As RNA costumam ser utilizadas na resolução de problemas complexos, onde o comportamento das variáveis não é totalmente conhecido. Quando uma RNA de múltiplas camadas (Figura 3.6) tem mais de três camadas ocultas, a rede é chamada então de *rede neural profunda* e a sua aprendizagem é chamada de *aprendizagem profunda* (*deep learning*) [16]. Uma das principais características de uma RNA é a capacidade de aprender por meio de exemplos e de generalizar a informação aprendida [14].

3.4.1 Perceptron de camada única

Em 1958, Frank Rosenblatt propôs o *perceptron*, o primeiro modelo de aprendizagem supervisionada. O perceptron é uma rede neural de camada única e alimentação direta para a classificação de padrões linearmente separáveis (*i.e.*, padrões localizados em lados opostos de um hiperplano) [10] [11]. Este tipo de rede neural consiste essencialmente de um único neurônio com

pesos numéricos ajustáveis e uma função de ativação que é também um limiar rígido, conforme visto na Figura 3.7.

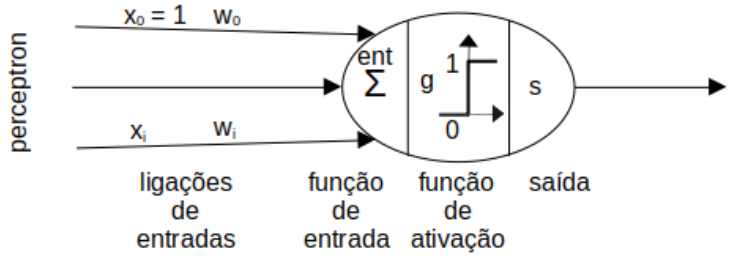


Figura 3.7: Modelo de um perceptron de camada única e alimentação direta proposto por Rosenblatt. A saída do neurônio é igual a 1 se $\sum_{i=0}^{n} x_i w_i \geq 0$ e igual a 0 se $\sum_{i=0}^{n} x_i w_i < 0$, onde $g(ent) = g(\sum_{i=0}^{n} x_i w_i)$, onde x_i é o valor de entrada, w_i é o peso numérico, e s é a saída do neurônio. Aguiar *et al.* (2022).

O ajuste de pesos é realizado conforme a regra de aprendizagem do perceptron, definida pela equação 3-1.

$$w_i \leftarrow w_i + \eta(y - s)x_i \tag{3-1}$$

Nesta equação:
- w_i é o peso da conexão entre o *i-ésimo* valor de entrada e o neurônio;
- η é a taxa de aprendizagem;
- y é o valor verdadeiro que deve ser atingido (*valor alvo*);
- s é o valor de saída obtido do neurônio;
- x_i é o *i-ésimo* valor de entrada.

Desta forma, se $y = s$ então w_i permanece o mesmo. Mas, se $y > s$ então w_i aumenta caso x_i seja positivo, e diminui caso x_i seja negativo. E, se $y < s$ então w_i diminui caso x_i seja positivo, e aumenta caso x_i seja negativo. As iterações do algoritmo de aprendizagem de um perceptron estão representadas na Figura 3.8.

A Figura 3.9 apresenta as operações lógicas de multiplicação (*e*) e de adição (*ou*), e a função lógica *ou-exclusivo* (*xou*), e.g., utilizadas em circuitos digitais [17] [18]. As operações *e* e *ou* são linearmente separáveis, já a função *xou* não é linearmente separável [10]. Desta forma, um perceptron de camada única seria capaz de aprender os padrões de saídas das operações lógicas *e* e *ou*, mas não conseguiria aprender o padrão de saída da função *xou*. Combinar camadas de perceptrons é uma forma de possibilitar o aprendizado nestes casos em que as fronteiras de decisão podem ser curvas complexas [15].

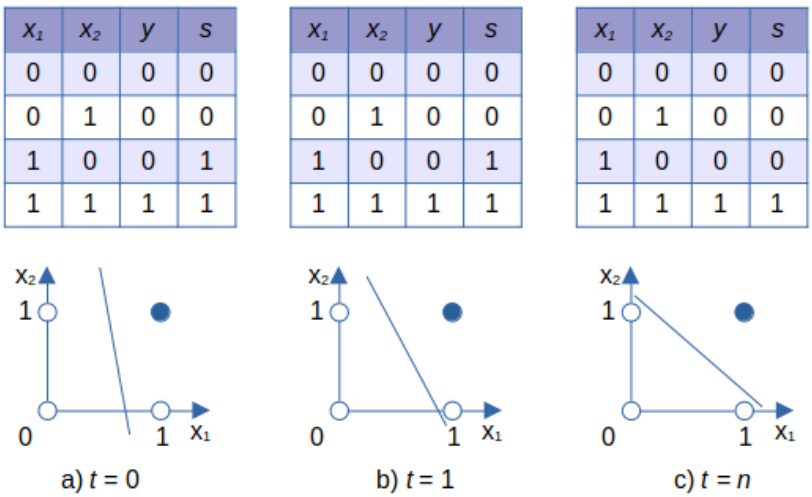


Figura 3.8: Representação de sucessivas iterações do treinamento de um perceptron. Nas tabelas, x_i são valores de entrada, y é o valor alvo e s é o valor de saída do neurônio. Nos gráficos, os círculos preenchidos em azul representam $y = 1$, já os círculos vazios representam $y = 0$, e t é o passo da iteração do treinamento. A área à esquerda da reta representa $s = 0$, e a área à direita da reta representa $s = 1$. (a) Passo inicial do treinamento, uma saída incorreta quando $x_1 = 1$ e $x_2 = 0$. (b) Primeiro ajuste dos pesos aproximando a região de $s = 0$ a $y = 0$ quando $x_1 = 1$ e $x_2 = 0$. (c) Após a *n-ésima* iteração foi possível uma separação linear onde $s = y$.

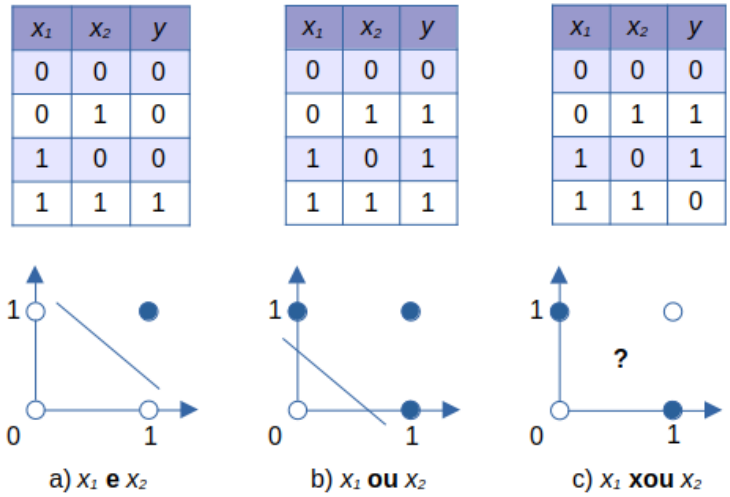
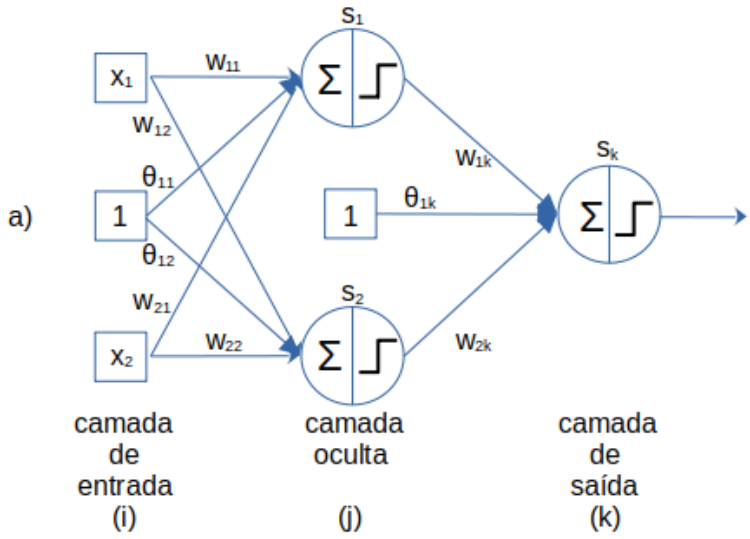
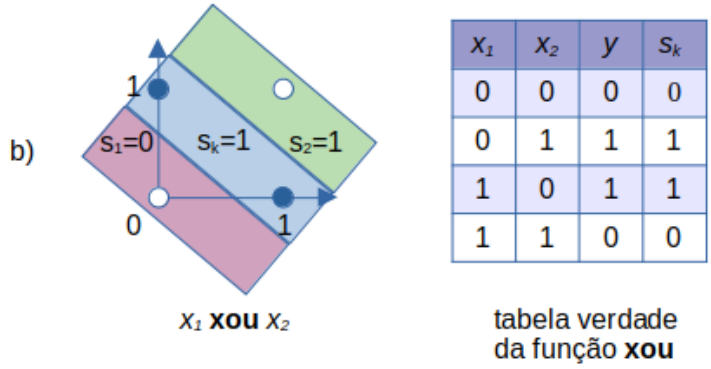


Figura 3.9: Separabilidade linear em operações e funções lógicas. Os círculos preenchidos em azul representam $y = 1$, já os círculos vazios representam $y = 0$. (a) Tabela Verdade e gráfico da separação linear para a operação lógica *e*. (b) Tabela Verdade e gráfico da separação linear para a operação lógica *ou*. (c) Tabela Verdade e gráfico apresentando a impossibilidade de separação linear para a função lógica *xou*. Adaptado de Russel e Norvig (2016), e Vieira (2000).

3.4.2 Perceptron de múltiplas camadas

A rede neural PMC foi desenvolvida a partir da combinação de camadas de perceptrons a fim de superar as limitações do perceptron de camada única (Figura 3.9). Tal como o perceptron de camada única, o PMC também utiliza alimentação direta, *i.e.*, as respostas de uma camada de perceptrons servem de entrada para a próxima camada de perceptrons, conforme o grafo arquitetural de uma rede neural visto na Figura 3.6. Uma rede PMC com uma camada oculta é capaz de aprender o padrão de saída da função lógica *xou* (Figura 3.10).





x_1	x_2	y	s_k
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	0

tabela verdade da função **xou**

Figura 3.10: Problema de classificação da função *xou* e uma solução utilizando uma rede neural PMC. (a) Grafo de um PMC com uma camada oculta, onde x é o valor de entrada da instância de treinamento, w é o peso da conexão entre o valor de entrada e o valor de saída, s é o resultado da função degrau, e θ é o vetor de viés da conexão entre os neurônios de viés e os neurônios artificiais. (b) Tabela Verdade e gráfico apresentando o aprendizado para a função lógica *xou*. No gráfico, os círculos preenchidos em azul representam $y = 1$, já os círculos vazios representam $y = 0$.

O cálculo da função degrau dos perceptrons na camada oculta é definido pelas equações 3-2 e 3-3, e o cálculo da função degrau na camada de saída de um PMC é dado pela equação 3-4.

$$s_1 = \begin{cases} 1, \text{se} \sum_{i=1}^{2} x_i w_{i1} + \theta_{11} \geq 0; \\ 0, \text{se} \sum_{i=1}^{2} x_i w_{i1} + \theta_{11} < 0. \end{cases} \tag{3-2}$$

$$s_2 = \begin{cases} 1, \text{se} \sum_{i=1}^{2} x_i w_{i2} + \theta_{12} \geq 0; \\ 0, \text{se} \sum_{i=1}^{2} x_i w_{i2} + \theta_{12} < 0. \end{cases} \tag{3-3}$$

$$s_k = \begin{cases} 1, \text{se} \sum_{j=1}^{2} s_j w_{jk} + \theta_{jk} \geq 0; \\ 0, \text{se} \sum_{j=1}^{2} s_j w_{jk} + \theta_{jk} < 0. \end{cases} \tag{3-4}$$

Para a solução do problema de classificação da função *xou* a partir da aquitetura de rede PMC proposta na Figura 3.9, são dados os seguintes pesos e vieses numéricos:

$$w_{11} = w_{12} = w_{22} = w_{21} = 1,$$
$$w_{1k} = 0{,}5,$$
$$w_{2k} = -1,$$
$$\theta_{11} = -0{,}5,$$
$$\theta_{12} = -1{,}5,$$
$$\theta_{1k} = -0{,}5.$$

Logo,

$$s_1 = \begin{cases} 1, \text{se } x_1 + x_2 - 0{,}5 \geq 0; \\ 0, \text{se } x_1 + x_2 - 0{,}5 < 0. \end{cases} \tag{3-5}$$

$$s_2 = \begin{cases} 1, \text{se } x_1 + x_2 - 1{,}5 \geq 0; \\ 0, \text{se } x_1 + x_2 - 1{,}5 < 0. \end{cases} \tag{3-6}$$

$$s_k = \begin{cases} 1, \text{se } 0{,}5s_1 - s_2 - 0{,}5 \geq 0; \\ 0, \text{se } 0{,}5s_1 - s_2 - 0{,}5 < 0. \end{cases} \tag{3-7}$$

Portanto, se $x_1 = 0$ e $x_2 = 0$, então:

$$s_1 = 0,$$
$$s_2 = 0,$$
$$0{,}5s_1 - s_2 - 0{,}5 = -0{,}5 < 0, \text{ então } s_k = 0.$$

Se $x_1 = 1$ e $x_2 = 1$, então:

$$\begin{gathered} s_1 = 1, \\ s_2 = 1, \\ 0{,}5s_1 - s_2 - 0{,}5 = -1 < 0, \text{ então } s_k = 0. \end{gathered}$$

Se $x_1 = 0$ e $x_2 = 1$, então:

$$\begin{gathered} s_1 = 1, \\ s_2 = 0, \\ 0{,}5s_1 - s_2 - 0{,}5 = 0 \geq 0, \text{ então } s_k = 1. \end{gathered}$$

E por fim, se $x_1 = 1$ e $x_2 = 0$, então:

$$\begin{gathered} s_1 = 1, \\ s_2 = 0, \\ 0{,}5s_1 - s_2 - 0{,}5 = 0 \geq 0, \text{ então } s_k = 1. \end{gathered}$$

Desta forma, a rede PMC com uma camada oculta soluciona o problema de classificação da função *xou*.

