



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CAMPUS DE ARARANGUÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM TECNOLOGIA DA INFORMAÇÃO E COMUNICAÇÃO

Iury Melo Américo

**Predição de Curvas de Escoamento de Aço Utilizando Algoritmos de
Inteligência Artificial**

Araranguá/SC

2026

PREDIÇÃO DE CURVAS DE ESCOAMENTO DE AÇO UTILIZANDO ALGORITMOS DE
INTELIGÊNCIA ARTIFICIAL

Iury Melo Américo

Dissertação de Mestrado submetida ao Programa de Pós-Graduação em Tecnologia da Informação e Comunicação (PPGTIC), da Universidade Federal de Santa Catarina para a obtenção do título de Mestre.

Orientador: Prof. Roderval Marcelino, Dr.

Araranguá/SC

2026

Ficha catalográfica gerada por meio de sistema automatizado gerenciado pela BU/UFSC.
Dados inseridos pelo próprio autor.

Melo Américo, Iury
Predição de Curvas de Escoamento de Aço Utilizando
Algoritmos de Inteligência Artificial / Iury Melo Américo
; orientador, Roderval Marcelino, 2026.
80 p.

Dissertação (mestrado) - Universidade Federal de Santa
Catarina, Campus Araranguá, Programa de Pós-Graduação em
Tecnologias da Informação e Comunicação, Araranguá, 2026.

Inclui referências.

1. Tecnologias da Informação e Comunicação. 2.
Inteligência Artificial. 3. Aprendizado de Máquina. 4.
Curva de Escoamento. 5. Conformação Mecânica. I. Marcelino,
Roderval. II. Universidade Federal de Santa Catarina.
Programa de Pós-Graduação em Tecnologias da Informação e
Comunicação. III. Título.

Iury Melo Américo

Predição de Curvas de Escoamento de Aço Utilizando Algoritmos de Inteligência Artificial

O presente trabalho em nível de mestrado foi avaliado e aprovado por banca examinadora composta pelos seguintes membros:

Profa. Eliane Pozzebon, Dra.
Universidade Federal de Santa Catarina

Prof. Antonio Carlos Sobieranski, Dr.
Universidade Federal de Santa Catarina

Prof. André Rosiak, Dr.
Universidade Federal do Rio Grande do Sul

Certificamos que esta é a **versão original e final** do trabalho de conclusão que foi julgado adequado para obtenção do título de mestre em Ensino de Física



Documento assinado digitalmente
Giovani Mendonça Lunardi
Data: 11/02/2026 14:46:31-0300
CPF: ***.394.559-**
Verifique as assinaturas em <https://v.ufsc.br>

Coordenação do Programa de Pós-Graduação



Documento assinado digitalmente
Roderval Marcelino
Data: 11/02/2026 14:29:37-0300
CPF: ***.867.489-**
Verifique as assinaturas em <https://v.ufsc.br>

Prof.(a) Prof. Roderval Marcelino, Dr., Dr.(a)
Orientador

Araranguá, 2026.

Esse trabalho é dedicado a todos aqueles que, de forma direta ou indireta, auxiliaram a sua conclusão.

Agradecimentos

Muito obrigado ao Laboratório CPS Lab - *Cyber-Physical Systems* da Universidade Federal de Santa Catarina e ao seu Professor responsável, e também orientador, Dr. Roderval Marcelino, por todo apoio e paciência no desenvolvimento deste trabalho. O auxílio tanto técnico quanto pessoal foi crucial para que conseguíssemos chegar no resultado que chegamos. Um forte agradecimento também ao Professor Dr. Lirio Schaeffer da Universidade Federal do Rio Grande do Sul pela ajuda para conseguir os dados e com a expertisse em conformação mecânica que também foram pontos críticos para o entendimento do tema.

Os lábios do conhecimento estão fechados, exceto
para os ouvidos do entendimento.

The Kybalion

Resumo

Este trabalho avalia o desempenho de diferentes algoritmos de aprendizado de máquina para predição de curvas de escoamento de metais utilizando a composição química, temperatura, deformação e velocidade de deformação como variáveis de entrada. O objetivo é proporcionar uma alternativa à experimentação laboratorial, reduzindo custos com a geração dessas curvas, essenciais na análise do processo industrial de conformação mecânica. A pesquisa classifica-se como tecnológica aplicada, empregando a implementação de um protótipo computacional dividido em duas etapas: seleção e avaliação de modelos de regressão, e o desenvolvimento de uma ferramenta web para uso por engenheiros. Foram utilizados métodos de validação cruzada, limpezas e padronizações no dataset extraído da biblioteca do software de elementos finitos QForm UK, além de otimização de hiperparâmetros empregando bibliotecas como scikit-learn, Tensorflow e Optuna. Os principais modelos avaliados foram Random Forest, XGBoost, GradientBoosting, SVR e redes neurais rasas e profundas. O modelo XGBoost apresentou o melhor desempenho, superando inclusive redes neurais profundas, com erro absoluto médio de 7,53 no conjunto de validação, demonstrando robustez mesmo diante de materiais não presentes no treinamento. Constatou-se que tanto temperatura quanto velocidade de deformação podem influenciar o desempenho, especialmente em aços de alta liga, levantando potenciais temas para estudos futuros. Ao final, foi desenvolvida uma API e aplicação web para facilitar a aplicação prática dos modelos em ambiente industrial. Os resultados confirmam que técnicas de aprendizado de máquina são capazes de auxiliar na predição confiável de curvas de escoamento, atendendo aos objetivos propostos.

Palavras-chave: aprendizado de máquina. inteligência artificial. curva de escoamento. conformação mecânica.

Abstract

This work evaluates the performance of different machine learning algorithms for predicting metal strain curves using chemical composition, temperature, strain, and strain rate as input variables. The goal is to provide an alternative to laboratory experimentation, reducing the costs associated with generating these curves, which are essential for analyzing industrial metal forming processes. The research is classified as applied technological, involving the implementation of a computational prototype divided into two stages: the selection and evaluation of regression models, and the development of a web tool for engineers. Cross-validation methods, data cleaning, and standardization were applied to the dataset extracted from QForm simulations, along with hyperparameter optimization using libraries such as scikit-learn, TensorFlow, and Optuna. The main models evaluated were Random Forest, XGBoost, Gradient Boosting, SVR, and shallow and deep neural networks. The XGBoost model achieved the best performance, outperforming even deep neural networks, with a mean absolute error of 7.53 on the validation set, demonstrating robustness even for materials not present in training. It was found that both temperature and strain rate can influence performance, particularly in high-alloy steels, suggesting potential directions for future research. Finally, an API and web application were developed to facilitate the practical application of the models in industrial environments. The results confirm that machine learning techniques can reliably assist in predicting strain curves, meeting the proposed objectives.

Keywords: machine learning. artificial intelligence. flow curve. mechanical forming.

Lista de ilustrações

Figura 1 – Exemplo de uma árvore de decisão para regressão.	24
Figura 2 – Exemplo de uma rede neural. Escrito em verde temos a camada de entrada, em preto as camadas ocultas e em azul a camada de saída. . .	26
Figura 3 – Exemplo das etapas para fabricação de um cárter de óleo	28
Figura 4 – Curvas de escoamento do aço 22MnB5 para diferentes temperaturas e velocidades de deformação obtidas a partir do ensaio de tração para as temperaturas de: (a) 700°C, (b) 800°C	29
Figura 5 – Número de artigos que foram publicados utilizando os critérios de busca definidos neste trabalho	32
Figura 6 – Diagrama de experimentação adotada.	41
Figura 7 – Exemplo do <i>dataframe</i> com os dados de treino e teste.	42
Figura 8 – Comparação do erro médio absoluto entre os modelos. Em verde, o melhor modelo.	53
Figura 9 – Comparação da raiz do erro médio quadrático e do R^2 entre os modelos. Em verde, o melhor modelo.	53
Figura 10 – Distribuição e Diagrama de Caixa do erro médio absoluto para o modelo de XGBoost.	54
Figura 11 – Distribuição e Diagrama de Caixa do erro médio absoluto para o modelo de GradientBoosting.	54
Figura 12 – Distribuição e Diagrama de Caixa do erro médio absoluto para o modelo de MLP do scikit-learn.	55
Figura 13 – Distribuição e Diagrama de Caixa do erro médio absoluto para o modelo de <i>Random Forest</i>	55
Figura 14 – Distribuição e Diagrama de Caixa do erro médio absoluto para o modelo de Árvore de Decisão.	55
Figura 15 – Distribuição e Diagrama de Caixa do erro médio absoluto para o modelo de SVR.	56
Figura 16 – Distribuição do RMSE dos modelos.	56
Figura 17 – Distribuição do MAPE dos modelos.	57
Figura 18 – Diagrama de barra mostrando a média do erro absoluto com o desvio padrão para o modelo XGBoost.	58
Figura 19 – Diagrama de barra mostrando a média do erro absoluto com o desvio padrão para o modelo <i>Gradient Boosting</i>	58
Figura 20 – Diagrama de barra mostrando a média do erro absoluto com o desvio padrão para o modelo MLP do scikit-learn.	59

Figura 21 – Diagrama de barra mostrando a média do erro absoluto com o desvio padrão para o modelo de <i>Random Forest</i>	59
Figura 22 – Diagrama de barra mostrando a média do erro absoluto com o desvio padrão para o modelo de Árvore de Decisão.	60
Figura 23 – Diagrama de barra mostrando a média do erro absoluto com o desvio padrão para o modelo SVR.	60
Figura 24 – Diagrama de barra mostrando a média do erro absoluto com o desvio padrão para o modelo XGBoost para a velocidade de deformação. . . .	61
Figura 25 – Diagrama de barra mostrando a média do erro absoluto com o desvio padrão para o modelo <i>Gradient Boosting</i> para a velocidade de deformação.	62
Figura 26 – Diagrama de barra mostrando a média do erro absoluto com o desvio padrão para o modelo MLP do scikit-learn para a velocidade de deformação.	62
Figura 27 – Diagrama de barra mostrando a média do erro absoluto com o desvio padrão para o modelo <i>Random Forest</i> para a velocidade de deformação. . . .	63
Figura 28 – Diagrama de barra mostrando a média do erro absoluto com o desvio padrão para o modelo de Árvore de Decisão para a velocidade de deformação.	63
Figura 29 – Diagrama de barra mostrando a média do erro absoluto com o desvio padrão para o modelo SVR para a velocidade de deformação.	64
Figura 30 – Comparação do erro absoluto médio entre o modelo raso (<i>tf_mlp_shallow</i> e profundo <i>tf_mlp_deep</i>).	65
Figura 31 – Comparação do RMSE e R^2 entre o modelo raso (<i>tf_mlp_shallow</i>) e profundo (<i>tf_mlp_deep</i>)	65
Figura 32 – Distribuição e Diagrama de Caixa do erro médio absoluto para o modelo raso de rede neural.	66
Figura 33 – Distribuição e Diagrama de Caixa do erro médio absoluto para o modelo profundo de rede neural.	66
Figura 34 – Distribuição do RMSE dos modelos raso (direita) e profundo (esquerda). . . .	66
Figura 35 – Distribuição do MAPE dos modelos raso (direita) e profundo (esquerda).	67
Figura 36 – Diagrama de barra mostrando a média do erro absoluto com o desvio padrão para o modelo raso.	67
Figura 37 – Diagrama de barra mostrando a média do erro absoluto com o desvio padrão para o modelo profundo.	68
Figura 38 – Diagrama de barra mostrando a média do erro absoluto com o desvio padrão para o modelo raso para a velocidade de deformação.	68