As RNA utilizadas atualmente em tarefas complexas de processamento costumam ser conjuntos de perceptrons organizados em múltiplas camadas (e.g., PMC utilizados em aprendizagem profunda) [15].

4. Revisão bibliográfica

Este capítulo apresenta uma revisão da bibliográfica das aplicações de ensaios de tração em resíduos plásticos, e também da utilização de aprendizagem de máquina na ciência dos materiais.

4.1 Ensaios de tração

Em um ensaio de tração uniaxial utilizado para determinar as propriedades mecânicas dos materiais, um corpo de prova é submetido a uma carga axial que o alonga até a fratura. O comprimento de medição é a região específica do corpo de prova onde as medições da deformação são realizadas (Figura 4.1) [19]. A deformação é uma medida fundamental nestes ensaios e pode ser expressa como a variação percentual no comprimento do corpo de prova em relação ao comprimento original.

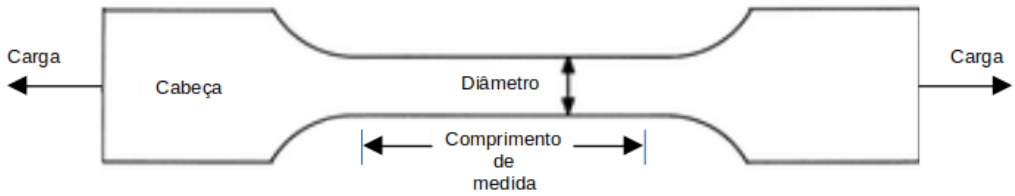


Figura 4.1: Exemplo de um corpo de prova. A seção do comprimento de medição tem menor diâmetro que o das cabeças para que a deformação e falha ocorram nesta região. Adaptado de Davis (2004).

Bajracharya *et al.* (2016) utilizam ensaios de tração para gerar curvas tensão-deformação e caracterizar mecanicamente corpos de prova formados por resíduos plásticos sólidos. De acordo com Bajracharya *et al.* (2016) caracterizar mecanicamente os resíduos plásticos sólidos é importante devido a procura por sua utilização nas áreas da construção civil e *design* de produtos. O estudo aponta que os resíduos plásticos sólidos são compostos em sua maioria por PEAD, PP e polietileno de baixa densidade (PEBD).

Turku *et al.* (2017) utilizam ensaios de tração para comparar propriedades mecânicas entre resíduos plásticos reciclados da construção civil contendo somente PP e PEAD, resíduos plásticos domésticos contendo diferentes tipos de plásticos, e um grupo controle contendo uma mistura de PP e PEAD puros. Neste estudo, foram quantificadas as porcentagens de polietileno (PE) e PP nas amostras por meio da espectroscopia no infravermelho por transformada de Fourier. Em seguinda, foi possível verificar que o aumento da porcentagem de PE presente nos corpos de prova diminui a tensão de escoamento do material

e também o seu módulo de elasticidade. Desta forma, além de serem utilizadas para determinar as propriedades mecânicas, as curvas tensão-deformação poderiam ser utilizadas também para classificar amostras em termos da quantidade de PE na sua composição, bem como a classificação do PE quanto a sua densidade, ou seja, se é PEAD ou PEBD, o que não ocorreu por meio da espectroscopia no infravermelho por transformada de Fourier.

4.2 Aprendizagem de máquina na ciência dos materiais

No campo da ciência dos materiais, Zhu *et al.* (2022) apontam que a AM possibilitou uma mudança de paradigma ao demonstrar sua capacidade em acelerar o desenvolvimento de materiais a custos mais baixos e também de automatizar laboratórios.

Altarazi *et al.* (2019) empregam nove algoritmos de AM para gerar modelos e avaliar as suas capacidades na predição da resistência à tração de filmes poliméricos de diferentes composições e classificação binária dos filmes quanto à conformidade ou não da resistência à tração. Neste estudo foram implementados modelos criados a partir de algoritmos de k-vizinhos mais próximos (kVP); árvores de decisão (AD); RNA; máquina de vetores de suporte (MVS); estímulo adaptativo (EA); floresta aleatória (FA); gradiente descendente estocástico (GDE); e análises de regressão (regressão linear (RL) para predição e regressão logística (RLo) para classificação binária).

A seguir são detalhadas características de cada um dos nove algoritmos de AM utilizados no estudo de Altarazi *et al.* (2019):

- **kVP** é um algoritmo AM supervisionado onde a predição ou classificação dos dados de teste é realizada a partir dos k vizinhos mais correlacionados do conjunto de dados de treinamento. O kVP é não paramétrico, não assume separabilidade linear dos dados, é estável para ligeiras mudanças nos dados e é capaz de aprender com um conjunto pequeno de dados mantendo um desepenho competitivo frente aos demais algoritmos AM [22] [23];
- **AD** é um algoritmo de suporte à decisão de estrutura semelhante a uma árvore, com nós e ramificações. Das as AD disponíveis, a árvore de classificação e regressão (ACER) é uma das mais aplicadas [22] [24] [25]. Para o caso de estudo de Altarazi *et al.* (2019), a ACER é capaz de fornecer informações sobre as relações e interações entre as variáveis de entrada, e portanto, pode ser usado na compreensão do comportamento dos materiais [22] [24];

- As **RNA** são os algoritmos AM não paramétricos e não lineares mais comumente utilizados [22]. Para o caso de estudo de Altarazi *et al.* (2019), foi adotada uma rede neural PMC.
- **MVS** é também um dos mais robustos e acurados algoritmos AM, podendo ser usado em problemas de classificação e regressão. Este algoritmo é baseado na Teoria de Aprendizado Estatístico, a qual está baseada no fato de que o erro de generalização de um modelo é limitado pelo erro de treinamento somado a dimensão Vapnik-Chervonenkis (VC), que por sua vez depende do número efetivo de parâmetros de modelos pertencentes a um determinado espaço de hipóteses. A MVS também é baseada em minimização de risco estrutural, *i.e.*, na transformação da dimensão VC em uma variável controlável, buscando a complexidade ótima do modelo para o tamanho da amostra [22] [23];
- **EA** é um método de aprendizagem por impulsionamento baseado em comitês (*boosting*, neste método ocorre a combinação de diversos classificadores fracos para criar um classificador forte [22];
- A **FA** é outro método baseado em comitês que utilizam a técnica de agregação de *bootstrap*, ou *bagging*, nesta técnica são treinados diversos classificadores paralelamente a partir de amostras independentes, denominadas *bootstrap*, e o classificador final é obtido por meio da média destes classificadores [22] [26];
- **GDE** são algoritmos de otimização amplamente utilizados em AM durante o treinamento de modelos que possuem um grande volume de dados de entrada. Esses algoritmos podem ser combinados com um algoritmo AM para o ajuste de seus parâmetros de forma iterativa, minimizando a função de custo e atingindo um mínimo local [22] [27];
- RL avalia a relação linear entre uma variável dependente e um ou mais preditores independentes. RLo é um método similar a RL, porém utiliza as funções logística ou sigmóide para transformar um número real predito em 0 ou 1 [22].

No estudo de Altarazi *et al.* (2019), os algoritmos que tiveram os melhores desempenhos na predição da resistência à tração foram o MVS, kVP, RNA, ACER, EA e FA, obtendo os maiores valores para os coeficientes de determinação e erro percentual absoluto médio. Já para a classificação binária entre a conformidade ou não da resistência à tração, a rede neural PMC teve o valor mais alto para a área sob a curva característica de operação do receptor (COR), indicando 92,9 % de probabilidade de um filme polimérico

não conforme ser classificado como não conforme. COR é uma curva de sensibilidade (taxa de verdadeiros-positivos) em função da taxa de falsos positivos, e a área sob a curva COR é uma interpretação ituitiva da probabilidade de um caso verdadeiro positivo aleatório ser classificado como positivo e de um caso verdadeiro-negativo ser classificado como negativo por um modelo [28]. O melhor desempenho para a área sob a curva COR no problema de classificação binária e um dos melhores desempenhos para a predição da resitência à tração pela rede neural PMC indicam a robustez do algoritmo para a modelagem de conunto de dados com relações não lineares, principalmente conjuntos de dados multivariados.

Amor *et al.* (2021) utilizam algoritmos de RNA para classificar problemas em processos têxteis e em compósitos poliméricos reforçados com fibras. De acordo com este estudo, modelos tradicionais não são capazes de descobrir relações complexas entre variáveis, e.g. dados multivariados e com relações não lineares, que podem ocorrer entre o desempenho de um tecido e as configurações de uma máquina na produção de uma peça de roupa. Este estudo apresenta ainda uma comparação entre o desenvolvimento de modelos tradicionais determinísticos (*e.g.* modelos matemáticos, modelos empíricos e método dos elementos finitos) e modelos não-determinísticos que utilizam AM. Estas diferenças estão ilustradas na Figura 4.2.

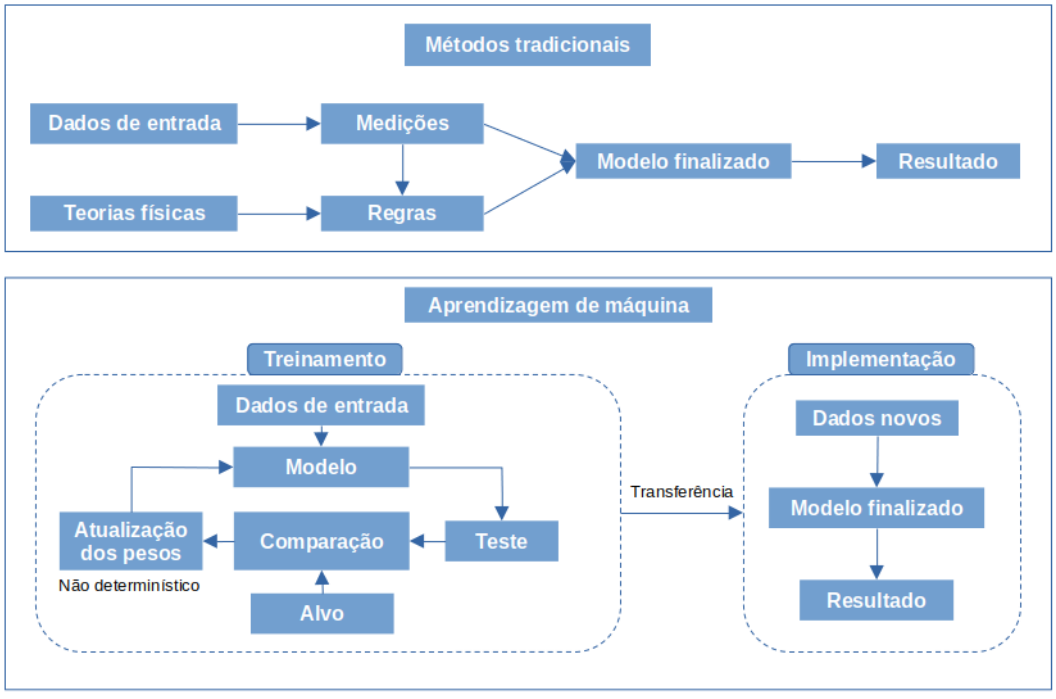


Figura 4.2: Comparação entre os desenvolvimentos de modelos AM e modelos tradicionais determinísticos. Adaptado de Amor *et al.* (2021).

Yousef *et al.* (2011) utilizam RNA para modelar e analisar as propriedades mecânicas de uma mistura de polipropileno (PP) e polietileno (PE) em diferentes proporções. O modelo foi desenvolvido para predizer a curva tensão-deformação de uma dada mistura de PP:PE ao variar a proporção de mistura. Os resultados indicam que uma RNA do tipo PMC pode simular com alta precisão o efeito da proporção de mistura de polímeros no comportamento mecânico e em suas propriedades. O modelo desenvolvido por Yousef *et al.* (2011) é eficaz na predição dos resultados do processo apenas para a faixa elástica da curva tensão-deformação com precisão. No entanto, o modelo teve sucesso na predição da tendência da curva tensão-deformação para a faixa não elástica. Além disso, o modelo RNA desenvolvido é capaz de generalizar, ou seja, pode ser facilmente utilizado várias vezes para realizar o mapeamento não linear de entradas/saídas sem a necessidade de alteração do modelo. Portanto, a utilização de RNA é uma ferramenta eficaz que pode ser adotada para reduzir custo e tempo.

Neto *et al.* (2023) utilizam uma RNA com arquitetura do tipo PMC no desenvolvimento de uma estrutura que processa dados de espectroscopia no infravermelho de materiais poliméricos para a identificação da presença ou não de PP. Neste estudo, são gerados 308 modelos diferentes, considerando combinações distintas de arquiteturas e hiperparâmetros de redes PMC para cada uma das 22 configurações de pré-processamento dos dados realizada. As métricas escolhidas para a seleção do melhor modelo foram a média da acurácia e seus desvios-padrão para 5 resultados da validação cruzada, com diferenças além da segunda casa decimal. As maiores acurácias atingidas ocorreram utilizando a função de ativação Unidade Linear Retificada (`ReLU`) na camada escondida da rede neural. O melhor modelo de classificação binária teve 94,8 % de acurácia e um desvio padrão de 0,512 %. O estudo aponta ainda a importância do pré-processamento dos dados para a melhoria do desempenho dos modelos, especialmente em casos onde a disponibilidade de dados é limitada.

5. Metodologia

A metodologia para a obtenção de um modelo de classificação de polímeros seguiu a hierarquia apresentada no fluxograma da Figura 5.1.