Figura 39 – Diagrama de barra mostrando a média do erro absoluto com o desvio padrão para o modelo profundo para a velocidade de deformação. . . .	69
Figura 40 – XGBoost - Erro por material.	70
Figura 41 – Página Inicial da ferramenta web de predição.	71
Figura 42 – Carregamento do CSV com os dados dos materiais.	72
Figura 43 – Seleção de um material.	72
Figura 44 – Auto preenchimento dos valores de materiais e preenchimento dos experimentos.	73
Figura 45 – Pontos preditos para o modelo utilizando o sistema web.	73
Figura 46 – Curva gerada pelo modelo via o sistema web em MPa para cada ponto da curva predito	74

Lista de tabelas

Tabela 1	–	Quantidade de publicações por ano.	31
Tabela 2	–	Quantidade de elementos presentes no <i>dataset</i> . Fonte: do Autor	44
Tabela 3	–	Comparativo dos modelos.	53
Tabela 4	–	Comparação dos modelos rasos e profundos	64
Tabela 5	–	Estatística descritiva do XGBoost.	69
Tabela 6	–	Melhores Predições do XGBoost	70
Tabela 7	–	Piores predições do XGBoost.	70
Tabela 8	–	Estatísticas de erro por categoria de material.	70

Lista de abreviaturas e siglas

PPGTIC: Programa de Pós-Graduação em Tecnologia da Informação e Comunicação

IA: Inteligência Artificial

VANT: Veículo Aéreo Não Tripulado

API: Interface de Programação de Aplicações

MLP: Perceptron Multicamadas

MAE: Erro Absoluto Médio (*Mean Absolute Error*)

RMSE: Raiz do Erro Quadrático Médio (*Root Mean Squared Error*)

R²: Coeficiente de Determinação

MAPE: Erro Percentual Absoluto Médio (*Mean Absolute Percentage Error*)

CPU: Unidade Central de Processamento

GPU: Unidade de Processamento Gráfico

CSV: Arquivo de Valores Separados por Vírgula (*Comma-Separated Values*)

CAD: Desenho Assistido por Computador (*Computer-Aided Design*)

CNN: Rede Neural Convolucional (*Convolutional Neural Network*)

DNN: Rede Neural Profunda (*Deep Neural Network*)

TG-DNN: Rede Neural Profunda Guiada por Teoria (*Theory-Guided Deep Neural Network*)

U-NET: Arquitetura de Rede Neural Profunda para Segmentação de Imagens

k-fold: Validação Cruzada

Optuna: Biblioteca de Otimização de Hiperparâmetros

API REST: Interface de Programação de Aplicações baseada em REST (*Representational State Transfer*)

SVR: Regressão por Vetores de Suporte (*Support Vector Regression*)

SVM: Máquinas de Vetores de Suporte (*Support Vector Machines*)

ANN: Redes Neurais Artificiais (*Artificial Neural Networks*)

XGBoost: *Extreme Gradient Boosting*

WSL: Subsistema Windows para Linux (*Windows Subsystem for Linux*)

Sumário

1	INTRODUÇÃO	17
1.1	Contextualização do Problema e Justificativa da Pesquisa	17
1.2	Objetivos	20
1.2.1	Objetivo Geral	20
1.2.2	Objetivos Específicos	20
1.3	Hipóteses	20
1.4	Aderência da Pesquisa ao PPGTIC	21
1.5	Organização	21
2	INTELIGÊNCIA ARTIFICIAL	23
2.1	Aprendizado de Máquina	23
2.1.1	Árvores de Decisão	24
2.1.2	Modelos <i>Ensemble</i>	24
2.1.2.1	Florestas Aleatórias	25
2.1.2.2	<i>Boosting</i>	25
2.1.3	Redes Neurais Artificiais	25
2.1.4	<i>Deep Learning</i>	26
3	CONFORMAÇÃO MECÂNICA	27
3.1	Estampagem	27
3.2	Curvas de Escoamento	28
4	REVISÃO SISTEMÁTICA DA LITERATURA	30
4.1	Metodologia	30
4.1.1	Planejamento	30
4.2	Inteligência Artificial na Conformação Mecânica	30
4.2.1	Design de Peças	31
4.2.2	Predição e Detecção	34
5	METODOLOGIA	40
5.1	Tecnologias Usadas	40
5.1.1	Software de Experimentação	41
5.2	Dataset	41
5.3	Análise dos Dados e Dataset de Validação	43
5.3.1	Separação em Treino e Teste	44
5.4	Estratégia de Avaliação	45

5.5	Seleção de Modelos	45
5.6	Otimização dos Hiperparâmetros	45
5.7	Métricas de Avaliação	50
5.8	Apresentação do Sistema ao Usuário	51
6	RESULTADOS E AVALIAÇÕES	52
6.1	Modelos Tradicionais	52
6.1.1	Distribuição de erros	53
6.1.2	Erro por Temperatura	57
6.1.3	Erro por velocidade de deformação	61
6.2	Modelos do TensorFlow	64
6.2.1	Distribuição de Erros	65
6.2.2	Erro por Temperatura	67
6.2.3	Erro por velocidade de deformação	68
6.3	Análise do Melhor Modelo	69
6.4	Ferramenta para o Usuário	71
7	CONCLUSÃO	75
	REFERÊNCIAS	77

1 INTRODUÇÃO

1.1 Contextualização do Problema e Justificativa da Pesquisa

O campo de estudo da Inteligência Artificial (IA) é muito amplo e engloba diversos subtópicos que vão desde realizar ações mais generalistas, como aprender, raciocinar e perceber, até as mais específicas, tais quais jogar xadrez, diagnosticar uma doença e pilotar um VANT (veículo aéreo não tripulado). É uma área que é relevante para qualquer tarefa intelectual e é considerada um campo universal. A IA é constantemente ranqueada como um dos campos de maior crescimento e maior interesse, gerando, atualmente, receitas na casa dos trilhões de dólares americanos. Ela consiste em entender e criar agentes inteligentes, máquinas capazes de computar como agir de forma eficiente e segura em uma grande gama de situações novas. (RUSSELL; NORVIG, 2020)

Por ser um tema muito discutido atualmente, podemos falsamente ter a impressão que a ela pode ser algo relativamente novo. Porém, o matemático inglês Alan Turing, que é chamado por muitos de pai da IA, já escrevia artigos sobre o assunto na década de 1950. Seu artigo intitulado *Computing Machinery and Intelligence* (Maquinário Computável e Inteligência, em tradução livre), focava no conceito de uma máquina que fosse inteligente e, para atingir isso, ele propôs que era necessário medir o que era inteligência, pelo menos para uma máquina. Foi neste trabalho, e tentando resolver esse problema, que surgiu o Teste de Turing. Esse teste consiste em um jogo de 3 jogadores, 2 humanos e um não. Os jogadores devem fazer perguntas abertas entre si e determinar quem é o humano (TAULLI, 2019) (TURING, 1950). No ano de 2014, uma pesquisa mostrou uma máquina que aparentemente passou no teste pela primeira vez na história. Ela enganou 33% dos entrevistados dizendo a eles que era um adolescente de 13 anos chamado Eugene Goostman. Junto a essa informação, os jurados foram ludibriados justamente porque algumas respostas dadas pelo suposto Eugene possuíam erros. (WARWICK; SHAH, 2016)

O Teste de Turing não é único e também não é aceito por todos sem ressalvas. Muitos falam que ele pode ser manipulado. A verdade é que determinar o que é inteligência em si já é algo muito complicado. Para tratar do tema, alguns pesquisadores abordam a inteligência em termos de fidelidade a performance humana, outros preferem algo mais abstrato, a racionalidade, ou fazer a coisa certa. O próprio assunto varia dentro dele mesmo. Alguns consideram inteligência o processo interno de pensamento e razão, já outros focam no comportamento inteligente, ou seja, uma característica interna. Com isso temos duas dimensões: humano vs racional e pensamento vs comportamento. Fazendo o par-a-par de cada, temos 4 métodos que podem ser tratados para medir inteligência: Agir

como um humano, pensar como um humano, agir racionalmente e pensar racionalmente. O Teste de Turing abordava o primeiro dessa lista, existem outros métodos que abordam os outros, porém não faz parte do escopo desse trabalho essa discussão. (TAULLI, 2019) (RUSSELL; NORVIG, 2020)

Alan Turing foi um dos pioneiros, mas não o único. A história da IA começou mesmo a ser escrita em 1943 por Warren S. McCulloch e Walter Pitts no artigo *A logical calculus of the ideas immanent in nervous activity* (Um cálculo lógico das ideias imanentes na atividade nervosa, em tradução livre). Esse trabalho propôs um modelo de neurônios artificiais aonde cada neurônio era caracterizado como estando ligado ou desligado, com uma chaveamento de ligar ocorrendo em resposta ao estímulo de um número suficiente de neurônios vizinhos (MCCULLOCH; PITTS, 1943). Essa ideia possui traços do modelo computacional proposto por Turing, e deveria, já que os autores beberam da fonte do matemático inglês, além de outras. Podemos definir toda a história da inteligência artificial em épocas. O trabalho de McCulloch e Pitts deu início a era inicial da IA, que durou até o início da década de 1950. Logo após veio a era da euforia inicial, aonde haviam grandes propostas e grandes expectativas, que durou até o final da década de 1960 e logo veio a dose de realidade. Em 1957, Hebert Simon afirmará que em 10 anos haveria um computador campeão mundial de xadrez, porém a afirmação caiu por terra devido a vários fatores, mas principalmente a capacidade computacional da época. Foi nesse mesmo período que começaram a aparecer os sistemas especialistas, algoritmos que chamamos de métodos fracos, pois apesar de tentar generalizar um problema, eles não são escaláveis para problemas ou instâncias de larga escala. No final da década de 1980 as redes neurais (neurônios artificiais) voltaram com força e perduram até hoje. Foi nessa época também que os modelos probabilísticos e de *machine learning* começaram e surgir de forma mais consistente. Em 2001, já com a internet se populazirando, começou a era do *Big Data* (ou Dados Grandes, em tradução livre), que nada mais é que uma quantidade titânica de dados. Em 2011, devido aos avanços da capacidade computacional, surgiram os modelos de *Deep Learning* (Aprendizado Profundo, em tradução livre), que são modelos com várias camadas e capazes de generalizar diversas atividades, como por exemplo o GPT-3, base do chatGPT da OpenAI. (RUSSELL; NORVIG, 2020)

Com toda essa evolução da tecnologia, podemos aplicar a IA nos mais diversos setores, Automação de Processos robóticos, Processamento de Linguagem Natural, detecção de doenças, sensoriamento de terremotos, entre uma diversa gama de outros casos (TAULLI, 2019). Este trabalho vai focar no uso da inteligência artificial com o objetivo de melhorar o processo de conformação mecânica, gerando curvas de escoamento de metais sem a necessidade de experimentos reais.

O setor industrial é considerado um dos mais importante de uma economia forte, gerando tecnologia e riquezas para uma nação. A indústria fornece insumos valiosos para

outros setores, como peças e tecnologias para o agronegócio, mineração, construção e para o setor de serviços (GROOVER, 2013). A industrialização influencia positivamente no crescimento econômico de um país, gerando grande retorno e sendo a principal difusora do progresso tecnológico (OREIRO; FEIJO, 2010). Uma nação competitiva se dá através de uma base industrial forte e hoje, no Brasil, temos uma indústria pouco competitiva devido a fatores estruturais e sistêmicos (SILVA et al., 2019).

Dentre os fatores estruturais destacam-se um comportamento inovativo modesto, custo da mão de obra unitária, redução de produtividade e altos salários. Há uma nova geografia de crescimento populacional e econômico que provoca um aumento da demanda de alimentos e recursos naturais. Neste cenário, a produção industrial é crescente, principalmente no intuito de abastecer as demandas dos países em desenvolvimento. A produção industrial se concentra em grandes produtores: Estados Unidos da América, China, Japão e Alemanha, que juntos representam 55% da produção mundial. O Brasil ocupa a 11^a posição, contribuindo com 1,59% da produção industrial, de acordo com dados de 2014. (SILVA et al., 2019)

As novas tecnologias estão modificando os modelos de produção atuais. Essa mudança exige um constante crescimento na capacidade das empresas de fazer o melhor uso possível das novas tecnologias para se tornarem mais competitivas mundialmente. (CNI, 2013)

Dentro das tecnologias de manufatura industrial, destaca-se um conjunto de processos baseados em conformação mecânica. Nestes processos, o material metálico é deformado mecanicamente através de pressões exercidas por forças externas, que alteram suas características de forma. Essa pressão é aplicada por ferramentas que dão forma a produtos de diferentes características e performance. O material de trabalho e o equipamento de conformação são fatores importantes para gerar de forma eficaz o produto final. (HU et al., 2013)

Além do mais, o processo de conformação pode ser dividido em dois, a frio e a quente. O processo de conformação a quente é usado quando o trabalho é feito principalmente com metais mais resistentes e deformação mecânica e com baixa plasticidade (ou seja, baixa capacidade de se moldar quando aplicado a uma força externa). Já a conformação a frio é utilizada para moldar materiais em temperatura ambiente. Assim como a IA, o processo de conformação não é recente. Possuímos registros que data de 4000 A.C., com peças como ferramentas, armas e bens de lazer sendo fabricadas com essa técnica. Atualmente, utilizamos ela de forma totalmente automatizada e em massa, produzindo automóveis, aviões, utilidades domésticas, latas de bebidas e uma infinidade de outros produtos. (HU et al., 2013) (GROOVER, 2013)

Esse trabalho tem como justificativa resolver lacunas encontradas atualmente na geração de curvas de escoamento. Como mostrado na seção "Revisão Sistemática

da Literatura", gerar as curvas possui um grande custo de tempo, dinheiro e pessoal. Normalmente, é preciso que um especialista utilize um laboratório e faça os experimentos de forma manual. Além disso, são necessários equipamentos específicos e amostras dos materiais que serão testados. Tais custos podem ser reduzidos utilizando um sistema com base em inteligência artificial capaz de aproximar as curvas de escoamento, além de liberar o especialistas para tarefas que exijam menos trabalho repetitivo e passível de automatização.

Com base nisso, este trabalho responderá a seguinte questão de pesquisa: algoritmos de aprendizado de máquina são capazes de prever curvas de escoamento de metais a partir de suas características químicas, temperatura, deformação e velocidade de deformação, tornando possível realizar predições sem a necessidade de experimentos laboratoriais?

1.2 Objetivos

1.2.1 Objetivo Geral

O trabalho tem como objetivo avaliar o desempenho de diferentes algoritmos de aprendizado de máquina para aproximar uma curva de escoamento de metais usando sua composição química, temperatura, deformação e velocidade de deformação e apresentar uma ferramenta para que engenheiros possam prever as curvas sem o alto custo de um experimento em laboratório.

1.2.2 Objetivos Específicos

Considerando o desenvolvimento do trabalho e o objetivo geral apresentado, destacam-se os seguintes objetivos específicos:

- Apontar os fundamentos teóricos, técnicos e científicos da conformação mecânica;
- Identificar os modelos de inteligência artificial que podem corroborar com a conclusão desse trabalho;
- Avaliar a performance e erro dos algoritmos através das métricas definidas na Metodologia;
- Desenvolver uma ferramenta web para predição de curva de escoamento.

1.3 Hipóteses

As hipóteses deste trabalho são:

- Algoritmos de aprendizado de máquina são capazes de aproximar uma curva de escoamento;
- O algoritmo de aprendizado de máquina consegue generalizar novos materiais que não estavam em seu conjunto de treinamento;
- O sistema web é capaz de carregar um modelo e gerar uma curva de forma rápida e simplificada mesmo para novos materiais e novos parâmetros de experimentação;

1.4 Aderência da Pesquisa ao PPGTIC

O PPGTIC é um programa de pós-graduação na área de concentração da Tecnologia e Inovação e se divide em três linhas de pesquisa: Tecnologia Computacional, Tecnologia Educacional e Gestão e Inovação. Esse trabalho se adere na linha de pesquisa da Tecnologia Computacional, que tem como objetivo “desenvolver modelos, técnicas e ferramentas computacionais auxiliando na resolução de problemas de natureza interdisciplinar” [PPGTIC, 2018]. Aplicações industriais inteligentes, ou indústria 4.0, se enquadram nessa classificação.

Foram encontrados 4 trabalhos no PPGTIC relacionados a Inteligência Artificial, porém nenhum deles aplicado a conformação mecânica, a indústria ou a predição de erro, demonstrando que esse trabalho abordará um tópico inédito ao programa, se diferenciando dos demais.

Em aderência ao programa, o caráter multidisciplinar é mostrado pela integração das áreas de inteligência artificial, ciência dos materiais, engenharia mecânica e tecnologia da informação. A união dessas áreas contribui para avanços e preenche uma lacuna entre diferentes campos, implementando soluções para desafios industriais.

1.5 Organização

O presente trabalho está organizado da seguinte forma:

Capítulo 1: Capítulo para contextualizar o leitor nos objetivos do trabalho e na pergunta da pesquisa, apresentando a justificativa do trabalho e delimitando na metodologia e aderência ao PPGTIC (Programa de Pós-Graduação em Tecnologia da Informação e Comunicação).

Capítulo 2: Introdução ao assunto de Inteligência Artificial, passando pelos tópicos que foram utilizados neste trabalho.

Capítulo 3: Introdução sobre conformação mecânica, estampagem e curvas de escoamento.

Capítulo 4: Será feito a revisão sistemática da literatura, atualizando o conhecimento com o que está sendo feito na área de conformação mecânica e inteligência artificial atualmente.

Capítulo 5: Metodologia, como foi feita a coleta, processamento e armazenamento dos dados, passando métodos utilizados. Falaremos sobre os algoritmos usados, métodos de treinamento e seleção de modelos e também como foi feita a validação.

Capítulo 6: Resultados dos modelos e uma análise dos erros.

Capítulo 7: Apresentará as conclusões deste trabalho, sugestões de pesquisas futuras e o referencial bibliográfico.

2 Inteligência Artificial

Inteligência artificial pode ser definido como o projeto e construção de agentes inteligentes que recebem percepções do ambiente e executam ações que afetam esse ambiente (RUSSELL; NORVIG, 2020). Ela é a área que tem como objetivo trazer inteligência adequada ao computador para que possa "pensar" e agir de forma similar ao ser humano (LESINSKI; CORNS; DAGLI, 2016).

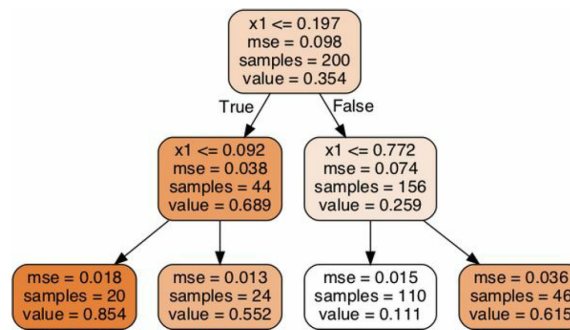
O campo de estudo da Inteligência Artificial (IA) é muito amplo e engloba diversos subtópicos que vão desde realizar ações mais generalistas, como aprender, raciocinar e perceber, até as mais específicas, tais quais jogar xadrez, diagnosticar uma doença e pilotar um VANT (veículo aéreo não tripulado). É uma área que é relevante para qualquer tarefa intelectual e é considerada um campo universal. A IA é constantemente ranqueada como um dos campos de maior crescimento e maior interesse, gerando, atualmente, receitas na casa dos trilhões de dólares americanos. Ela consiste em entender e criar agentes inteligentes, máquinas capazes de computar como agir de forma eficiente e segura em uma grande gama de situações novas. (RUSSELL; NORVIG, 2020)

2.1 Aprendizado de Máquina

O Aprendizado de Máquina, também conhecido como *Machine Learning*, é uma sub área da Inteligência Artificial que dá a programas a habilidade de aprender sem que seja programado de forma explícita. São algoritmos treinados em conjuntos de dados arranjados e que fazem predição em dados nunca vistos antes. Dentro dessa sub área, temos o aprendizado supervisionado e não supervisionado. Aprendizado supervisionado pode ainda ser do tipo classificação ou regressão. No aprendizado supervisionado, temos um mapeamento entre a entrada de dados com uma saída em classes pré definidas. Durante o processo de treinamento, o algoritmo busca correlação entre os dados de treino com suas respectivas saídas. Entre algoritmos comuns de classificação temos *Naive Bayes*, Redes Neurais Artificiais, *Support Vector Machines* e árvores de decisão. (MICHALSKI; CARBONELL; MITCHELL, 2013)

A regressão é exatamente igual a classificação, com exceção que a saída, em vez de uma classe, é contínua. A ideia é que os dados de treinamento sejam acondicionados na função que apresenta o menor erro, que pode ser a distância quadrada dos pontos em relação a curva gerada pelo modelo, por exemplo. Exemplos de modelos de regressão são: predição do preço de uma ação, predição da idade de um determinado grupo de usuários ou predição da temperatura em um local específico. Podemos perceber aqui que estamos prevendo valores no domínio real e não classes. Não estamos prevendo, por exemplo, se

Figura 1 – Exemplo de uma árvore de decisão para regressão.



Fonte: (GERON, 2019).

vai ser frio ou calor, ou se o usuário está ativo ou não (MURPHY, 2012).

Aprendizado supervisionado pode ainda ser do tipo classificação ou regressão. No aprendizado supervisionado, temos um mapeamento entre a entrada de dados com uma saída em classes pré definidas. Durante o processo de treinamento, o algoritmo busca correlação entre os dados de treino com suas respectivas saídas. Entre algoritmos comuns de classificação temos *Naive Bayes*, Redes Neurais Artificiais, *Support Vector Machines* (SVM) e árvores de decisão. (MICHALSKI; CARBONELL; MITCHELL, 2013)

2.1.1 Árvores de Decisão

Modelos de Árvores de Decisão (*Decision Trees*) podem ser definidos como modelos de múltiplas seleções binárias em uma arquitetura de árvore. Dado um novo dado de entrada "x", o processo de seleção de um modelo específico consiste em uma decisão sequencial correspondendo a transversal da árvore de decisão. Para um modelo aprender, precisamos definir alguns hiperparâmetros, incluindo quais variáveis serão escolhidas como critério de quebra de nó e o *threshold* para um *split*. (BISHOP, 2009)

Modelos de *decision trees* podem ser usados tanto para problemas de classificação (*decision tree classifier*) quanto para problemas de regressão (*decision trees regressors*). A principal vantagem de se usar um modelo baseado em árvores de decisão é a interpretabilidade dos resultados, o que é crítico para muitas tarefas de IA. Um grande problema é que elas são muito sensíveis a entradas consideradas como ruído ou pontos fora da curva. Qualquer mínima alteração nos valores gera resultados completamente diferentes. (BISHOP, 2009)

2.1.2 Modelos *Ensemble*

Dado um problema muito complexo de ser resolvido, podemos utilizar de um conhecimento de uma grande população para tentar aproximar uma resposta. A teoria

das massas diz que um grupo grande de pessoas consegue resolver esse problema melhor que um único especialista. Esse tipo de abordagem também pode ser aplicada em modelos de aprendizado de máquina. Quando temos um conjunto de classificadores ou regressores atuando como um único modelo, chamamos-os de modelos *ensemble*. Um exemplo de modelo *ensemble* é o que chamamos de floresta aleatória, onde um grupo de árvores de decisão são treinadas em conjunto. (GERON, 2019)