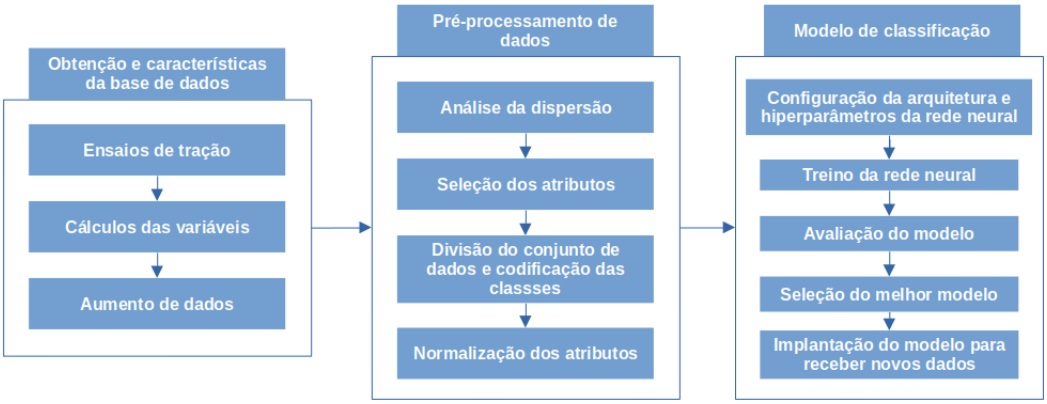


Figura 5.1: Fluxograma das etapas da medologia de obtenção do modelo de classificação de polímeros.

Durante as etapas da metodologia, foram utilizadas as seguintes bibliotecas e programas computacionais:

- Python: versão 3.10.12;
- ScyPy: versão 1.11.4;
- *pandas*: versão 2.1.1;
- Keras: versão 2.15.0;
- DynaView™.

5.1 Obtenção e características da base de dados

Esta etapa compreende a metodologia utilizada para a obtenção de uma base de dados a partir de ensaios de tração.

5.1.1 Ensaios de tração

Utilizando uma máquina universal de ensaios, foram realizados seis ensaios de tração em corpos de prova para cada um dos três polímeros estudados, PEAD, PVC e PP, totalizando 18 ensaios de tração. Todos os ensaios foram realizados a uma velocidade de 50 mm/min, distância entre garras de 40 mm, temperatura de 19,4 °C e umidade realtiva do ar de 61 %. A largura e a espessura do centro dos corpos de prova foram medidas por meio de um paquímetro para o cálculo da área da seção transversal retangular de cada corpo de prova.

Por meio do *software* DynaView™, ao fim de um ensaio de tração foi gerada uma tabela contendo os dados da força aplicada ao corpo de prova para cada intervalo de 0,2 segundos. Foi gerado também um relatório para cada ensaio de tração contendo dados constantes da distância entre garras da máquina universal de ensaios, da área da seção transversal do corpo de prova e da velocidade do ensaio, *i.e.*, a velocidade com que o corpo de prova é alongado.

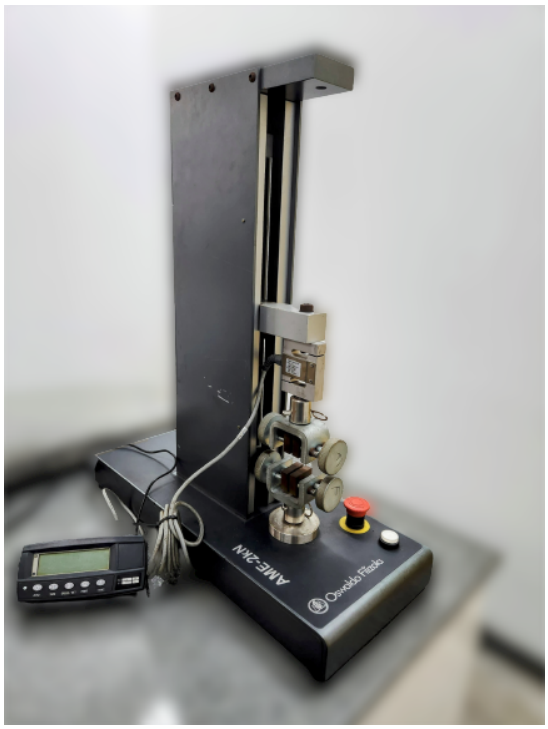


Figura 5.2: Máquina universal de ensaios Oswaldo Filizola, modelo AME-2kN, utilizada nos ensaios de tração.

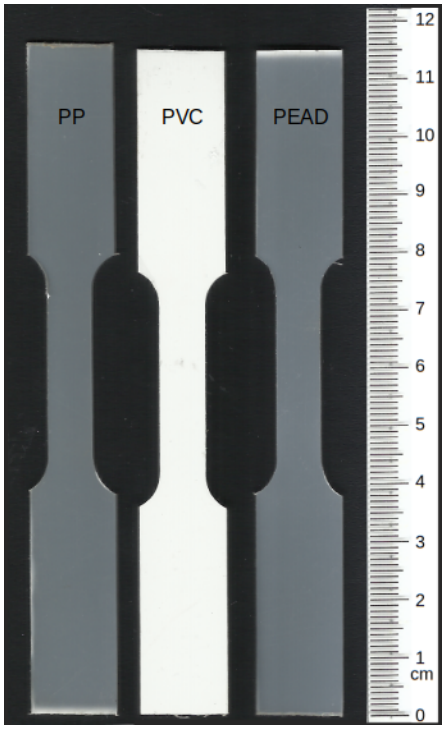


Figura 5.3: Corpos de prova de PEAD, PVC e PP utilizados nos ensaios de tração.

5.1.2 Cálculos de variáveis

A partir dos dados de força e área inicial da seção transversal do corpo de prova foi calculada a tensão aplicada ao corpo de prova (equação 5-1).

$$s = \frac{F}{A_0} \tag{5-1}$$

Onde S é a tensão $[MPa]$, F é a força $[N]$, e A_0 é a área da seção trasnversal do corpo de prova$[mm^2]$.

O alongamento do corpo de prova (ΔL) durante o ensaio de tração foi calculado a partir da velocidade do ensaio e do tempo decorrido de ensaio (equação 5-2).

$$\Delta L = velocidade\ de\ alongamento\ [\frac{mm}{s}] \times tempo\ [s] \tag{5-2}$$

A deformação do corpo de prova durante o ensaio de tração foi calculada a partir do alongamento do corpo de prova e da distância inicial entre garras (equação 5-3).

$$e = \frac{\Delta L\ [mm]}{L_0\ [mm]} \tag{5-3}$$

Onde L_0 é o comprimento inicial do corpo de prova, o qual foi considerada a distância inicial entre garras da máquina universal de ensaios.

A partir da tensão e da deformação calculadas, foram preparadas curvas $tensão \times deformação$ para cada ensaio de tração realizado.

A tenacidade foi calculada a partir da tensão e da deformação, conforme a equação 5-4.

$$tenacidade = \int_0^{e_r} s\ de \tag{5-4}$$

Para o cálculo da integral na equação 5-4 foi utlizada a regra trapezoidal composta cumulativamente por meio do método `cumtrapz` da biblioteca `scipy`, em linguagem Python.

5.1.3 Aumento de dados

Todos os dados obtidos nos ensaios de tração foram então reunidos em uma única estrutura de dados contendo 1905 linhas, por meio da biblioteca `pandas`. Para aumentar a quantidade de dados, foram inseridas à estrutura de dados, a média, a soma da média com o desvio-padrão e a subtração da média pelo desvio-padrão das tensões entre os 6 ensaios de cada classe de polímero. Estes dados de tensão foram inseridos nas curvas tensão-deformação para cada

polímero. As tensões negativas, resultantes do cálculo da subtração da média das tensões pelo desvio padrão, foram excluídas.

A estrutura de dados após o aumento de dados teve uma configuração de 3062 linhas e 8 colunas, conforme mostra a Figura 5.4.

	tempo [s]	força [N]	σ [MPa]	ε [mm/mm]	alongamento [mm]	tenacidade [MPa]	ensaio	classe
0	0.0	0.0	0.000000	0.000000	0.000000	0.000000	1	PEAD
1	0.2	1.1	0.116034	0.004167	0.166666	0.000242	1	PEAD
2	0.4	8.1	0.854430	0.008333	0.333332	0.002264	1	PEAD
3	0.6	17.4	1.835443	0.012500	0.499998	0.007867	1	PEAD
4	0.8	32.0	3.375527	0.016667	0.666664	0.018724	1	PEAD
...	...	...	...	...	...	...	...	...
3057	7.8	392.0	18.789304	0.162499	6.499974	6.130941	DP negativo	PVC
3058	8.0	390.6	17.962255	0.166666	6.666640	6.316215	DP negativo	PVC
3059	8.2	388.7	16.322827	0.170833	6.833306	6.500708	DP negativo	PVC
3060	8.4	386.5	11.952174	0.174999	6.999972	6.684230	DP negativo	PVC
3061	8.6	385.2	2.702452	0.179166	7.166638	6.866923	DP negativo	PVC

3062 rows × 8 columns

Figura 5.4: Estrutura contendo o conjunto de dados obtidos por meio dos ensaios de tração realizados em corpos de prova de PEAD, PP e PVC.

5.2 Pré-processamento de dados

Esta etapa compreende os métodos de seleção, organização e estruturação dos dados.

5.2.1 Análise da dispersão

Para verificar a dispersão entre os dados de tensão de um mesmo polímero, foram calculados os coeficientes de variação dos dados de tensão para cada um dos dados de deformação.

$$\widehat{c_v} = \frac{s}{\bar{x}} \tag{5-5}$$

Onde:

- $\widehat{c_v}$ é o coeficiente de variação de uma amostra;
- s é o desvio-padrão amostral dos dados de tensão de uma dada deformação;
- e $\bar{x}$ é a média amostral dos dados de tensão de uma dada deformação.

Para diminuir a dispersão entre os dados de tensão de um mesmo polímero, os dados de tensão com coeficiente de variação maior que 0,30 para uma dada deformação foram retirados da estrutura de dados.

5.2.2 Seleção dos atributos

Para a seleção dos atributos para treinamento do modelo de classificação foram verificadas as correlações de Spearman entre os 6 atributos da estrutura de dados. A correlação de Spearman avalia relações monotônicas entre duas variáveis. Logo, em uma correlação de Spearman igual a 1 para as variáveis x e y, todos os dados com valores x crescentes terão valores y crescentes também. As correlações de Spearman foram obtidas por meio do método `corr` da biblioteca `pandas`.

5.2.3 Divisão do conjunto de dados e codificação das classes

A estrutura de dados resultante da seleção dos atibutos foi dividida entre um conjunto `X` contendo 3 variáveis de atributos, e uma variável `y` contendo os alvos para o modelo de classificação, ou seja, as classes dos polímeros. A variável `y` foi submetida então ao codificador `LabelEncoder` do pacote `sklearn.preprocessing` para transformar as classes em números inteiros, e desta forma ser possível avaliar os resultados do cálculo da perda de entropia cruzada (*crossentropy loss*) entre os alvos e as predições do modelo a partir da função `SparseCategoricalCrossentropy`, da biblioteca `Keras`.

Em seguida, foi feita uma separação aleatória de 30 % dos dados em `X` e `y` para serem utilizados como conjuntos de dados de teste após o treinamento do modelo. Dos 70 % dos dados restantes, mais 25 % foram separados para a validação cruzada dos resultados do modelo, e o restante dos dados foi utilizado para treinamento do modelo. De forma resumida, subconjuntos de 52,5 %, 17,5 % e 30 % dos dados foram utilizados entre treino, validação e teste, respectivamente. Para estas divisões do conjunto de dados, foram utilizadas as funções `train_test_split` e `StratifiedKFold` do pacote `sklearn.model_selection`. A validação cruzada *k-fold* estratificada permite dividir o conjunto de dados em k grupos proporcionais em relação às classes para a validação do treino (Figura 5.5), foram utilizados 4 *folds* ($k = 4$). Os resultados da validação cruzada *k-fold* foram apresentados e, para reproduzir o resultado de múltiplas chamadas destas funções, foi atribuído o número 42 para o parâmetro `random_state`.

Figura 5.5: Diagrama das etapas de validação cruzada *k-fold*. Com $k = 4$, são realizadas 4 iterações onde é tomado um *fold* por vez para a validação do treinamento.

5.2.4 Padronização dos atributos

Para acelerar o processo de aprendizagem por retropropagação e melhorar os resultados do modelo de classificação, foi realizada a normalização dos subconjuntos de dados de treino, validação e teste a partir do subconjunto de treino. Para a normalização foi utilizada a função `StandardScaler` do pacote `sklearn.preprocessing`. A função `StandardScaler` subtrai as médias de cada atributo do modelo e equaliza as covariâncias entre os atributos.

5.3 Modelo de classificação

Para a implementação da rede neural PMC foi utilizada a classe `Sequential` da biblioteca `Keras`, esta função permite que camadas configuradas isoladamente sejam reunidas em camadas sequencialmente conectadas.

Diferentes configurações de arquitetura e de treino da rede foram testadas, totalizando 56 treinamentos de modelos distintos. A acurácia foi a principal métrica utilizada para avaliar os modelos obtidos a partir das diferentes configurações de arquitetura e de treino testadas, combinada à verificação da curva de aprendizagem do modelo para apurar a inexistência de sobreajustes (*overfitting*) e sub-ajuestes (*underfitting*).

Após a escolha das configurações de arquitetura e de treino da rede, o modelo treinado foi avaliado a partir dos resultados obtidos para a classificação do conjunto de dados de teste nas métricas de precisão, revocação, acurácia e, pontuação-F1, além da verificação da matriz de confundimento.

5.3.1 Configuração da arquitetura da rede neural

A configuração da camada de entrada foi realizada por meio da classe `keras.layers.InputLayer` e a configuração das camadas intermediárias e de saída por meio da classe `keras.layers.Dense`, que por sua vez permite a escolha do número de neurônios e a função de ativação para cada camada. Foram treinadas redes com até 2 camadas ocultas, variando em cada camada a quantidade de neurônios. A função de ativação utilizada nas camadas escondidas foi a Unidade Linear Retificada (`ReLU`), definida na Equação 5-6. Por não haver não ter um valor máximo de saída, a função de ativação `ReLU` permite a redução de instabilidades durante o gradiente descendente [16].

$$f(x) = \begin{cases} x, & x > 0 \\ 0, & x \leq 0 \end{cases} \qquad f'(x) = \begin{cases} 1, & x > 0 \\ 0, & x \leq 0 \end{cases} \tag{5-6}$$

Onde:

- $f(x)$ é a Unidade Linear Retificada.