2.1.2.1 Florestas Aleatórias

As florestas aleatórias, ou *random forests*, são modelos ensemble treinados em cima de múltiplas árvores de decisão. Nele, cada árvore é treinada em um diferente subgrupo dos dados e o valor será a média de cada árvore, para casos de regressão, ou a maioria dos votos em casos de classificação. Essa abordagem, seja média ou votação, faz com que o grupo acerte mais do que o melhor modelo dentro deste grupo. (GERON, 2019) (BISHOP, 2009)

2.1.2.2 Boosting

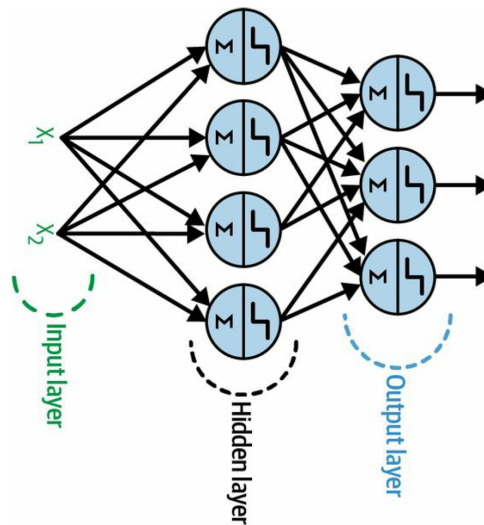
O *boosting* (antes chamado de melhoramento de hipótese) refere-se a qualquer modelo *ensemble* em que diversos classificadores fracos são combinados como um classificador forte. Nele, cada classificador é adicionado de forma sequencial e o seu subsequente aprende com os erros do classificador anterior. Podemos citar diversos algoritmos que utilizam essa técnica, como Adaboost e *Gradient Boost*. Neste trabalho, usaremos o segundo. Nele, cada classificador aprende com o erro residual do anterior. (GERON, 2019)

2.1.3 Redes Neurais Artificiais

Redes Neurais Artificiais são técnicas de *machine learning* com nós interconectados que tentam simular o mecanismo de aprendizado de seres biológicos. Uma rede neural computa a função das entradas propagando os valores calculados de um neurônio de entrada para um neurônio de saída e usando os pesos como parâmetros intermediários. O aprendizado ocorre pela mudança dos pesos que conectam os neurônios (AGGARWAL, 2019).

A função da rede é resultado da computação cascadeada dos nós (ou neurônios) individuais através da definição correta dos pesos. Ela aprende como relacionar uma ou mais entradas em uma ou mais saídas usando exemplos da entrada. Através dessa técnica, é possível computar praticamente qualquer função da entrada até a saída. As redes podem ser de camada única ou multicamadas. Quando temos uma rede neural de camada única, chamamos de Perceptron (AGGARWAL, 2019). Chamamos cada nó de uma unidade. Uma unidade, tradicionalmente, calcula a soma dos pesos da entrada do seu antecessor e então aplica uma função de ativação não linear para produzir uma saída (MCCULLOCH;

Figura 2 – Exemplo de uma rede neural. Escrito em verde temos a camada de entrada, em preto as camadas ocultas e em azul a camada de saída.



Fonte: (GERON, 2019).

PITTS, 1943). As redes neurais podem ser definidas em diferentes tipos, dependendo a sua função, e são utilizadas em situações específicas.

2.1.4 Deep Learning

O *Deep Learning*, são redes profundas, redes neurais onde temos uma grande quantidade de camadas e parâmetros (geralmente na casa dos milhões ou bilhões de parâmetros). De acordo com a estatística clássica, à medida que aumentamos o número de camadas de um modelo, este aumenta sua acurácia até um ponto e depois decai. O que a *Deep Learning* mostrou foi que esse comportamento não é necessariamente verdade. A acurácia realmente decai porém, à medida que continuamos aumentando o modelo, ela volta a aumentar e até passar o pico anterior (MURPHY, 2012). Hoje, os principais modelos são baseado em redes profundas, como Dall-E e Midjourney, modelos capazes de criar arte com base em uma entrada de texto, e também modelos de língua para geração de texto, código, tradução e outras tarefas do tipo, como GPT, Gemini e Grok.

3 Conformação Mecânica

A conformação mecânica é um conjunto de processos de alteração geométrica por deformação plástica, através de forças exercidas por ferramentas que dão forma ao material metálico. Ela inclui um grande número de processos: laminação, forjamento, trefilação, extrusão e conformação de chapas. Esses processos têm em comum o fato de que para produzir a peça, um esforço do tipo compressão, tração, flexão e cisalhamento, têm de ser aplicado sobre o material. (SCHAEFFER, 2004) (MORO; AURAS, 2007)

Como vantagem desses processos, têm-se: bom aproveitamento da matéria prima, rapidez na execução, controle de propriedades mecânicas e possibilidade de grande precisão e tolerância dimensional. Entretanto, a conformação exige maquinário pesado e de alto custo, o que se justifica economicamente para produção em larga escala. (MORO; AURAS, 2007)

A qualidade final das peças produzidas por conformação mecânica depende de dezenas de parâmetros. Dentre os parâmetros fundamentais, destacam-se a tensão e a deformação. A tensão pode ser definida como a intensidade da força por unidade de área, enquanto a deformação é uma medida da quantidade deformada. Quando um corpo é deformado plasticamente, os pontos deste corpo são deslocados. A deformação quantifica esse deslocamento. (HOSFORD; CADDELL, 2007)

Outro parâmetro crucial dos processos de conformação mecânica é a temperatura. Os processos podem ser classificados de acordo com a temperatura em que são realizados. Quando o trabalho é feito em uma temperatura maior que a temperatura que provoca a recristalização do metal, chamamos de trabalho a quente. Em temperaturas menores que esta dizemos que o trabalho é a frio. (ROCHA, 2012)

3.1 Estampagem

A estampagem é uma operação que envolve a prensa de uma chapa metálica e suas bordas sob uma cavidade matriz. A chapa é esticada e a pressão nos anéis de estiramento controla o quanto do material é prensado sob a matriz (HOSFORD; CADDELL, 2007). A forma final depende parcialmente ou totalmente da forma da matriz. Dependendo da complexidade da forma final, pode-se exigir várias etapas para chegar no resultado esperado. (SCHULER, 1998)

O processo de conformação de chapas pode ser dividido em dois grandes grupos: estampagem profunda ou embutimento e conformação geral. Na conformação plástica a partir de chapas, as operações de conformação são feitas em de chapa cortada. Os

Figura 3 – Exemplo das etapas para fabricação de um cárter de óleo



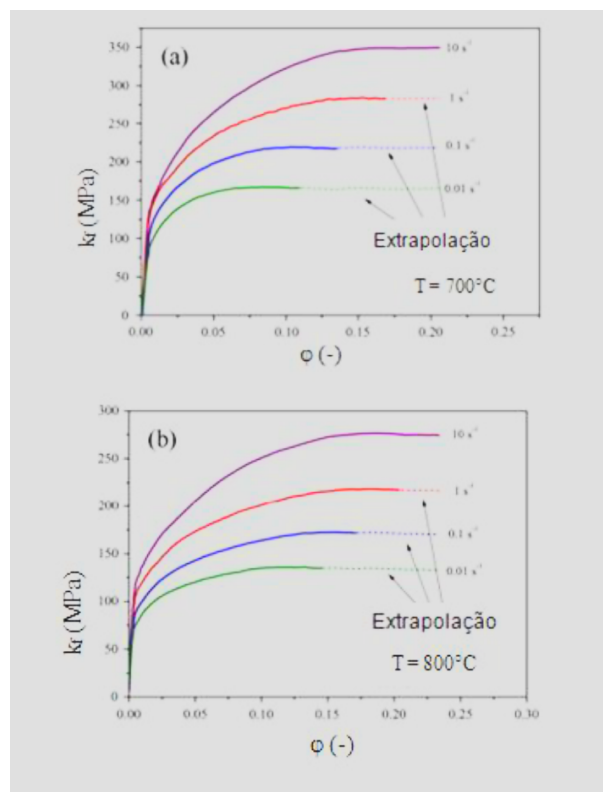
Fonte: (SCHULER, 1998).

materiais para ferramentas de estampagem são selecionados considerando: tamanho e tipo, temperatura de trabalho e natureza do material da peça. Os materiais mais comuns são os aços-ligas. (ROCHA, 2012)

3.2 Curvas de Escoamento

As curva de escoamento parâmetros importantes utilizadas em aplicações de conformação mecânica, que caracteriza o comportamento dos materiais em regimes de deformação plástica, fornecendo a tensão de escoamento do material como função da deformação verdadeira (ERCOLANI; KEHL, 2021). Elas nos diz uma das características mais importantes para a definição de um processo de conformação mecânica a frio, a resistência que o material oferece à deformação. Alguns fatores são decisivos para a determinação do comportamento da curva de escoamento, tais como, a composição química do material, e a sua microestrutura prévia ao processo de conformação (SANTOS et al., 2018).

Figura 4 – Curvas de escoamento do aço 22MnB5 para diferentes temperaturas e velocidades de deformação obtidas a partir do ensaio de tração para as temperaturas de: (a) 700°C, (b) 800°C



Fonte: (ROSIK, 2018).

4 Revisão Sistemática da Literatura

4.1 Metodologia

Para a revisão sistemática da literatura, o trabalho será dividido em três fases: planejamento, revisão e seleção. A etapa de planejamento é a definição da pergunta de pesquisa e objetivos, realizados no capítulo 1. Na revisão, será feito um teste de aderência e palavras-chave, a definição das estratégias de busca, seleção de artigos e a varredura nas bases de dados. Na seleção é aplicado a filtragem e classificação dos artigos.

4.1.1 Planejamento

Na etapa de planejamento, definimos a estratégia de busca de artigos que serão usados na revisão da literatura. Como critérios de inclusão, foram selecionados artigos em inglês com as palavras chaves citadas abaixo, revisado por pares, publicados em revistas e conferências e com texto completo disponível. Como critério de exclusão, foram removidos os artigos duplicados e estudos que não são relacionados a área desta pesquisa nem são aplicados. Como texto de busca para encontrar os artigos nas base de dados propostas, utilizamos o seguinte padrão:

("artificial intelligence"OR "machine learning"OR "deep learning") AND ("mechanical forming"OR "metal forming"OR "sheet metal forming"OR "forging"OR "stamping")

A primeira parte da seleção, foi verificar pelo título e pelo resumo se o trabalho não é relacionado a área de IA aplicada a conformação. Durante essa etapa, foram removidos 531 artigos. Desses, 5 eram artigos duplicados.

Após, os artigos foram então lidos e aqueles que não se encaixavam foram removidos. Foram removidos aqueles em que a inteligência artificial não era o foco do trabalho. Durante essa etapa, foram removidos mais 74 artigos. Os que sobraram foram os selecionados para entrar na revisão. Ao final, selecionamos 18 artigos.

4.2 Inteligência Artificial na Conformação Mecânica

A conformação, em poucas palavras, tem como objetivo criar peças metálicas no formato, dimensões e propriedades desejadas para atender múltiplos requisitos funcionais como, por exemplo, na indústria automotiva e aeroespacial, durabilidade a condições adversas e leveza. Os processos da conformação são conhecidos por serem difíceis de prever e controlar devido às condições complexas de carga e fluxo de cada material. Os estados dos materiais são invisíveis para sensores externos. Os pesquisadores têm se esforçado para

desenvolver modos de garantir a sua qualidade, tendo a Inteligência Artificial ganhando destaque nas pesquisas mais recentes. (CAO et al., 2024)

A área de conformação mecânica é abrangente e com diversas técnicas de conformação. Foi possível encontrar trabalhos desde a extrusão (DING, 2024) e laminação, até na modelagem de materiais (WANG et al., 2024). Dentro dos processos da conformação mecânica, temos pesquisas na área de simulação, modelagem, predição de defeitos, design de peças, ferramentas e processos, controle de processos e certificação. Sobre os modelos e algoritmos de IA utilizados, as técnicas também foram abrangentes, tendo pesquisas com modelos mais simples como *Support Vector*, regressão linear e logística, árvores de decisão e redes neurais e também pesquisas utilizando modelos maiores e técnicas mais complexas, como *Deep Learning*, redes neurais recorrentes, redes neurais em grafos e modelos de *Self-Attention*. (CAO et al., 2024)

A busca foi feita por artigos publicados dentro da janela temporal de 2018 até 2024. A quantidade de artigos publicados por ano encontra-se na tabela 1. A figura 5 mostra uma curva crescente de publicações na área em ambas as bases de dados.

Tabela 1 – Quantidade de publicações por ano.

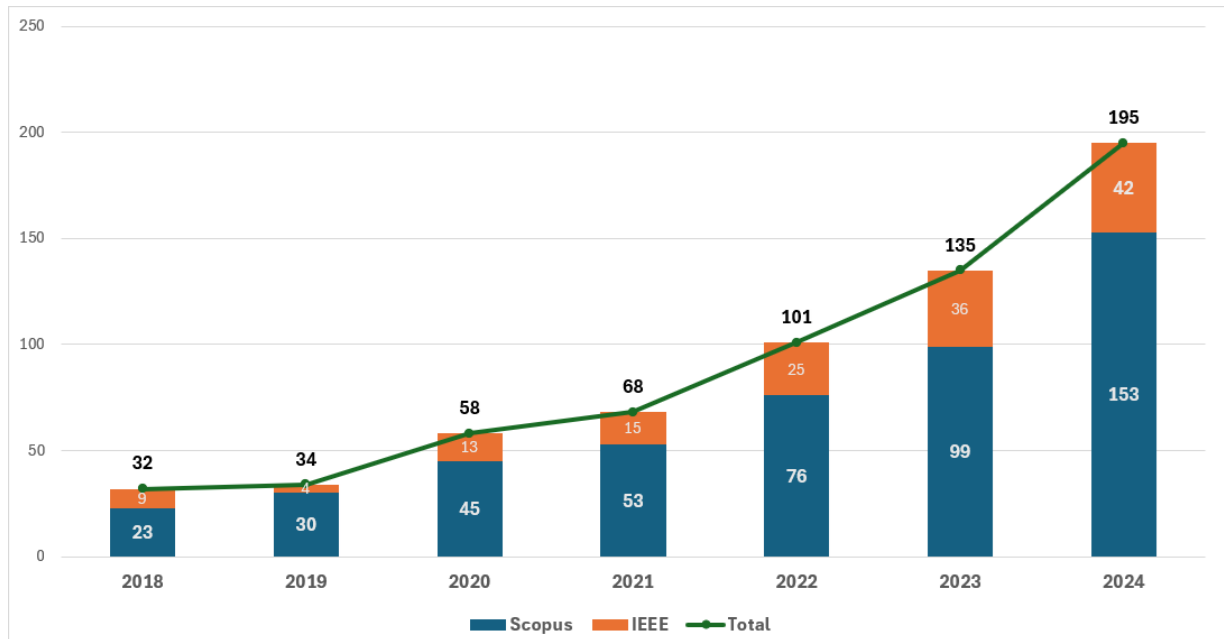
Ano	Scopus	IEEE	Total
2024	153	42	195
2023	99	36	135
2022	76	25	101
2021	53	15	68
2020	45	13	58
2019	30	4	34
2018	23	9	32

Fonte: Do Autor

4.2.1 Design de Peças

A conformação mecânica vem sendo usada há milênios, sendo os processos de dobra e estampagem os mais comuns. Neste último século, esses processos vêm sendo aperfeiçoados continuamente para atender a uma demanda cada vez maior por materiais mais leves e resistentes. Um problema que essa busca traz e que vem sendo alvo de pesquisas é o grande efeito de retorno elástico inerente às ligas de alumínio e de ferro de alta resistência. A maioria dos trabalhos na área foca no desenvolvimento de métodos de compensação em experimentos ou simulações numéricas. O trabalho de (LIU et al., 2021) usa uma abordagem diferente através do *Deep Learning*. A busca por soluções utilizando inteligência artificial vem do consumo considerável de tempo para o ciclo de desenvolvimento e o alto custo financeiro e computacional dos experimentos. A hipótese levantada no trabalho é que algoritmos de *machine learning* trazem uma capacidade de

Figura 5 – Número de artigos que foram publicados utilizando os critérios de busca definidos neste trabalho



Fonte: Do Autor

predição notável em pouco tempo, o que pode trazer benefícios para a área. (LIU et al., 2021)

O trabalho apresenta uma série de experimentos numéricos para avaliar a eficácia de uma SVR (*Support Vector Regression*), redes neurais profundas (DNN, do inglês *Deep Neural Networks*) e redes neurais profundas guiadas por teoria (TG-DNN, do inglês *Theory-Guided Deep Neural Networks*), um tipo de DNN que incorpora conhecimento teórico ou físico prévio do processo. As redes foram treinadas para prever a carga necessária para dobrar a chapa metálica, tendo como entrada somente o desenho CAD (*Computer-Aided Design*) da peça. (LIU et al., 2021)

Como resultados, após realizar o *fine-tuning* da *learning rate* e do coeficiente de regularização, o modelo foi treinado 3 vezes com o melhor resultado de L2 encontrado, e o erro foi calculado para esses modelos, tirando a média para apresentar como resultado final. Os autores comparam os modelos de SVR e DNN e de DNN com o TG-DNN. Os dados de treino e teste foram separados em 80%:20%, 50%:50% e 20%:80%, com amostras selecionadas de forma aleatória. A SVR mostrou um grande desvio da resposta verdadeira comparada à DNN, que apresentou bons resultados. Já a TG-DNN apresentou desempenho muito similar ao da DNN. Esse trabalho trouxe como contribuição as comparações de modelos e um ponto de partida para o *tuning* de hiperparâmetros de redes neurais na área de conformação mecânica. (LIU et al., 2021)

Outro trabalho na área de design de peças utilizando IA foi desenvolvido por

(SAUER; SCHLEICH; WARTZACK, 2018), mas agora aplicado a estampagem. Assim como no trabalho de (LIU et al., 2021), os autores têm como motivação a necessidade de criação de peças cada vez mais complexas e eficientes. Os autores também levantam que com o passar do tempo, a quantidade de peças desenhadas e geradas se tornou tão grande que já se fala em *big data* para a área, resultando em um dilema para os pesquisadores e profissionais da área. De um lado, o design de peças mudou de um padrão bom e ruim baseado em regras para um sistema mais complexo, com problemas de otimização de múltiplos objetivos. Do outro lado, os designers possuem uma quantidade de informação e processos nunca vistos antes. Esse dilema criou o que é chamado de *Workbench* de Engenharia, que dá aos designers uma ferramenta capaz de prover informação sobre o produto desenvolvido no tempo correto. (SAUER; SCHLEICH; WARTZACK, 2018)

O trabalho também utilizou a aquisição de dados via o que os autores chamaram de mineração de dados, mas que era um processo de aumento de dados, utilizando um método de geração de conhecimento com base em dados experimentais. A ferramenta utilizada foi o SLASSY, um programa de autoaprendizado para design de peças formadas por conformação. Os dados gerados foram salvos em uma planilha e avaliados por um especialista humano. A ideia de um *Workbench* de Engenharia é o desenvolvimento de metamodelos, que podem ser caracterizados como uma fórmula matemática que representa certo material. Os modelos até então usados para a geração dos metamateriais eram regressão linear, regressão polinomial, regras M5 e árvores M5. Esses modelos permitem somente a predição de propriedades em um nível de peça inteira, sendo que as pesquisas atuais exigem uma granularidade maior. (SAUER; SCHLEICH; WARTZACK, 2018)

(SAUER; SCHLEICH; WARTZACK, 2018) resolve o problema ao implementar uma rede neural profunda para prever a propriedade do material a nível local. Foi utilizado uma divisão dos dados em treino e teste, em uma proporção de 80:20, e o R^2 -COP (Coeficiente R^2 de Prognóstico, do inglês *R2-Coefficient of Prognosis*) como métrica de validação. Durante o treino, os autores também consideraram a possibilidade de *overfitting* dos dados. Como método preventivo, eles analisaram a convergência das curvas de treino e teste e utilizaram um *dropout*, sem especificar a chance de dropout nem em quais camadas e neurônios foram aplicados. O modelo gerado foi alimentado no SLASSY e foi possível gerar os metamateriais e prever as propriedades em pequenos locais, o que não é possível com os modelos até então usados. Os autores recomendam o uso para protótipos para depois então serem aplicados a um nível real. (SAUER; SCHLEICH; WARTZACK, 2018)

O trabalho de (ATTAR; FOSTER; LI, 2023b) também cita o problema de baixa granularidade na estampagem de geometrias com características de baixa escala. Aqui novamente é citado a exigência cada vez maior de peças eficientes e complexas e como modelos clássicos não atendem mais as necessidades da pesquisa e da indústria. Como diferença de outros trabalhos, os autores utilizaram Redes Neurais Convolucionais (CNN,

do inglês *Convolutional Neural Network*) para criar modelos substitutos para a predição da possibilidade de manufatura de uma chapa metálica para estampagem, criando uma nova abordagem de treinamento de uma rede neural para gerar a função de distância sinalizada, que é usada para representar de forma implícita a geometria da peça. (ATTAR; FOSTER; LI, 2023b)

Para a amostragem de dados, os autores geraram um *dataset* de geometrias que foi alimentado para a CNN. Eles usaram uma técnica popular para geração de dados chamado de *Latin Hypercube (LHC) DoE*. Essa técnica cria uma distribuição uniforme aproximada de amostras dentro do espaço de design evitando recorrências. Com isso foram gerados datasets de treino e teste, com amostras diferentes para cada um, com um total de 1200 amostras para treino e 300 para teste. Os dados foram gerados utilizando apenas o CAD e não simulações de dados reais, tornando o processo computacionalmente barato e rápido. (ATTAR; FOSTER; LI, 2023b)

Para aproximar uma SDF para uma geometria indexada, o trabalho se valeu de um *auto-encoder*. A entrada do *auto-encoder* era um vetor de espaço latente concatenado com as coordenadas x, y e z da *query*. Os autores então avaliaram duas abordagens para aprender o SDFs da geometria de estampagem. Na primeira abordagem, testaram o estado da arte realizando fornecendo a rede os valores reais como *ground truth*. Na segunda abordagem, foi explorado os fatores chave da geometria do SDF para aprende-la de forma implícita, não requerendo que seja fornecido um *ground truth* e melhora a acurácia para escalas próxima do zero. (ATTAR; FOSTER; LI, 2023b)

(ATTAR; FOSTER; LI, 2023b) finalizam entregando uma forma inédita de gerar SDFs utilizando CNNs e uma abordagem implícita. O modelo resultante do trabalho conseguiu uma acurácia geométrica melhor que o estado da arte, capturando características muito pequenas, o que não acontece atualmente. Esse trabalho gerou um segundo que é uma aplicação do modelo proposto, em (ATTAR; FOSTER; LI, 2023a).