Na camada de saída foi utilizada a função `softmax` (Equação 5-7). A função `softmax`, ou *regressão logística multinomial*, é uma generalização do modelo de regressão logística para lidar com problemas de classificação multiclasse, sem a necessidade de treinar e combinar classificadores binários [16].

$$\widehat{p}_k = \sigma(s(x))_k = \frac{\exp(s_k(x))}{\sum_{j=1}^{K} \exp(s_j(x))} \tag{5-7}$$

Onde:

- K é a quantidade de classes;
- $s(x)$ é um vetor que contém as pontuações de cada classe para a instância x;
- $\sigma(s(x))_k$ é a probabilidade estimada de que a instância x pertence à classe k, considerando as pontuações de cada classe para essa instância.

5.3.2 Configuração de treino da rede neural

No treinamento do modelo, o objetivo é que o modelo seja capaz de estimar probabilidades altas para a classe-alvo e, portanto, probabilidades baixas para as demais classes. Minimizar a função de custo da entropia cruzada permite penalizar o modelo quando ocorre uma estimativa de probabilidade baixa para uma classe-alvo [16]. No compilador `keras.Sequential.compile`, foi utilizado o método `SparseCategoricalCrossentropy` para o cálculo da função de custo da entropia cruzada, definida pela Equação 5-8.

$$J(\Theta) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{K} y_k^{(i)} \log \widehat{p}_k^{(i)} \tag{5-8}$$

Onde:

- $J(\Theta)$ é a função de custo da entropia cruzada;
- m é a quantidade de instâncias de treinamento;
- K é a quantidade de classes;
- $y_k^{(i)}$ é o valor da *i-ésima* saída que pertence à classe k. O valor é 1 se a saída é igual à classe k, caso contrário, o valor é igual é 0;
- $\widehat{p}_k^{(i)}$ é a probabilidade-alvo de que a *i-ésima* instância pertence à classe k. De maneira geral, é igual a 1 se a instância pertence à classe k, e igual a 0, caso contrário.

E para minimizar o resultado da função de custo da entropia cruzada, foi utilizado o otimizador estocástico `Adam` (Equação 5-9), proposto por Kingma e Ba (2017) [33] [34].

$$\Theta_t = \Theta_{t-1} - \frac{\eta \times \widehat{m_t}}{\sqrt{\widehat{v_t}} + \epsilon} \tag{5-9}$$

Onde:

- Θ_t é o parâmetro Θ atualizado da função objetivo $J(\Theta)$, a ser minimizada;
- Θ_{t-1} é o parâmetro Θ a ser atualizado;
- η é a taxa de aprendizagem;
- $\widehat{m_t}$ é a estimativa de viés do primeiro momento;
- $\widehat{v_t}$ é a estimativa de viés do segundo momento;
- ϵ é um valor baixo, na ordem de 10^{-8}, usado para evitar a divisão por zero na atualização dos pesos.

A taxa de aprendizagem utilizada foi de 10^{-4}, e o tamanho do lote (`batch_size`) para treinamento igual a 8. Os modelos foram treinados variando o parâmetro de paciência entre 2 e 5 na parada antecipada. Também foram utilizados os limites estabelecidos de 0,3, 0,5, 0,7, 1,0, 2,0 e 3,0 para o coeficiente de variação entre as tensões no pré-processamento de dados.

A seleção do melhor modelo foi feita com base na maior acurácia média e menor desvio padrão obtidos na validação do modelo. A acurácia é calculada conforme a Equação 5-10. Esta métrica de avaliação permite avaliar a proporção de predições corretas entre o total de predições realizadas.

$$acurácia = \frac{PV + NV}{PV + NV + PF + NF} \tag{5-10}$$

Onde:

- PV é a quantidade de predições positivas verdadeiras;
- NV é a quantidade de predições negativas verdadeiras;
- PF é a quantidade de predições positivas falsas;
- NF é a quantidade de predições negativas falsas.

A curva de aprendizagem também foi verificada para observar se houve a ocorrência de sobreajustes ou subajustes no treinamento. Por fim, o modelo foi salvo utilizando o método `keras.Sequential.save` para a sua utilização em novos conjuntos de dados.

5.3.3 Avaliação do modelo de classificação

Para avaliar o modelo de classificação selecionado, foram verificadas a acurácia, a precisão, a revocação, e a pontuação-F1, além da verificação da matriz de confundimento, a partir dos resultados obtidos para a classificação do conjunto de dados de teste.

A precisão é calculada conforme a Equação 5-11. Esta métrica de avaliação permite avaliar a proporção de predições positivas que estão corretas para uma dada classe.

$$precisão = \frac{PV}{PV + PF} \tag{5-11}$$

A revocação é calculada conforme a Equação 5-12. Esta métrica de avaliação permite avaliar a proporção entre a quantidade de predições positivas verdadeiras e a quantidade total de saídas verdadeiras para uma dada classe.

$$revocação = \frac{PV}{PV + NF} \tag{5-12}$$

A pontuaçao-F1 é calculada conforme a Equação 5-13. Esta métrica de avaliação permite avaliar a média harmônica ponderada entre a precisão e a revocação. Desta forma, é possível integrar a precisão e a revocação em apenas uma métrica, para o caso em que ambas as métricas de precisão e revocação tenham o mesmo peso na avaliação do modelo.

$$F_1 = \frac{2}{\frac{1}{precisão} + \frac{1}{revocação}} \tag{5-13}$$

A pontuaçao-F1 é calculada conforme a Equação 5-13. Esta métrica de avaliação permite avaliar a média harmônica ponderada entre a precisão e a revocação. Desta forma, é possível integrar a precisão e a revocação em apenas uma métrica, para o caso em que ambas as métricas de precisão e revocação tenham o mesmo peso na avaliação do modelo.

A matriz de confundimento é realizada conforme a Figura 5.6. Esta matriz permite a visualização do desempenho do modelo de classificação, possibilitando ainda a obtenção das informações para o cálculo da acurácia, precisão, revocação e pontuação-F1.

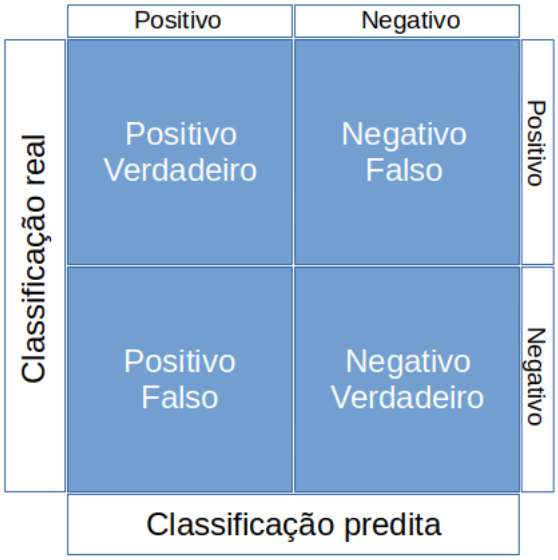


Figura 5.6: Matriz de confundimento em termos abstratos. Nesta matriz são comparadas as colunas contendo as predições de uma classificação binária, com as linhas contendo a quantidade real de saídas para desta classificação. Portanto, predições positivas para uma classe que são de fato positivas constam como Positivo Verdadeiro, predições negativas para uma classe que são na verdade positivas constam como Negativo Falso, predições positivas para uma classe que são na verdade negativas constam como Positivo Falso, e por fim, predições negativas para uma classe que são de fato negativas constam como Negativo Verdadeiro.

6. Resultados e discussões

Neste capítulo são apresentas as curvas tensão-deformação para os 6 ensaios em cada um dos três polímeros estudados. Em seguida, são apresentadas as curvas tensão-deformação após o estabelecimento dos limites superiores de 0,30, 0,50, 0,70, 1,00, 2,00 e 3,00 para os coeficientes de variação entre as tensões. Para a seleção dos atributos utilizando as correlações de Spearman foi gerado um mapa de calor onde é possível verificar as correlações entre todos os atributos. Nos resultados dos treinamentos dos modelos de classificação podem ser verificadas as acurácias e desvios padrão tanto para o treinamento quanto para a validação, bem como as curvas de aprendizagem, ao variar a arquitetura e hiperparâmetros de treino das RNA. Por fim, são apresentas a matriz de confundimento para a predição das classes e relatório de classificação do conjunto de dados de teste do modelo selecionado.

6.1 Ensaios de tração

As Figuras 6.1, 6.2 e 6.3 apresentam as curvas tensão-deformação dos ensaios de tração para o PEAD, PP, e PVC respectivamente.

As áreas das seções transversais retangulares de cada corpo de prova estão disponíveis na Tabela 6.1.

Tabela 6.1: Área da seção transversal do corpo de prova [mm^2] para cada polímero e ensaio de tração.

# Ensaio	PEAD	PP	PVC
1	9,483	8,693	8,914
2	9,481	8,691	8,690
3	9,484	8,103	8,912
4	9,482	8,008	8,808
5	9,485	8,692	8,911
6	9,483	8,002	8,693

Por meio das curvas tensão-deformação do PP e do PVC foi possível observar que na tensão máxima ocorreu um ligeiro aumento da dispersão entre os dados dos ensaios. Isto pode ter ocorrido devido aos corpos de prova terem tido áreas iniciais da seção transversal diferentes para um mesmo polímero. Já durante a deformação plástica dos corpos de prova, logo após o ponto de tensão máxima, houve um aumento substancial da dispersão entre as tensões.

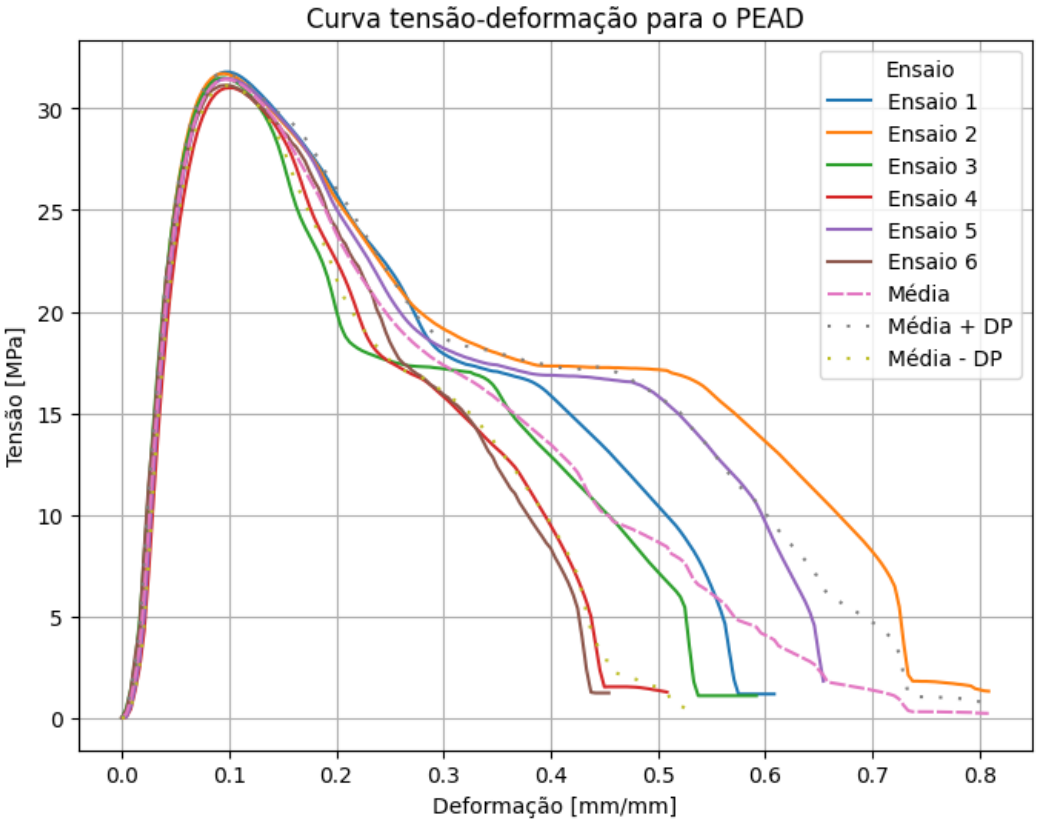


Figura 6.1: Curva tensão-deformação para o PEAD com dados de tensão dos 6 ensaios de tração, das médias das tensões entre os 6 ensaios, e das somas destas médias com os desvios-padrão e da subtração destas médias pelos desvios-padrão.