4.2.2 Predição e Detecção

Além do design de peças, pesquisas também estão sendo feitas na predição de espessura das peças metálicas durante o processo de estampagem. Assim como os métodos de *machine learning*, como vistos nas pesquisas anteriores, conseguiram melhorar o processo de fabricação de peças, garantir a qualidade do produto final e saber os limites físicos do material. O controle de qualidade também está se beneficiando de pesquisas na área de inteligência artificial, com trabalhos em detecção de defeitos aparecendo nas buscas.

O uso de redes neurais convolucionais tem aparecido com certa frequência em trabalhos de detecção de defeitos em estampagem a frio. O trabalho de (YE et al., 2021) cita que um dos grandes problemas atuais da detecção de defeitos é a inspeção manual

visual, propensa a erros e muito lenta, além de necessitar especialistas em muitos casos, que também contribui para um aumento nos custos de produção. Os autores também citam que isso traz a necessidade de uma modernização e melhora nos processos, o que motivou a escrita do trabalho. Como solução, eles propõem o uso de *Faster R-CNN*. (YE et al., 2021)

A ideia do artigo é utilizar os métodos de *Deep Learning* para detectar defeitos, desenhando uma borda ao redor do defeito, tendo uma imagem da peça como entrada. Os autores combinaram uma R-CNN, adicionaram como a espinha dorsal do projeto uma Resnet50 e, para extrair os defeitos com formas diferentes, usaram redes convolucionais deformável. Os dados utilizados para treino e teste foram retirados da competição de perfil de detecção de defeitos em alumínio (ALIBABA, 2018), que foram separados em treino, teste e validação na proporção 60:20:20. Para avaliar o modelo, os autores utilizaram a taxa de acurácia. Para validar o modelo proposto, eles compararam com outros modelos, YoloV3, *Faster R-CNN* e *Mask R-CNN*. (YE et al., 2021)

Além da taxa de acurácia, os autores compararam o tempo de predição. A YoloV3 se mostrou o modelo mais rápido, porém com a pior acurácia. O modelo de *Mask R-CNN* teve um incremento de 6% comparado a YoloV3 e também foi melhor que a *Faster R-CNN* com um incremento de 2%, porém o tempo de detecção foi 0.3 e 0.2 segundos mais lento, respectivamente. O modelo proposto pelos autores, chamado de *Improve Faster R-CNN* apresentou um desempenho muito superior aos outros, com acréscimo de 24%, 21% e 8%, comparados com a YoloV3, *Faster R-CNN* e *Mask R-CNN* respectivamente, porém teve um tempo de resposta maior próximo da *Faster R-CNN*, sendo apenas 0.02 segundos mais lenta. (YE et al., 2021)

Já (ZHU; LI, 2023) utilizou CNNs para prever o efeito retorno elástico, ou *springback*. Ligas de alumínio de aço possuem um efeito *springback* muito grande e são muito propícios a defeitos dessa natureza (LIU et al., 2021). O retorno elástico causa, principalmente, problemas de geometria das peças estampadas. Na pesquisa, os autores propõem um modelo de CNN com base em um modelo *surrogate*, ou modelo substituto, para prever o comportamento de afinamento e *springback* para estampagem a frio. (ZHU; LI, 2023)

Os autores selecionaram redes conhecidas com U-NET, redes capazes de receber uma imagem como entrada e retornar também uma imagem como saída. Foram treinados modelos para dois tipos de peças, uma dobra em U e a porta de um carro. Os dados para o treino de ambos os modelos foram gerados pelos autores para este trabalho. A dobra em U é usada para testes do efeito elástico e afinamento do material e por isso foi também adotado na pesquisa. Para a geração dos dados, foram utilizados 2 modelos de *surrogate* com simulações de elemento finito, um para a dobram em U e um para gerar as portas. (ZHU; LI, 2023)

Após, foram treinados 2 modelos de *surrogate* utilizando CNNs. A rede foi base-

ada na Res-U-Net, uma rede em U baseado no modelo Resnet (ZHANG; LIU; WANG, 2018). Os autores optaram por essa arquitetura para evitar o problema de gradiente de desaparecimento (ou, pelo nome mais comum usado, *vanishing gradient*). O processo de treinamento foi básico, aonde a forma geométrica 2D das imagens eram alimentados na rede que então gerava os campos de afinamento e *springback* da peça. O erro era calculado em cima das imagens alvos e eles utilizaram o erro médio quadrado para calcular o quão correto a predição era. Como conclusão, os autores ressaltam a habilidade do modelo de CNN gerar em quase tempo-real a predição dos campos, podendo ser adotado na indústria para ajudar na criação de componentes conceituais e finais, sem a necessidade de conhecimento de manufatura por parte dos designers. (ZHU; LI, 2023)

No artigo de (LOW et al., 2023), as redes neurais convolucionais foram utilizadas para detectar defeitos geométricos na estampagem incremental de peças (da sigla inglesa SPIF). Aqui, como em outros casos, o problema de não acurácia na peça final vem do efeito de *springback*. Como diferença desse trabalho, os autores pontuam a utilização de um modelo automático e flexível para quantificar problemas de geometria locais. No trabalho os autores apresentam o uso do *framework* CNN-FP (*Convolutional Neural Network - Forming Prediction*), um modelo que prevê os resultados e distribuição de erros de uma peça em geometria CAD. (LOW et al., 2023)

A CNN-FP proposta é um modelo de imagem-para-imagem (*Pix-2-Pix*) modificado para prever o desvio geométrico dos SPIF utilizando apenas as formas em CAD. A rede foi modificado para que pudesse trabalhar com imagens de maior resolução, já que os testes iniciais feito pelos autores demonstrarem resultados ruins quando trabalhando com a dimensão original da rede CNN-FP. Com essas modificações, a rede recebia uma imagem em escala de cinzas de, 2D, de 16 bits. (LOW et al., 2023)

Uma série de experimentos SPIF foram conduzidos para gerar os dados de treino e validação, utilizando escaneamento e geometrias 3D. Após o treinamento da rede, os autores chegaram a resultados satisfatórios para a predição de erros geométricos com uma raiz do erro médio quadrático de 0.2435mm. Para geometrias contendo ângulos que não foram usados para o treinamento da rede, o modelo foi capaz de prever o erro com uma raiz do erro médio quadrático máximo de 0.5078. Já quando utilizando uma geometria complexa que também não estava no treino, o modelo teve resultados mistos. Para profundidades de 10mm e 30mm os resultados foram satisfatórios, porém houve uma grande queda de performance em profundidades de 40mm e 50mm. Para os autores, a rede demonstrou uma grande capacidade de prever o efeito de *springback* usando apenas os desenhos de CAD. (LOW et al., 2023)

Ainda com o uso das redes convolucionais, (WANG et al., 2024) também citou, como outros autores, o desafio da detecção de defeitos, sendo o processo, em sua maioria da vezes, feito de forma visual por especialistas. Nesse trabalho, foi utilizado algumas técnicas

de IA mais avançadas, como transferência de conhecimento para uma rede Resnet50 para detectar defeitos em peças, mesmo em um pequeno conjunto de dados. Neste trabalho, porém, a detecção foi feita nas chapas metálicas antes de serem prensadas. (WANG et al., 2024)

Para os autores, a detecção de defeitos como riscos, picotamento e ferrugem, atrapalham a qualidade das peças, afetando a qualidade das peças e seu tempo de vida. O estudo abordou a complexidade das imperfeições superficiais, o problema dos resíduos de óleo e como a alta refletividade pode dificultar a detecção de defeitos nas chapas metálicas. (WANG et al., 2024)

Diferente dos outros estudos, que utilizaram dados do CAD, esse trabalho fez a captura de fotos das peças utilizando uma câmera industrial CCD de alta qualidade. Durante o processo de coleta das imagens, houve uma preocupação em evitar que as condições ambientais da oficina de trabalho afetassem as imagens. Os autores fizeram a captura em um ambiente controlado, com proteção contra luz e utilizando um sistema de iluminação coaxial para garantir que a luz fosse consistente em todos os locais da foto. Também foram feitas técnicas de aumento de dados introduzindo ruído nas imagens, técnica que aumenta quantidade de imagem e melhora a capacidade de generalização da rede neural. (WANG et al., 2024)

Quanto a rede neural convolucional, o trabalho partiu de uma Resnet50 pré-treinada. Assim como em outros trabalhos, a Resnet foi selecionado para evitar o problema de *vanishing gradient*. A ResNet foi treinada em cima do conjunto de dados conhecido como *ImageNet*. Para fazer o modelo aprender o contexto das imagens da pesquisa, ele utilizam *transfer learning*, ou transferência de conhecimento, que faz com que uma nova rede, aqui chamada pelos autores de *Improved Resnet*, adquira novo conhecimento, a partir de um *dataset* menor, aproveitando o conhecimento da rede original. Pra isso, a camada de saída da ResNet foi substituída por camadas lineares, função de ativação e camadas de *Dropout*. (WANG et al., 2024)

Como resultado do trabalho, a rede conseguiu uma taxa de acurácia de 89,8%, o que, para os autores, foi impressionante. Outro fator de surpresa foi a velocidade de detecção, 680 imagens por minuto, ou uma imagem a cada 0.8 segundos. O maquinário de estampagem trabalha a uma velocidade de 30 batidas por minuto e o molde desse tipo de maquinário consegue 60 peças no mesmo espaço de tempo. Para os autores, isso demonstra que o sistema pode ser acoplado sem nenhum prejuízo de tempo na linha de produção atual. A precisão de detecção da rede foi de 0.1mm na superfície de metais. Uma contribuição grande dos autores foi provar que a introdução do ruído nos dados de treino e teste foi um fator crucial nesta aplicação. (WANG et al., 2024)

(GUO et al., 2024) utilizou uma abordagem diferente dos autores anteriores. Para detectar rachaduras na manufatura automobilística, o trabalho propôs a criação e uso de

um *framework* em 2 estágios incrementais e adaptativos. Os autores citam a dificuldade em se implementar uma abordagem totalmente supervisionada para detecção de rachaduras pois há uma escassez de dados de treino, que são difíceis de conseguir. Os custos de tempo e pessoal para fazer a anotação desses dados são complicações reais. Por outro lado, modelos não supervisionados, para os autores, possuem uma acurácia limitada e não trazem resultados satisfatórios. Os *datasets* são poucos o que fazem os pesquisadores usarem dados públicos que não espelham os desafios reais de uma linha de produção. (GUO et al., 2024)

Para solucionar os problemas levantados, os autores criaram um *framework* chamado de *Adaptative Incremental Two-Stages*, ou AITS. Ele é dividido em dois estágios. No primeiro, um modelo não supervisionado de detecção de anomalias detecta rachaduras. O segundo estágio então usa um algoritmo de segmentação supervisionado para etiquetar os defeitos detectados no Estágio I e melhorar a área de detecção. O primeiro estágio utiliza uma rede convolucional pré-treinada para aprender uma matriz de parâmetros gaussianos e gerar um mapa de calor das anomalias. O segundo estágio recebe o mapa de calor e gera as etiquetas através de uma rede *AffinityNet*, que é então repassada para uma rede de segmentação. Como dados de treino e teste, os autores criaram um *dataset* com dados reais de uma indústria automotiva. Para testar a generalização do modelo, eles também aplicaram a rede no KolektorSDD, um *dataset* de defeitos de 2019. Como métrica, foi escolhida a curva RoC. (GUO et al., 2024)

Os autores compararam o método estudado com modelos considerados estado-da-arte: SVDD, SPADE, STFPM e PaDiM. Eles testaram a acurácia dos pixels e o mIoU, métrica que calcula o *overlapping* da segmentação predita com a real. O modelo proposto foi melhor que os concorrentes, chegando a alcançar 99,026% de acurácia de pixels, contra 97,508% do PaDiM (até então o melhor na tarefa) e 66,268 de mIoU contra 57,131 do PaDiM. Para os autores, outro fator positivo do modelo proposta é o uso de um mecanismo de gatilho acumulativo para treinamento adaptativo. Esse gatilho melhora o modelo continuamente utilizando o *feedback* da saída dos resultados da detecção para melhorar o modelo. O trabalho também apontou limitações, como a necessidade de ajustes do treinamento incremental para cada caso, que precisa ser estudo por um especialista. Outro ponto de melhora é a performance que pode ser otimizada para reduzir possíveis vies do modelo. (GUO et al., 2024)

No trabalho de (MEHTEDI et al., 2019), foram utilizados modelos baseados em redes neurais para prever a curva de escoamento de uma liga específica de magnésio, chamada de ZAM100. Neste, desenvolveram um dataset baseado em coleta de dados com testes reais em uma faixa de temperatura e velocidade de deformação. Os autores destacam a relação entre temperatura, curva de estresse, deformação, velocidade de deformação e dos materiais é muito útil, porém é altamente cara e leva muito tempo para se conseguir.

Outro ponto de alerta levantado pelos autores é que é difícil achar uma relação entre esses dados e a saída utilizando técnicas matemáticas tradicionais, para resolver isso os autores trataram o problema como um problema de aprendizado de máquina, utilizando redes neurais para prever essas relações. (MEHTEDI et al., 2019)

Para a realização do trabalho, foi utilizado a liga de magnésio ZAM100 em discos de 3.1 mm de espessura. Para a rede neural, foi utilizado uma arquitetura de 1 camada de entrada de tamanho 6, 2 camadas ocultas com 6 neurônios cada e uma camada de saída com 1 neurônio (problema de regressão). Durante a fase de treino, os autores optaram por utilizar um *early-stop*, técnica que para o treinamento antes do número de épocas definido caso o modelo já não melhore suas métricas de teste, evitando o *overfitting*. (MEHTEDI et al., 2019)

Como resultado, os autores viram que valores de temperatura e velocidade de deformação, o modelo se saiu muito bem com erros relativamente pequenos, conseguindo prever tanto o formato quanto os valores da curva. Os autores concluíram que, para este problema, as redes neurais artificiais se saíram muito bem, conseguindo captar os nuances das curvas de escoamento da liga ZAM100. (MEHTEDI et al., 2019)

Já como um trabalho mais similar com a proposta desta dissertação, (KOCATÜRK et al., 2021) utilizou redes neurais artificiais para predição da curva de escoamento de aço forjado. Aqui também se utilizou diferentes temperaturas (20, 100, 200 e 400 °C) com 3 diferentes taxas de deformação (1, 10 e 50 s⁻¹). Assim como no trabalho de (MEHTEDI et al., 2019), os dados para treino e teste foram coletados em experimentos reais. (MEHTEDI et al., 2019)

Nesse trabalho, eles tentaram prever a tensão verdadeira com base na temperatura, velocidade de deformação e deformação plástica. A arquitetura utilizada foi um rede neural com uma camada de entrada de tamanho 3, uma camada oculta com 9 neurônios e 1 camada de saída com 1 neurônio (problema de regressão). Os resultados foram similares aos dados experimentais, com uma raiz do erro quadrado médio de 7,31. (MEHTEDI et al., 2019)

Como diferencial das pesquisas acima citadas, este trabalho propõe investigar os modelos de aprendizado de máquina que melhor aproximam as curvas de escoamento com base na composição química dos materiais, temperatura e velocidade de deformação, disponibilizando uma ferramenta web para interagir com esses modelos de forma simplificada, podendo o usuário adicionar novos materiais e testar o comportamento deles em temperaturas e velocidade de deformação personalizadas.

5 Metodologia

Neste capítulo, será definido o processo metodológico adotado para o desenvolvimento do presente trabalho. De acordo com a natureza desta pesquisa, ela é classificada como tecnológica aplicada, sendo necessário o desenvolvimento de um protótipo com o objetivo de alcançar os objetivos propostos no capítulo 1.

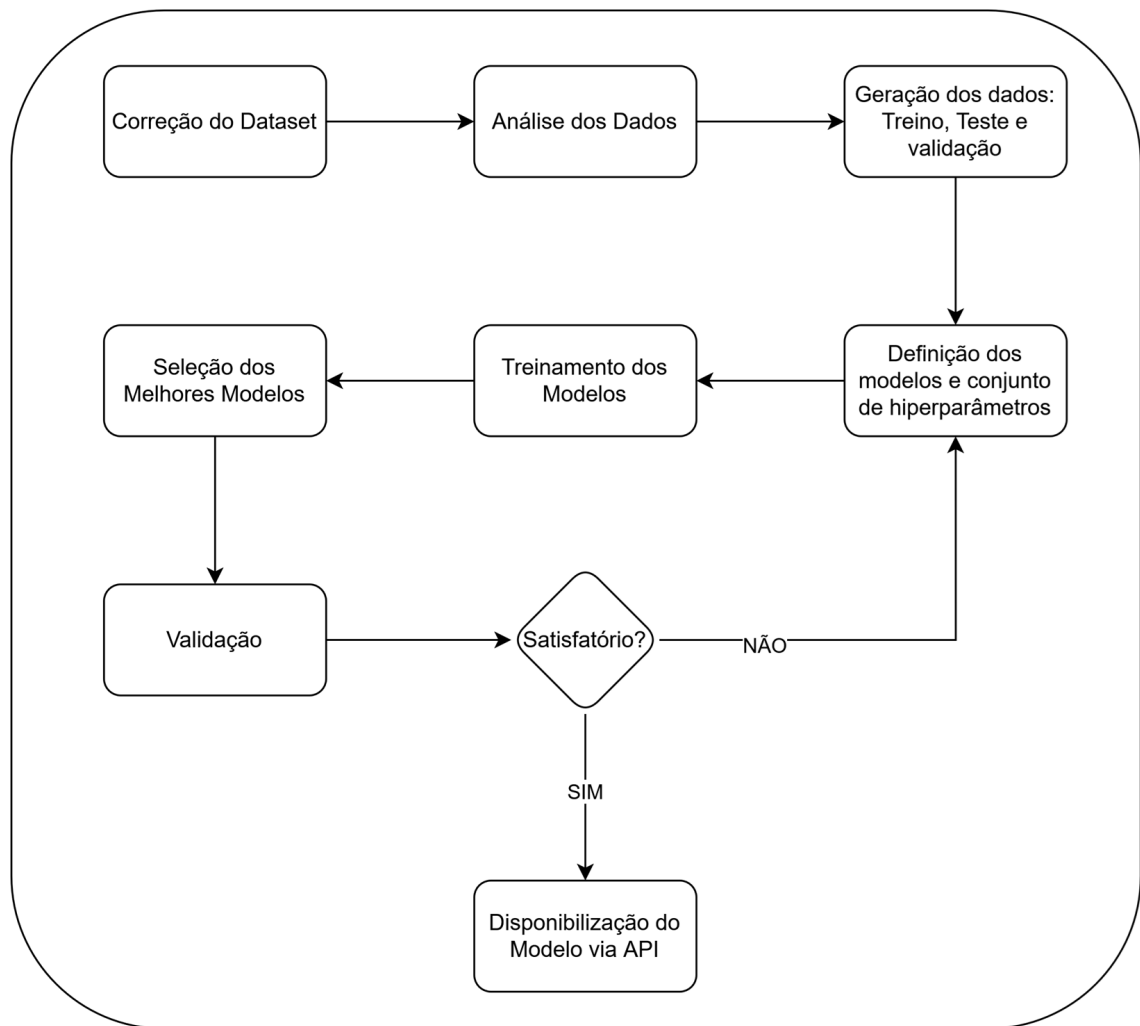
A figura 6 mostra os passos adotados neste trabalho. Primeiro, é feito a correção do dataset, como preenchimento de valores nulos, correção de separador decimal e verificação de duplicidade. Após, a análise dos dados, para tentar entender o conjunto e fazer então a separação entre treino, teste e validação. Após, é feito a seleção dos modelos que irão para treinamento e o conjunto de hiperparâmetros que serão usados para treinar o modelo e selecionar os melhores que são validados em cima do conjunto de validação. Durante a primeira interação, foi procurado o melhor conjunto de hiperparâmetros e adicionar valores próximos para então voltar ao treinamento. Após isso, foi verificado se os modelos estão tendo resultados satisfatórios, que nesse caso, é verificar se há melhorias com os novos conjuntos de hiperparâmetros ou não. Ao chegar em um ponto em que alterar os hiperparâmetros não traz mais nenhuma melhoria, sendo considerado como satisfatório e então disponibilizado o melhor modelo em uma API.

5.1 Tecnologias Usadas

Para o desenvolvimento do protótipo, serão criados 2 softwares. O primeiro, de experimentação, construído em Python, servirá para buscarmos o melhor modelo para geração da curva de escoamento, fazendo uma análise dos dados, testes de algoritmos e comparação de hiperparâmetros, tendo como resultado o melhor modelo para geração de curvas de escoamento. O segundo software é o protótipo em si, sendo uma API de predição construída utilizando o FastAPI que ira disponibilizar o modelo para o usuário. Ela receberá a composição química do material e as informações de temperatura e velocidade de deformação e retornará os pontos preditos e a curva de escoamento.

Como ambiente de desenvolvimento foi utilizado um computador com processador Ryzen 5700x, placa de vídeo nVidia RTX 3060 com 12 GB de memória, 16 GB de memória RAM DDR4 3000 GHz/s e um SSD 1TB. Tudo foi desenvolvido em ambiente Linux dentro do Windows através da tecnologia WSL (*Windows Subsystem for Linux*).

Figura 6 – Diagrama de experimentação adotada.



Fonte: Do Autor

5.1.1 Software de Experimentação

Para o software de experimentação foi utilizado a linguagem Python na versão 3.10. Também utilizamos como principais ferramentas a biblioteca tensorflow, habilitada para GPUs (CUDA), Keras, scikit-learn, matplotlib e pandas.

Para gerenciar o ambiente virtual e as bibliotecas, foi utilizado o "uv", que agrega diversas ferramentas como pip, venv e poetry, sendo escrita em linguagem Rust adicionando muito desempenho na instalação do ambiente virtual de desenvolvimento.

5.2 Dataset

O trabalho utilizou um *dataset* feito com base em testes contendo as *features* e a *label* que serão utilizadas para a regressão. Eles foram carregados em um arquivo tabular

de formato .csv, separado por vírgula, com as seguintes colunas: nome, carbono, silício, manganês, fósforo, enxofre, alumínio, cromo, níquel, molibdênio, titânio, vanádio, berílio, nitrogênio, cobre, nióbio, temperatura, velocidade de deformação, deformação e resultado. Os dados foram extraídos da biblioteca do software de simulação QForm UK.

A coluna nome é referente ao nome dado ao aço. Já as colunas carbono, silício, manganês, fósforo, enxofre, alumínio, cromo, níquel, molibdênio, titânio, vanádio, berílio, nitrogênio, cobre e nióbio dizem o quanto de cada elemento químico está presente naquele material. Temperatura e velocidade de deformação são *features* com valores variáveis relacionadas ao experimento físico do material, são valores que o operador colocou como configuração na prensa para gerar os resultados. Já a deformação é um parâmetro do experimento, sendo 8 pontos fixos em todos os experimentos. Resultado é a tensão de escoamento, ele servirá de *label*, ou alvo, que iremos predizer. As outras colunas (menos a coluna nome) servirão de dados de entrada. A cada 8 linhas (deformação) do *dataframe*, possuímos uma curva inteira, ou seja, cada curva é formada por 8 pontos. A figura 5.2 mostra um exemplo da tabela com os dados.