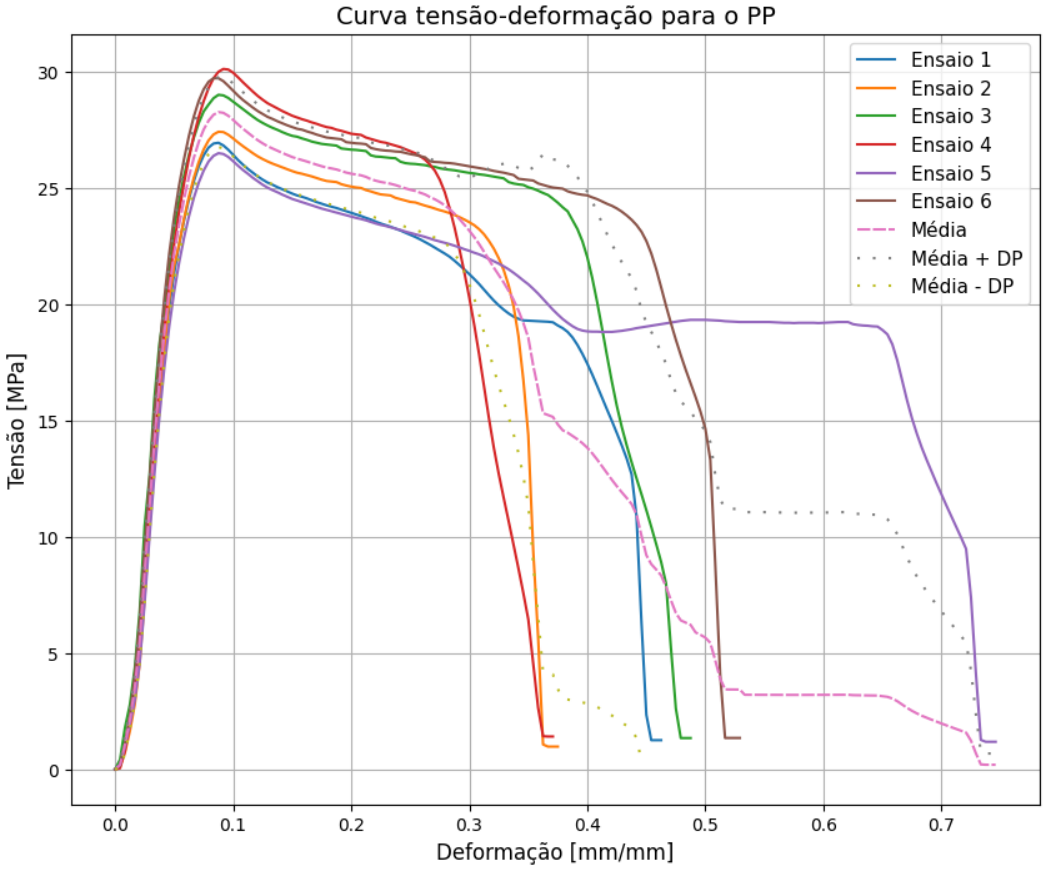


Figura 6.2: Curva tensão-deformação para o PP com dados de tensão dos 6 ensaios de tração, das médias das tensões entre os 6 ensaios, das somas destas médias com os desvios-padrão, e da subtração destas médias pelos desvios-padrão.

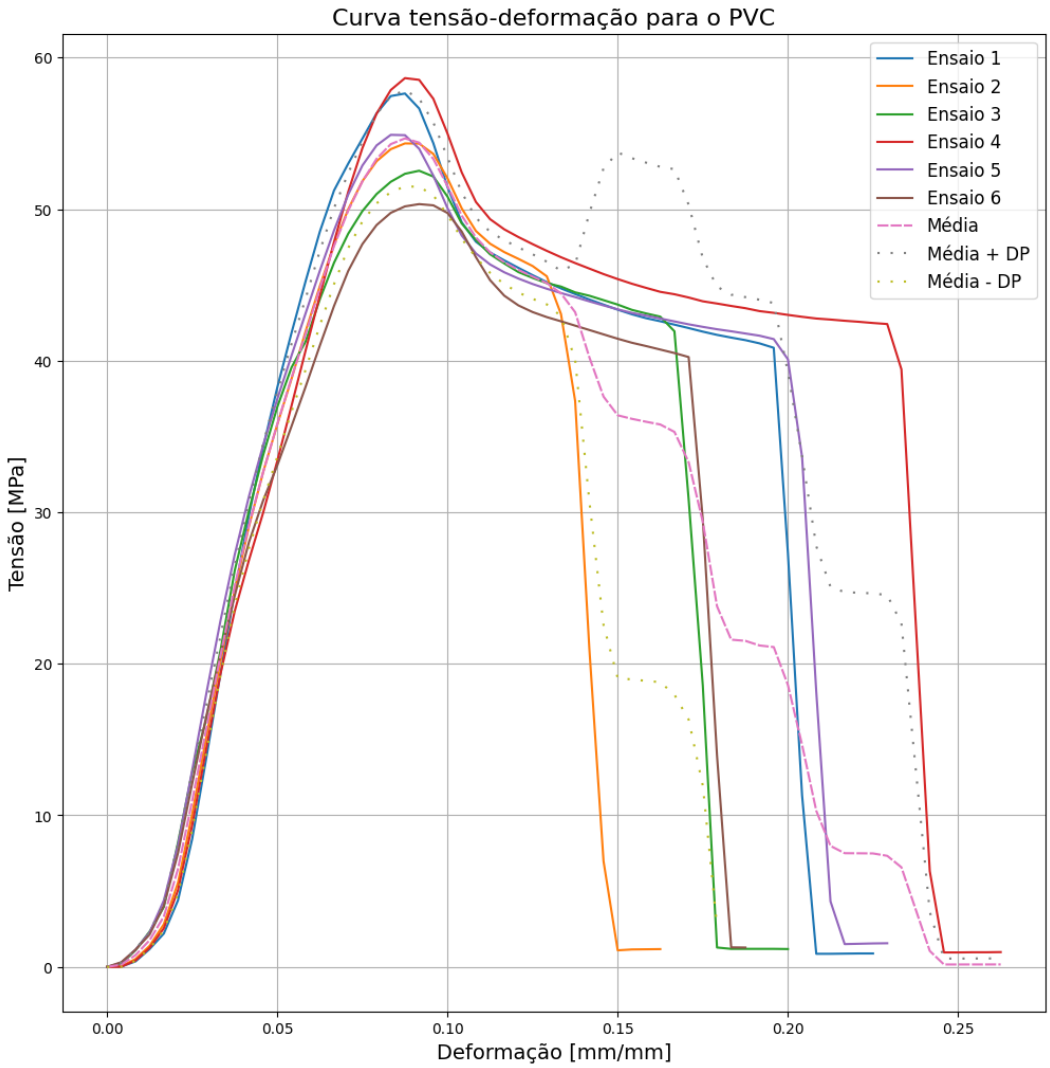


Figura 6.3: Curva tensão-deformação para o PVC com dados de tensão dos 6 ensaios de tração, das médias das tensões entre os 6 ensaios, e das somas destas médias com os desvios-padrão e da subtração destas médias pelos desvios-padrão.

6.2 Pré-processamento de dados

Esta etapa compreende os resultados sobre a seleção, a organização e a estruturação dos dados.

6.2.1 Análise da dispersão

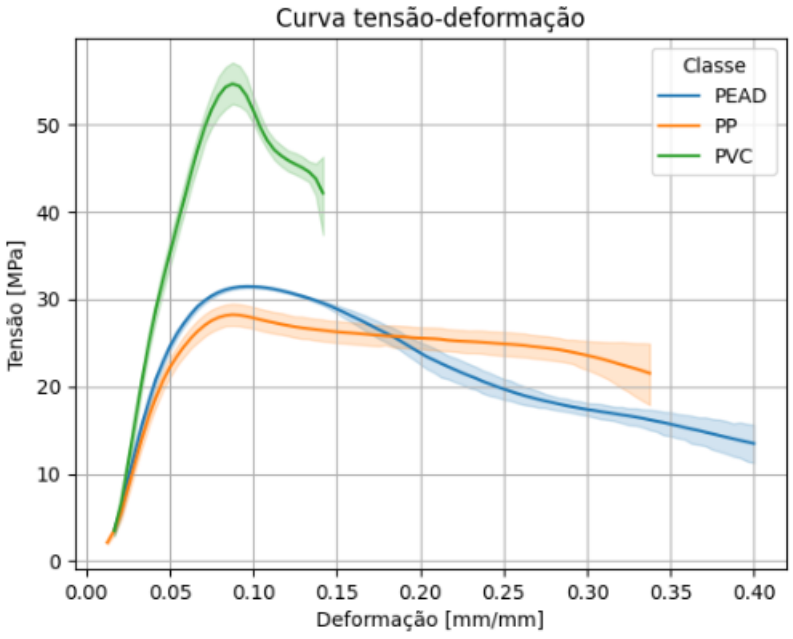
A Figura 6.4(a) apresenta as curvas tensão-deformação após a retirada dos dados de tensão com coeficiente de variação maior que 0,30. É possível observar uma baixa dispersão dos dados de tensão para um mesmo polímero. Desta forma, é esperado que a rede neural tenha mais facilidade na aprendizagem do padrão de comportamento da curva tensão-deformação para um dado polímero, melhorando o ajuste entre as curvas de aprendizagem para os conjuntos de treino e validação.

As Figuras 6.4, 6.5, 6.6 são curvas tensão-deformação após a retirada dos dados de tensão com coeficientes de variação maiores que 0,30, 0,50, 0,70, 1,00, 2,00 e 3,00, respectivamente. Quanto maiores os limites inferiores estabelecidos para os coeficientes de variação, maiores são os intervalos de deformação das curvas tensão-deformação.

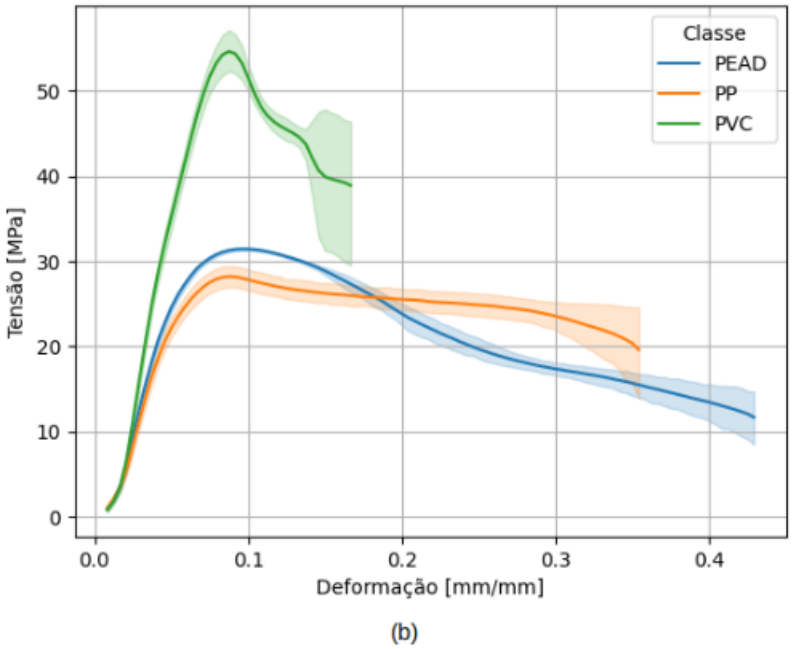
6.2.2 Seleção dos atributos

Na Figura 6.7 o mapa de calor das correlações de Spearman mostra que o tempo do ensaio de tração está totalmente correlacionado ao alongamento e à deformação do corpo de prova, e estes três atributos têm as mesmas correlações com a força e tensão aplicados nos ensaios. Isto se deu devido ao cálculo da deformação incluir o alongamento, e o cálculo do alongamento por sua vez incluir o tempo, como pode ser obeservado nas equações 5-2, 5-3. Desta forma, entre o tempo, o alongamento e a deformação, foi selecionada apenas a deformação como atributo para treinamento do modelo de classificação. O mesmo ocorre para a correlação de 0,95 verificada entre a força e a tensão, na equação 5-1 o cálculo da tensão inclui a força.

A tenacidade tem uma correlação de 0,99 com a deformação, isto ocorre devido ao cálculo da tenacidade incluir a deformação, conforme a equação 5-4, e a deformação por sua vez ter uma taxa de crescimento constante. Já a correlação entre a tensão e a tencidadade é de apenas -0,43. Diferente da deformação, a tensão não tem uma taxa de variação constante ao longo do tempo, e como o método `cumtrapz` calcula a tenacidade cumulativamente, a relação entre a tenacidade e a curva tensão-deformação não é pontual e sim dada por intervalos de $[0, e]$.

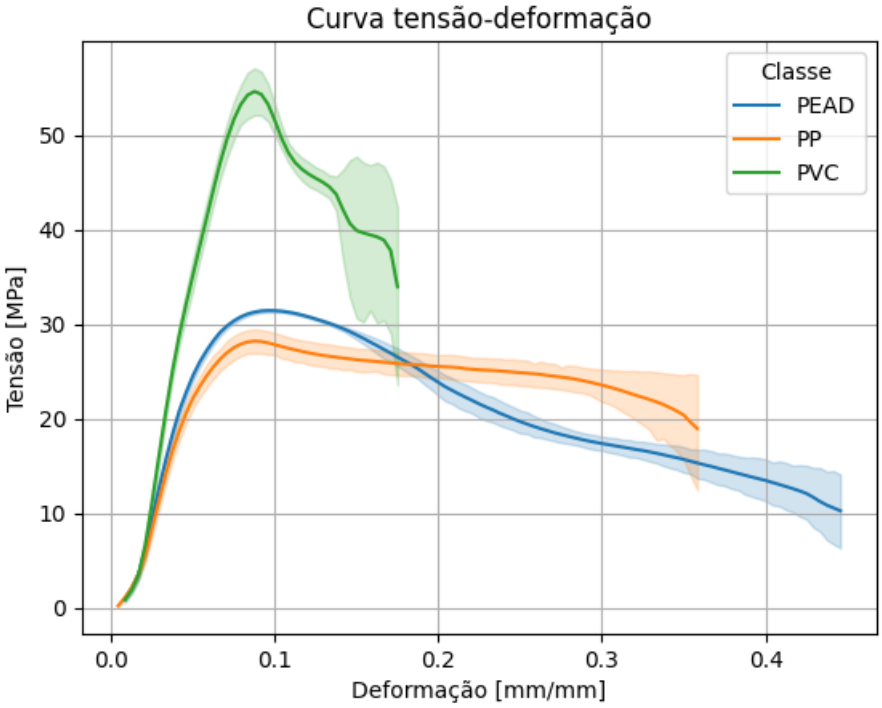


(a)

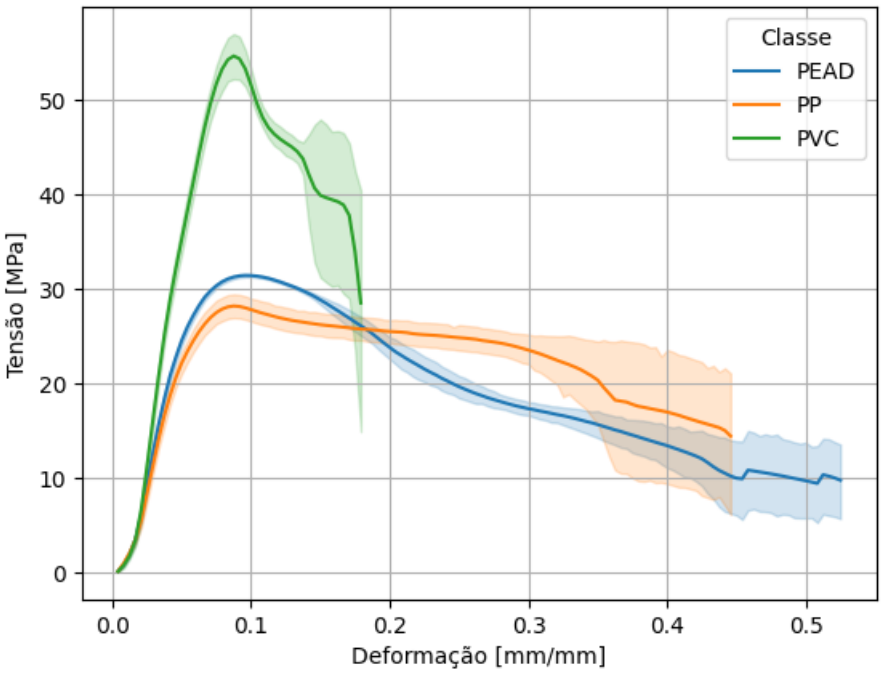


(b)

Figura 6.4: (a) Curvas tensão-deformação após a retirada dos dados de tensão com coeficientes de variação maiores que 0,30. (b) Curvas tensão-deformação após a retirada dos dados de tensão com coeficientes de variação maiores que 0,50.

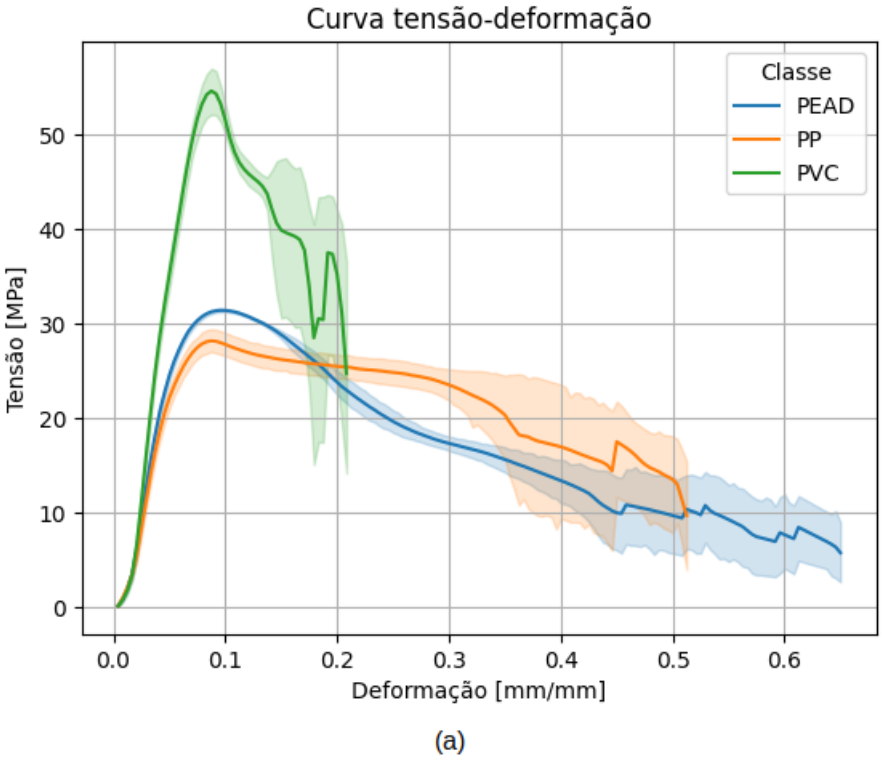


(a)

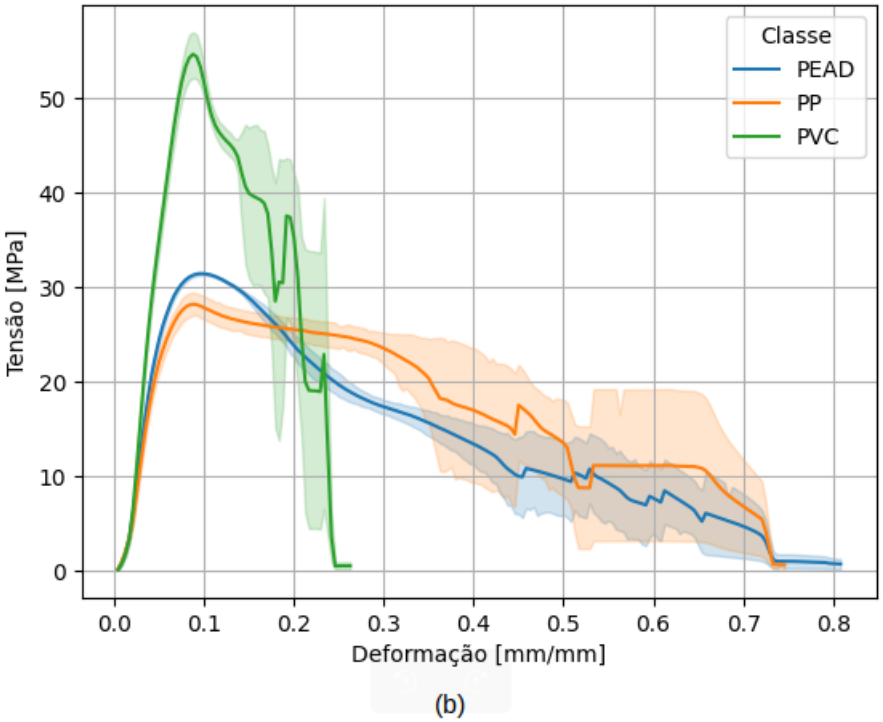


(b)

Figura 6.5: (a) Curvas tensão-deformação após a retirada dos dados de tensão com coeficientes de variação maiores que 0,70. (b) Curvas tensão-deformação após a retirada dos dados de tensão com coeficientes de variação maiores que 1,00.



(a)



(b)

Figura 6.6: (a) Curvas tensão-deformação após a retirada dos dados de tensão com coeficientes de variação maiores que 2,00. (b) Curvas tensão-deformação após a retirada dos dados de tensão com coeficientes de variação maiores que 3,00.

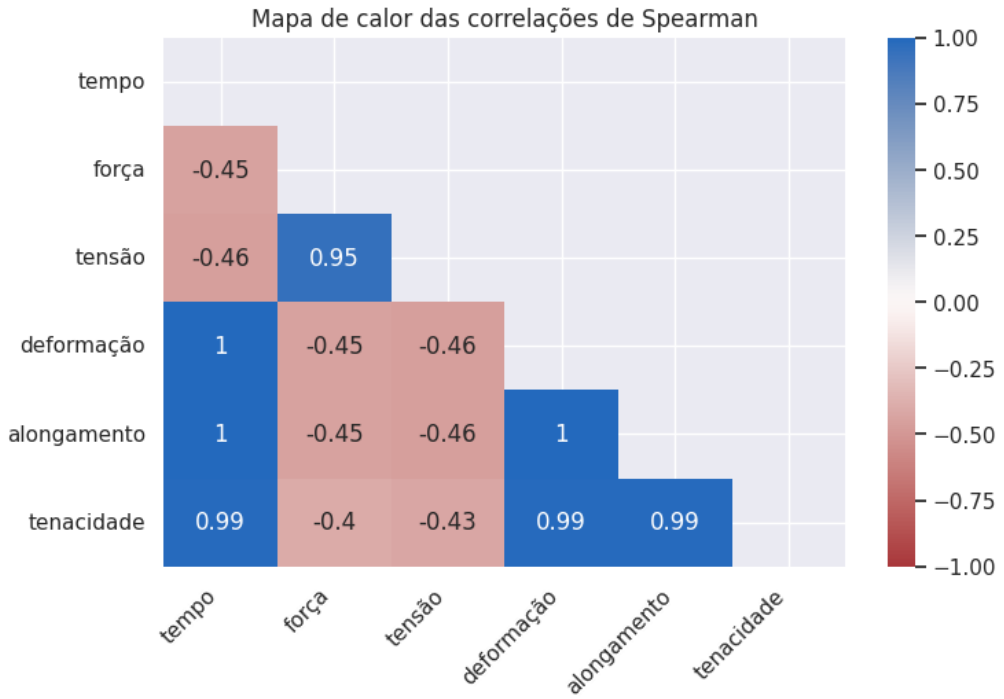


Figura 6.7: Mapa de calor das correlações de Spearman.

Portanto, os atributos selecionados para o treinamento do modelo de classificação foram a tensão, a deformação e a tenacidade.

6.3 Modelo de classificação

A Tabela 6.2 apresenta as acurácias de 15 modelos com diferentes números de neurônios na única camada escondida. Nesta tabela é possível verificar que uma arquitetura com 12 neurônios na camada escondida obteve o melhor desempenho para a acurácia no conjunto de dados de validação, porém a arquitetura com 15 neurônios na camada escondida teve uma acurácia 0,0001 menor que o modelo com 12 neurônios e um desvio padrão 4 vezes menor para a validação, o que confere consistência ao modelo. Ainda que o modelo com 15 neurônios em uma única camada escondida tenha tido um bom desempenho, foram avaliados modelos com duas camadas escondidas.

Tabela 6.2: Ajuste da quantidade de unidades para uma arquitetura com uma camada escondida.

NU[a]	MAT[b]	DPAT[c]	MAV[d]	DPAV[e]
12	0.8854	0.0368	0.8837	0.0244
15	0.8907	0.0090	0.8836	0.0056
16	0.8901	0.0061	0.8828	0.0059
20	0.8906	0.0184	0.8818	0.0130
18	0.8877	0.0151	0.8810	0.0123
25	0.8862	0.0104	0.8783	0.0108
17	0.8871	0.0208	0.8775	0.0222
8	0.8854	0.0156	0.8739	0.0243
11	0.8822	0.0081	0.8731	0.0109
19	0.8824	0.0207	0.8731	0.0290
14	0.8777	0.0175	0.8730	0.0165
13	0.8766	0.0141	0.8713	0.0216
9	0.8739	0.0100	0.8695	0.0089
10	0.8610	0.0368	0.8598	0.0342
7	0.8430	0.0207	0.8272	0.0467

[a] Número de unidades na camada escondida.
[b] Média das acurácias no treinamento.
[c] Desvio padrão das acurácias no treinamento.
[d] Média das acurácias na validação.
[e] Desvio padrão das acurácias na validação.

A Tabela 6.3 apresenta as acurácias de 27 modelos com diferentes combinações de neurônios nas duas camadas escondidas. Nesta tabela é possível verificar que uma arquitetura com 22 neurônios nas duas camadas escondidas obteve o melhor desempenho para a acurácia no conjunto de dados de validação, porém a arquitetura com 9 neurônios nas duas camadas escondidas teve uma acurácia 0,0035 menor que o modelo com 22 neurônios e um desvio padrão 10 vezes menor para a validação. Os modelos com 9 neurônios nas duas camadas escondidas e com 17 neurônios nas duas camadas escondidas também tiveram desempenhos satisfatórios, com acurácias próximas a 90 % e desvios padrão menores que 1 %.

Tabela 6.3: Ajuste da quantidade de unidades para uma arquitetura com duas camadas escondidas.

NU1[a]	NU2[b]	MAT[c]	DPAT[d]	MAV[e]	DPAV[f]
22	22	0.9110	0.0140	0.9100	0.0202
9	9	0.9071	0.0091	0.9065	0.0021
24	24	0.9204	0.0124	0.9048	0.0165
18	18	0.9165	0.0178	0.9030	0.0158
15	15	0.9071	0.0187	0.9013	0.0174
25	25	0.9130	0.0212	0.9012	0.0229
20	20	0.9068	0.0099	0.9004	0.0143
21	21	0.9127	0.0166	0.9004	0.0136
17	17	0.9106	0.0056	0.8995	0.0066
11	11	0.9136	0.0121	0.8977	0.0148
19	19	0.9086	0.0139	0.8977	0.0051
15	14	0.9112	0.0126	0.8968	0.0061
12	12	0.9074	0.0186	0.8960	0.0184
14	15	0.9063	0.0122	0.8960	0.0092
16	16	0.9101	0.0141	0.8942	0.0126
15	16	0.9039	0.0149	0.8942	0.0103
16	15	0.9025	0.0176	0.8916	0.0100
8	8	0.8992	0.0123	0.8915	0.0156
14	14	0.8995	0.0266	0.8889	0.0221
10	10	0.9042	0.0173	0.8871	0.0144
13	13	0.9004	0.0101	0.8863	0.0194
23	23	0.9051	0.0156	0.8845	0.0242
7	7	0.8930	0.0218	0.8730	0.0238
5	5	0.8568	0.0221	0.8563	0.0261
13	14	0.8568	0.0221	0.8563	0.0261
6	6	0.8469	0.0534	0.8369	0.0776
14	13	0.8469	0.0534	0.8369	0.0776

[a] Número de unidades na primeira camada escondida.
[b] Número de unidades na segunda camada escondida.
[c] Média das acurácias no treinamento.
[d] Desvio padrão das acurácias no treinamento.
[e] Média das acurácias na validação.
[f] Desvio padrão das acurácias na validação.

Os modelos das Tabelas 6.2 e 6.3, foram treinados utilizando parada antecipada com o parâmetro de paciência igual a 2. Desta forma, como pode ser verificado na Figura 6.8, as curvas de aprendizagem tiveram um ajuste satisfatório entre as curvas de treino e validação para o modelo com 9 neurônios nas duas camadas escondidas. Contudo, quanto menor é o valor atribuído à paciência, menores são as épocas no treinamento, o que pode levar a uma menor acurácia.

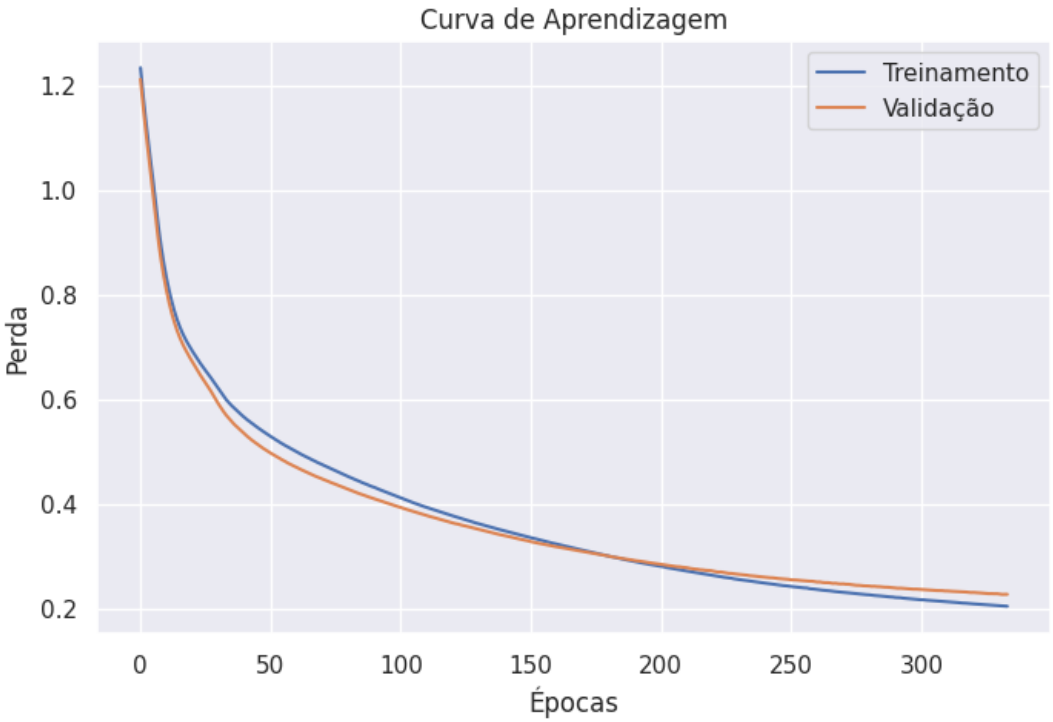


Figura 6.8: Curva de aprendizagem para um modelo com 15 neurônios nas duas camadas escondidas e paciência igual a 2.

A Tabela 6.4 apresenta modelos treinados utilizando parada antecipada com paciência igual a 5 e diferentes combinações de neurônios nas duas camadas escondidas. Nesta tabela é possível verificar que uma arquitetura com 16 neurônios nas duas camadas escondidas obteve o melhor desempenho para a acurácia no conjunto de dados de validação, porém a arquitetura com 17 neurônios nas duas camadas escondidas teve uma acurácia 0,0010 menor que o modelo com 16 neurônios e um desvio padrão cerca de 3 vezes menor para a validação, além de ter os menores desvios padrão comparado com todos os outros modelos desta tabela. Apesar do aumento da paciência, a curvas de aprendizagem para o modelo com 17 neurônios (Figura 6.9) estão praticamente sobrepostas e sem ruídos, indicando que não ocorreram sobreajustes no modelo.

Tabela 6.4: Ajuste da quantidade de unidades para uma arquitetura com duas camadas escondidas e paciência igual a 5.

NU1[a]	NU2[b]	MAT[c]	DPAT[d]	MAV[e]	DPAV[f]
16	16	0.9283	0.0177	0.9198	0.0087
17	17	0.9295	0.0059	0.9189	0.0029
18	18	0.9271	0.0067	0.9180	0.0108
15	15	0.9259	0.0082	0.9180	0.0107
25	25	0.9324	0.0098	0.9171	0.0072
20	20	0.9239	0.0134	0.9092	0.0093

[a] Número de unidades na primeira camada escondida.
[b] Número de unidades na segunda camada escondida.
[c] Média das acurácias no treinamento.
[d] Desvio padrão das acurácias no treinamento.
[e] Média das acurácias na validação.
[f] Desvio padrão das acurácias na validação.

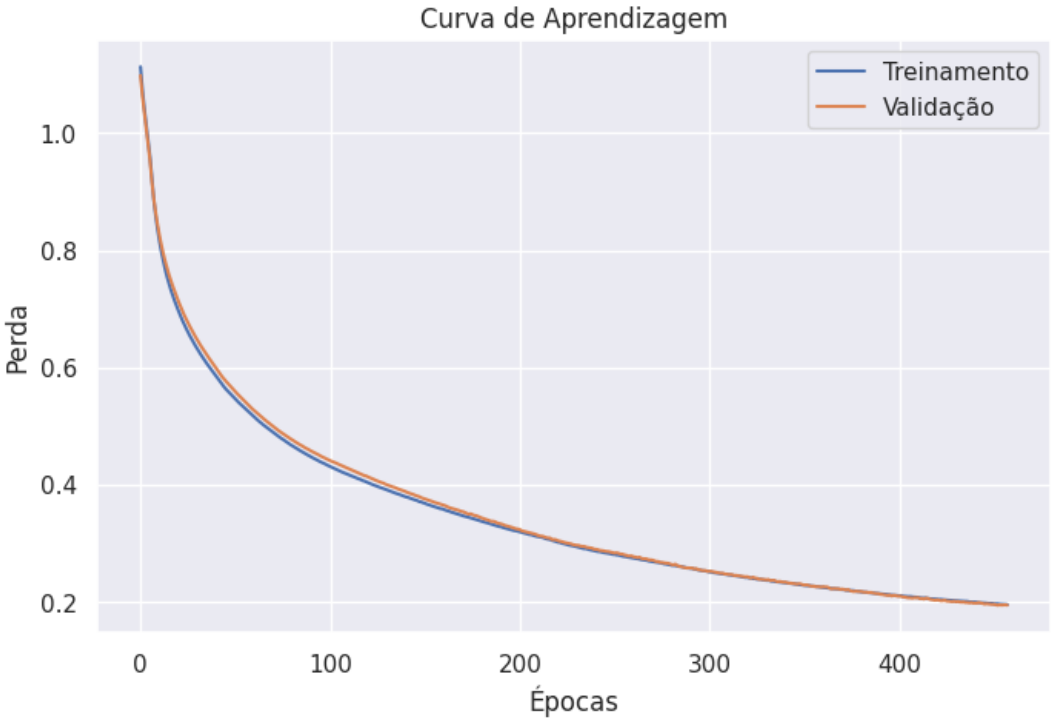


Figura 6.9: Curva de aprendizagem para um modelo com 17 neurônios nas duas camadas escondidas e paciência igual a 5.

Devido a acurácia verificada, os baixos desvios padrão e o comportamento da curva de aprendizagem, o modelo com 17 neurônios e duas camadas escondidas foi selecionado para verificar a sua acurácia no conjunto de dados de validação para diferentes limites de coeficiente de variação estabelecidos no pré-processamento dos dados (Tabela 6.5).

Tabela 6.5: Ajuste dos limites estabelecidos no pré-processamento dos dados para o coeficiente de variação entre as tensões. Foi utilizada uma arquitetura com 17 neurônios nas duas camadas escondidas e paciência igual a 5.

LCV[a]	MAT[b]	DPAT[c]	MAV[d]	DPAV[e]
0.3	0.9295	0.0059	0.9189	0.0029
0.7	0.9213	0.0038	0.9149	0.0110
0.5	0.9223	0.0074	0.9085	0.0187
1.0	0.9106	0.0089	0.9081	0.0218
1.5	0.9050	0.0141	0.8953	0.0243
2.0	0.8933	0.0124	0.8886	0.0223
2.5	0.8717	0.0247	0.8643	0.0167
3.0	0.8760	0.0213	0.8588	0.0266

[a] Limite estabelecido para o coeficiente de variação entre as tensões.
[b] Média das acurácias no treinamento.
[c] Desvio padrão das acurácias no treinamento.
[d] Média das acurácias na validação.
[e] Desvio padrão das acurácias na validação.

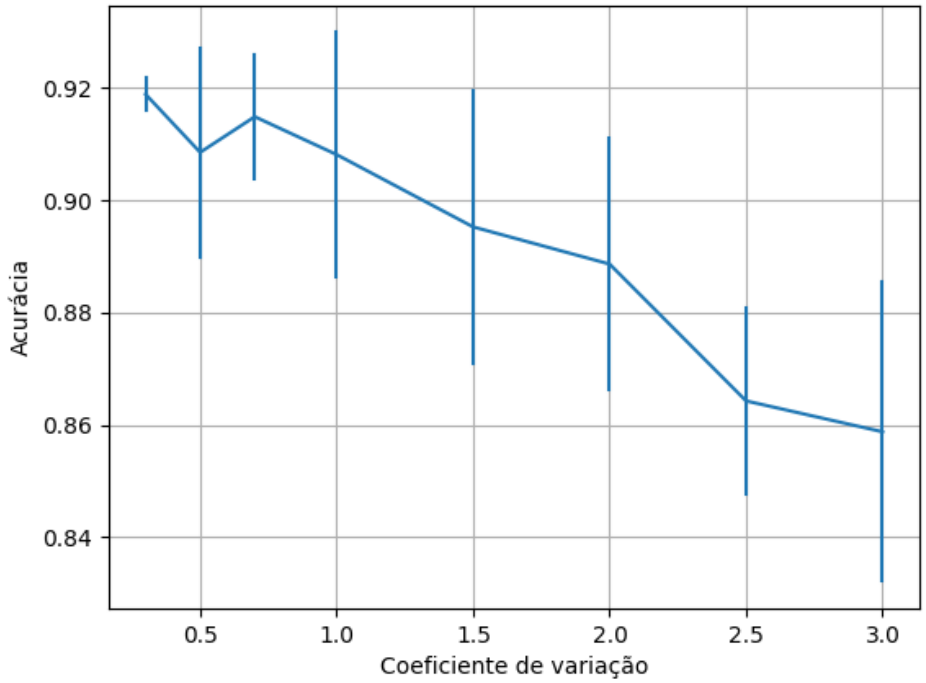


Figura 6.10: Acurácia na validação do modelo com 17 neurônios e duas camadas escondidas para diferentes limites de coeficiente de variação estabelecidos no pré-processamento dos dados.

As médias das acurácias para o treinamento e validação na Tabela 6.5 são maiores para o limite estabelecido de 0,3 para o coeficiente de variação entre as tensões. O modelo tem o menor desvio padrão da acurácia para validação e o segundo menor desvio padrão para a acurácia no treinamento, e portanto, foi este o modelo selecionado para a classificação dos polímeros.

Na Figura 6.10, é possível observar que existe uma tendência de queda da acurácia conforme é aumentado o limite estabelecido para o coeficiente de variação entre as tensões, exceto para o limite de 0,7. O comportamento para os limites de 0,5 e 0,7 pode não ter sido o esperado devido ao aumento de dados conforme é aumentado o limite para o coeficiente de variação, conforme pode ser observado nas Figuras 6.4(b) e 6.5(a). A quantidade de dados existente é uma limitação para o desempenho do modelo e o seu aumento pode compensar a perda de desempenho pela maior difusão dos dados em um limite maior para o coeficiente de variação entre as tensões.

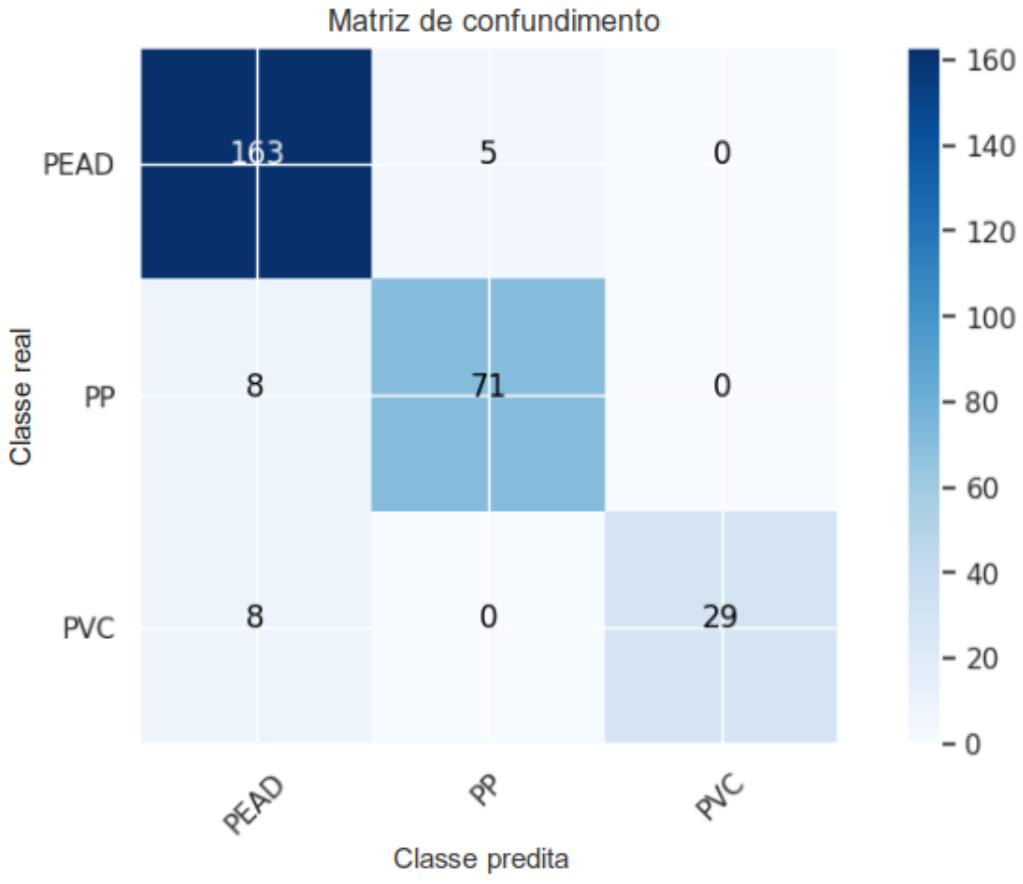


Figura 6.11: Matriz de confundimento para os dados de teste.

A Figura 6.11 apresenta a matriz de confundimento para os dados de teste, este gráfico possibilita a visualização da quantidade de acertos das predições do modelo para cada classe, *i. e.*, a precisão do modelo. Por exemplo,

é possível observar que dos 29 dados preditos como PVC, todos eram de fato PVC, visto que no eixo de classes verdadeiras, a classe PP e PEAD são iguais a 0. Isto se confirma ao verificar a precisão do modelo igual a 1 para o PVC na Tabela 6.6. Por meio da matriz de confundimento também é possível observar a quantidade de predições corretas em relação ao total de dados verdadeiros de uma classe, *i. e.*, a revocação do modelo. Por exemplo, é possível observar que dos 37 dados verdadeiros da classe PVC, 29 foram de fato preditos como PVC, isto se confirma ao verificar a revocação do modelo igual a 0,78 para o PVC na Tabela 6.6. As revocações verificadas para as classes PP e PVC foram menores quando comparadas a revocação obtida para a classe PEAD, isto provavelmente ocorreu por haver menos dados para estas classes. É possível verificar também que a revocação e a quantidade de dados para a classe PVC foram menores que a revocação e a quantidade de dados para a classe PP, demonstrando portanto a influência do desbalanceamento dos dados sobre as revocações encontradas.

A acurácia do modelo foi de 92,61 %, porém com a revocação de 78 % e precisão de 100 % para a classe PVC, é provável que caso haja o balanceamento dos dados antes do treinamento, ocorra o aumento da revocação e redução da precisão para a classe PVC, e por conseguinte a redução da acurácia do modelo, porém possibilitando uma capacidade de classificação mais uniforme para todas as classes.

Tabela 6.6: Relatório de classificação para os dados de teste.

Classe	Precisão	Revocação	Pontuação-F1	Amostras
PEAD	0.91061	0.97024	0.93948	168
PP	0.93421	0.89873	0.91613	79
PVC	1,00000	0.78378	0.87879	37

A definição do `random_state` no pré-processamento dos dados, apesar de garantir as mesmas distribuições nos conjuntos de dados para testes das configurações de arquitetura e treino dos modelos, também pode ter causado um viés no resultado do modelo selecionado e, portanto, é interessante a realização do treinamento do modelo selecionado sem a definição do `random_state`, garantindo assim a aleatoriedade da distribuição dos dados na divisão do conjunto de dados entre treino, validação e teste.

O treinamento do modelo com as configurações de arquitetura e treino selecionadas foi realizado em cerca de 45 segundos, com a utilização de até 30 % da CPU por *thread*, em uma máquina com as seguintes especificações:

- Modelo: Dell G15 5515;
- Processador: AMD® Ryzen 7 5800H (8 núcleos, 16 *threads*);
- Memória RAM: 16 GB;
- Placa gráfica: NVIDIA GeForce RTX 3060 (6GB, GDDR6);
- Sistema operacional: Ubuntu 22.04.4 LTS.

7. Conclusão

Com uma acurácia de 92,61 % para o conjunto de testes, o modelo obtido da rede neural PMC com 17 neurônios nas duas camadas escondidas foi capaz de classificar os polímeros PEAD, PP e PVC com base nos dados das curvas tensão-deformação com um limite estabelecido de 0,3 para o coeficiente de variação entre as tensões de uma dada deformação, indicando assim a possibilidade de utilização de um algoritmo ML para automatizar a classificação de materiais poliméricos com base em dados de ensaios de tração. Além disto, foi possível observar a influência do pré-processamento dos dados no resultado do modelo. Ao aumentar o limite para o coeficiente de variação no pré-processamento dos dados, houve a diminuição na acurácia para a maior parte dos limites estudados. Também foi possível aumentar a acurácia dos modelos relaxando a restrição de parada antecipada sem que houvesse a ocorrência de sobreajustes no modelo.

7.1 Trabalhos futuros

De uma forma geral, os modelos de inteligência artificial têm melhores desempenhos para a interpolação se comparados a extrapolação [35]. Portanto, para verificar a possibilidade de classificar misturas poliméricas ou um material plástico reciclado quanto ao teor de um determinado polímero em sua composição, são necessários treinamentos de modelos de classificação a partir de redes PMC utilizando conjuntos de dados obtidos a partir de ensaios de tração nestes materiais.

É necessária a implementação de técnicas de balanceamento dos dados em novos treinamentos à fim de evitar a baixa revocação obtida para o PVC. Uma das técnicas que poderia ser utilizada é a técnica de sobreamostragem minoritária sintética, conhecida como SMOTE, sigla para o nome da técnica em inglês (*Synthetic Minority Over-sampling Technique*). Esta técnica permite criar dados intermediários no conjunto de dados, por exemplo, em um caso hipotético, se no conjunto de dados para a classe PVC há uma tensão associada a uma deformação de 0,8 mm/mm, e uma outra tensão associada a uma deformação de 1,0 mm/mm, a técnica SMOTE poderia criar um dado de tensão associado a uma deformação de 0,9 mm/mm para a classe PVC, e o valor desta nova tensão seguirá a mesma tendência de variação dos dados de tensão em torno da deformação de 0,9 mm/mm.

Referências bibliográficas

[1] RIGAMONTI, L. et al. Environmental evaluation of plastic waste management scenarios. v. 85, p. 42–53. ISSN 09213449. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S0921344913002784>.

[2] BABAREMU, K. et al. Sustainable plastic waste management in a circular economy. v. 8, n. 7, p. e09984. ISSN 24058440. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S2405844022012725>.

[3] LEBRETON, L.; ANDRADY, A. Future scenarios of global plastic waste generation and disposal. Palgrave, v. 5, n. 1, p. 1–11. ISSN 2055-1045. Disponível em: <https://www.nature.com/articles/s41599-018-0212-7>.

[4] JAMBECK, J. R. et al. Plastic waste inputs from land into the ocean. American Association for the Advancement of Science, v. 347, n. 6223, p. 768–771, 2015. Disponível em: <https://www.science.org/doi/10.1126/science.1260352>.

[5] PAGLIARO, M. Waste-to-wealth: The economic reasons for replacing waste-to-energy with the circular economy of municipal solid waste. p. 59–65. ISSN 2384-8677. Disponível em: <https://ojs.unito.it/index.php/visions/article/view/4421>.

[6] HURTADO, L. Ángeles et al. Viable Disposal of Post-Consumer Polymers in Mexico: A Review. v. 9. ISSN 2296-665X.

[7] TURKU, I. et al. Characterization of plastic blends made from mixed plastics waste of different sources. SAGE Publications Ltd STM, v. 35, n. 2, p. 200–206, 2017. ISSN 0734-242X. Disponível em: <https://doi.org/10.1177/0734242X16678066>.

[8] CANEVAROLO, S. V. Chapter 9 - Polymer Mechanical Behavior. In: CANEVAROLO, S. V. (Ed.). *Polymer Science*. [S.l.]: Hanser, 2020. p. 237–279. ISBN 978-1-56990-725-2.

[9] BILLMEYER, F. W. Chapter 9 - Analysis and Testing of Polymers. In: *Textbook of Polymer Science*. [S.l.]: Wiley. p. 229–257. ISBN 978-0-471-03196-3.

[10] RUSSELL, S.; NORVIG, P. *Artificial Intelligence: A Modern Approach*. Pearson, 2016. (Always Learning). ISBN 978-1-292-15396-4. Disponível em: <https://books.google.com.br/books?id=XS9CjwEACAAJ>.

[11] HAYKIN, S. *Redes Neurais: Princípios e Prática*. Bookman Editora, 2021. ISBN 978-85-7780-086-5. Disponível em: <https://books.google.com.br/books?id=bhMwDwAAQBAJ>.

[12] RAZA, M.; JAYASINGHE, N. D.; MUSLAM, M. M. A. A Comprehensive Review on Email Spam Classification using Machine Learning Algorithms. In: *2021 International Conference on Information Networking (ICOIN)*. [S.l.: s.n.]. p. 327–332. ISSN 1976-7684.

[13] WALCZAK, S. Artificial Neural Networks. In: *Advanced Methodologies and Technologies in Artificial Intelligence, Computer Simulation, and Human-Computer Interaction*. IGI Global. p. 40–53. ISBN 978-1-5225-7368-5. Disponível em: <https://www.igi-global.com/chapter/artificial-neural-networks/www.igi-global.com/chapter/artificial-neural-networks/213116>.

[14] FERREIRA, M. H.; FAMILY=MOARES GIVEN=MINéIA APARECIDA, p. u. Redes neurais artificiais: PrincÍpios bÁsicos. v. 1, n. 13.

[15] AGUIAR, D. et al. Perceptrões e redes neuronais artificiais. v. 10, n. 1, 2022. ISSN 2183-1270. Disponível em: <http://rce.casadasciencias.org/art/2022/004>.

[16] GéRON, A. *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media. ISBN 978-1-09-812246-1. Disponível em: <https://books.google.com.br/books?id=X5ySEAAAQBAJ>.

[17] VIEIRA, F. M. S. Álgebra booleana. v. 5, n. 1, 2000. ISSN 2317-7756. Disponível em: <https://www.periodicos.cefetmg.br/index.php/revista-et/article/view/341>.

[18] VAHID, F. *Sistemas Digitais*. Bookman. ISBN 978-85-7780-237-1. Disponível em: <https://books.google.com.br/books?id=8xT9sD0kpfUC>.

[19] DAVIS, J. *Tensile Testing, 2nd Edition*. ASM International, 2004. ISBN 978-1-61503-095-8. Disponível em: <https://books.google.com.br/books?id=5uRIb3emLY8C>.

[20] BAJRACHARYA, R. M. et al. Characterisation of recycled mixed plastic solid wastes: Coupon and full-scale investigation. v. 48, p. 72–80, 2016. ISSN 0956053X. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S0956053X15302014>.

[21] ZHU, L.; ZHOU, J.; SUN, Z. Materials Data toward Machine Learning: Advances and Challenges. American Chemical Society, v. 13, n. 18, p. 3965–3977, 2022. Disponível em: <https://doi.org/10.1021/acs.jpclett.2c00576>.

[22] ALTARAZI, S.; ALLAF, R.; ALHINDAWI, F. Machine Learning Models for Predicting and Classifying the Tensile Strength of Polymeric Films Fabricated via Different Production Processes. Multidisciplinary Digital Publishing Institute, v. 12, n. 9, p. 1475, 2019. ISSN 1996-1944. Disponível em: <https://www.mdpi.com/1996-1944/12/9/1475>.

[23] NIKAM, S. S. A comparative study of classification techniques in data mining algorithms. v. 8, n. 1, p. 13–19, 2015. Disponível em: <https://www.computerscijournal.org/vol8no1/a-comparative-study-of-classification-techniques-in-data-mining-algorithms/>.

[24] LI, Y. Predicting materials properties and behavior using classification and regression trees. v. 433, p. 261–268, 2006.

[25] SINGH, S.; GIRI, M. Comparative study id3, cart and c4.5 decision tree algorithm: A survey. 2014.

[26] WU, D. et al. A comparative study on machine learning algorithms for smart manufacturing: Tool wear prediction using random forests. v. 139, n. 71018, 2017. ISSN 1087-1357. Disponível em: <https://doi.org/10.1115/1.4036350>.

[27] HAJI, S. H.; ABDULAZEEZ, A. M. COMPARISON OF OPTIMIZATION TECHNIQUES BASED ON GRADIENT DESCENT ALGORITHM: A REVIEW. v. 18, n. 4, p. 2715–2743, 2021. ISSN 1567-214X. Number: 4. Disponível em: <https://archives.palarch.nl/index.php/jae/article/view/6705>.

[28] COOK, J.; RAMADAS, V. When to consult precision-recall curves. v. 20, n. 1, p. 131–148, 2020. ISSN 1536-867X. Publisher: SAGE Publications. Disponível em: <https://doi.org/10.1177/1536867X20909693>.

[29] AMOR, N.; NOMAN, M. T.; PETRU, M. Classification of Textile Polymer Composites: Recent Trends and Challenges. Multidisciplinary Digital Publishing Institute, v. 13, n. 16, p. 2592, 2021. ISSN 2073-4360. Disponível em: <https://www.mdpi.com/2073-4360/13/16/2592>.

[30] YOUSEF, B. F.; MOURAD, A.-H. I.; HILAL-ALNAQBI, A. Prediction of the mechanical properties of PE/PP blends using artificial neural networks. v. 10, p. 2713–2718, 2011. ISSN 1877-7058. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1877705811006400>.

[31] NETO, J. G. et al. Framework for data-driven polymer characterization from infrared spectra. v. 300, p. 122841. ISSN 13861425. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S1386142523005267>.

[32] KINGMA, D. P.; BA, J. *Adam: A Method for Stochastic Optimization*. arXiv, 2017. ArXiv:1412.6980 [cs]. Disponível em: <http://arxiv.org/abs/1412.6980>.

[33] RIBEIRO, A. M. Um Estudo Comparativo Entre Cinco Métodos de Otimização Aplicados Em Uma RNC Voltada ao Diagnóstico do Glaucoma. v. 10, n. 1, 2020.

[34] RUDER, S. *An overview of gradient descent optimization algorithms*. arXiv, 2017. ArXiv:1609.04747 [cs]. Disponível em: <http://arxiv.org/abs/1609.04747>.

[35] MUCKLEY, E. S. et al. Interpretable models for extrapolation in scientific machine learning. *Digital Discovery*, v. 2, n. 5, p. 1425–1435, 2023. ISSN 2635-098X. Disponível em: <https://xlink.rsc.org/?DOI=D3DD00082F>.