Figura 7 – Exemplo do *dataframe* com os dados de treino e teste.

ACO	COMPOSICAO QUIMICA DO ACO (VARIÁVEL)												TEMPERATURA	TAXA DE DEFORMACAO	DEFORMACAO	TENSÃO DE ESCOAMENTO		
	CARBONO	SILICIO	MANGANES	FOSFORO	ENXOFRE	ALUMINIO	CROMO	NIQUEL	MOIBDENIO	TITANIO	VANADIO	BERILIO	VARIÁVEL	REPETE	REPETE	RESULTADO		
X4NiCrTi25	0.06	1	2	0.025	0.015	0.35	16	27	1.5	2	0.5	0.01	900	0.01	0.03978087	80.03	TAXA DE DEFORMACAO 0,01 /s	
X4NiCrTi25	0.06	1	2	0.025	0.015	0.35	16	27	1.5	2	0.5	0.01	900	0.01	0.099820335	131.34		
X4NiCrTi25	0.06	1	2	0.025	0.015	0.35	16	27	1.5	2	0.5	0.01	900	0.01	0.199671195	150.01		
X4NiCrTi25	0.06	1	2	0.025	0.015	0.35	16	27	1.5	2	0.5	0.01	900	0.01	0.299754654	154.56		
X4NiCrTi25	0.06	1	2	0.025	0.015	0.35	16	27	1.5	2	0.5	0.01	900	0.01	0.400477567	155.78		
X4NiCrTi25	0.06	1	2	0.025	0.015	0.35	16	27	1.5	2	0.5	0.01	900	0.01	0.499226488	155.87		
X4NiCrTi25	0.06	1	2	0.025	0.015	0.35	16	27	1.5	2	0.5	0.01	900	0.01	0.599656837	155.49		
X4NiCrTi25	0.06	1	2	0.025	0.015	0.35	16	27	1.5	2	0.5	0.01	900	0.01	0.800732391	154.25		
X4NiCrTi25	0.06	1	2	0.025	0.015	0.35	16	27	1.5	2	0.5	0.01	900	0.1	0.03978087	109.43	TAXA DE DEFORMACAO 0,1 /s	
X4NiCrTi25	0.06	1	2	0.025	0.015	0.35	16	27	1.5	2	0.5	0.01	900	0.1	0.099820335	179.59		
X4NiCrTi25	0.06	1	2	0.025	0.015	0.35	16	27	1.5	2	0.5	0.01	900	0.1	0.199671195	205.13		
X4NiCrTi25	0.06	1	2	0.025	0.015	0.35	16	27	1.5	2	0.5	0.01	900	0.1	0.299754654	211.34		
X4NiCrTi25	0.06	1	2	0.025	0.015	0.35	16	27	1.5	2	0.5	0.01	900	0.1	0.400477567	213.02		
X4NiCrTi25	0.06	1	2	0.025	0.015	0.35	16	27	1.5	2	0.5	0.01	900	0.1	0.499226488	213.13		
X4NiCrTi25	0.06	1	2	0.025	0.015	0.35	16	27	1.5	2	0.5	0.01	900	0.1	0.599656837	212.61		
X4NiCrTi25	0.06	1	2	0.025	0.015	0.35	16	27	1.5	2	0.5	0.01	900	0.1	0.800732391	210.92		
X4NiCrTi25	0.06	1	2	0.025	0.015	0.35	16	27	1.5	2	0.5	0.01	900	1	0.03978087	149.64	TAXA DE DEFORMACAO 1 /s	T E M P E R A T U R A
X4NiCrTi25	0.06	1	2	0.025	0.015	0.35	16	27	1.5	2	0.5	0.01	900	1	0.099820335	245.57		
X4NiCrTi25	0.06	1	2	0.025	0.015	0.35	16	27	1.5	2	0.5	0.01	900	1	0.199671195	280.49		
X4NiCrTi25	0.06	1	2	0.025	0.015	0.35	16	27	1.5	2	0.5	0.01	900	1	0.299754654	288.99		
X4NiCrTi25	0.06	1	2	0.025	0.015	0.35	16	27	1.5	2	0.5	0.01	900	1	0.400477567	291.27		
X4NiCrTi25	0.06	1	2	0.025	0.015	0.35	16	27	1.5	2	0.5	0.01	900	1	0.499226488	291.43		
X4NiCrTi25	0.06	1	2	0.025	0.015	0.35	16	27	1.5	2	0.5	0.01	900	1	0.599656837	290.72		
X4NiCrTi25	0.06	1	2	0.025	0.015	0.35	16	27	1.5	2	0.5	0.01	900	1	0.800732391	288.41		
X4NiCrTi25	0.06	1	2	0.025	0.015	0.35	16	27	1.5	2	0.5	0.01	900	10	0.03978087	204.61	TAXA DE DEFORMACAO 10 /s	S O C I E D A D E
X4NiCrTi25	0.06	1	2	0.025	0.015	0.35	16	27	1.5	2	0.5	0.01	900	10	0.099820335	335.78		
X4NiCrTi25	0.06	1	2	0.025	0.015	0.35	16	27	1.5	2	0.5	0.01	900	10	0.199671195	383.53		
X4NiCrTi25	0.06	1	2	0.025	0.015	0.35	16	27	1.5	2	0.5	0.01	900	10	0.299754654	395.16		
X4NiCrTi25	0.06	1	2	0.025	0.015	0.35	16	27	1.5	2	0.5	0.01	900	10	0.400477567	398.28		
X4NiCrTi25	0.06	1	2	0.025	0.015	0.35	16	27	1.5	2	0.5	0.01	900	10	0.499226488	398.5		
X4NiCrTi25	0.06	1	2	0.025	0.015	0.35	16	27	1.5	2	0.5	0.01	900	10	0.599656837	397.53		
X4NiCrTi25	0.06	1	2	0.025	0.015	0.35	16	27	1.5	2	0.5	0.01	900	10	0.800732391	394.37		
X4NiCrTi25	0.06	1	2	0.025	0.015	0.35	16	27	1.5	2	0.5	0.01	900	50	0.03978087	254.63	TAXA DE DEFORMACAO 50 /s	O C I E D A D E
X4NiCrTi25	0.06	1	2	0.025	0.015	0.35	16	27	1.5	2	0.5	0.01	900	50	0.099820335	417.87		
X4NiCrTi25	0.06	1	2	0.025	0.015	0.35	16	27	1.5	2	0.5	0.01	900	50	0.199671195	477.29		
X4NiCrTi25	0.06	1	2	0.025	0.015	0.35	16	27	1.5	2	0.5	0.01	900	50	0.299754654	491.76		
X4NiCrTi25	0.06	1	2	0.025	0.015	0.35	16	27	1.5	2	0.5	0.01	900	50	0.400477567	495.65		
X4NiCrTi25	0.06	1	2	0.025	0.015	0.35	16	27	1.5	2	0.5	0.01	900	50	0.499226488	495.92		
X4NiCrTi25	0.06	1	2	0.025	0.015	0.35	16	27	1.5	2	0.5	0.01	900	50	0.599656837	494.71		
X4NiCrTi25	0.06	1	2	0.025	0.015	0.35	16	27	1.5	2	0.5	0.01	900	50	0.800732391	490.78		
X4NiCrTi25	0.06	1	2	0.025	0.015	0.35	16	27	1.5	2	0.5	0.01	900	100	0.03978087	279.78	TAXA DE DEFORMACAO 100 /s	
X4NiCrTi25	0.06	1	2	0.025	0.015	0.35	16	27	1.5	2	0.5	0.01	900	100	0.099820335	459.14		
X4NiCrTi25	0.06	1	2	0.025	0.015	0.35	16	27	1.5	2	0.5	0.01	900	100	0.199671195	524.44		
X4NiCrTi25	0.06	1	2	0.025	0.015	0.35	16	27	1.5	2	0.5	0.01	900	100	0.299754654	540.33		
X4NiCrTi25	0.06	1	2	0.025	0.015	0.35	16	27	1.5	2	0.5	0.01	900	100	0.400477567	544.6		
X4NiCrTi25	0.06	1	2	0.025	0.015	0.35	16	27	1.5	2	0.5	0.01	900	100	0.499226488	544.9		
X4NiCrTi25	0.06	1	2	0.025	0.015	0.35	16	27	1.5	2	0.5	0.01	900	100	0.599656837	543.58		
X4NiCrTi25	0.06	1	2	0.025	0.015	0.35	16	27	1.5	2	0.5	0.01	900	100	0.800732391	539.25		

Fonte: Do Autor

O primeiro passo dos experimentos foi garantir que não houvessem valores nulos ou em formato de texto. Os dados estavam relativamente limpos e corretos, exceto para 2 experimentos (16 linhas) que estavam em branco para nióbio. Como os experimentos eram repetidos para diferentes temperaturas e velocidade de deformação, ao analisar o *dataset* vimos que os valores corretos era 0, com isso preenchemos para não perder a informação.

Para criação da *pipeline* de experimentação, foram criadas funções tanto para

modelos do *scikit-learn* quanto para modelos do *tensorflow*.

5.3 Análise dos Dados e Dataset de Validação

Para garantir que o modelo se comporte bem em dados nunca vistos, foi criado um *dataset* de validação que foi separado do restante dos dados. Como uma curva é um conjunto de 8 pontos (nesse caso, 8 linhas seguidas da planilha), sendo criado uma função que fazia a separação por curvas, e não pontos. Isto foi feito para garantir que, primeiro, as curvas não teriam pontos quebrados, ou seja, pontos faltantes. Outro motivo foi para analisar o quão equilibrado estavam as *features*.

Um ponto importante foi que, durante essa análise, foi possível notar que alguns elementos químicos eram mais comuns de estar presentes na composição do metal, enquanto outros eram "raros". Vale salientar que a palavra "raro" neste contexto não quer dizer que o elemento era raro ou que ele aparecia muito poucas vezes. Foi adotado essa nomenclatura somente como um *threshold* para dizer se os elementos apareciam em menos de "X%" das amostras, sendo X um valor arbitrário definido pelo autor. Para o trabalho, foi definido que caso um elemento aparecesse em menos de 50% das amostras, eles eram raros. Também foi definido, de forma arbitrária, que seria usado 15% dos dados como um *dataset* de validação.

A tabela 2 mostra a quantidade de vezes que cada elemento aparece no *dataset*. A coluna "Elemento" é o nome do elemento, "# de Pontos" é a quantidade pontos esse elemento aparece com valor superior a zero, "Porcentagem Total" nos dá a porcentagem em relação a quantidade total de pontos e a "Raridade no Dataset" é como é chamado esse elemento para fins de separação para validação.

Outros dois pontos de atenção em relação aos dados originais foram que, na planilha original, a separação decimal estava no padrão brasileiro, ou seja, separado por vírgula. Isso quebra a pipeline, já que as bibliotecas de utilizadas adotavam o padrão de separação por ponto (.). Para evitar problemas, foi realizado a conversão de vírgula para ponto.

Também para adotar padrões e garantir a qualidade dos modelos, evitando que características que possuísem valores muito grandes acabassem levando o modelo para possível *overfitting*, sendo feita a normalização dos dados usando o *StandardScaler* da biblioteca *scikit-learn*.

Para separar os dados que seriam usados para validação após a seleção e treino do modelo, foi preciso garantir que pelo menos um número de elementos raros mínimos aparecesse. Com base nisso, foi definido a necessidade de pelo menos 2 exemplos de cada elemento. Além dessa condição, a base de dados foi estratificada por material, garantido que os dados de validação possuísem a mesma distribuição de materiais dos dados de

Tabela 2 – Quantidade de elementos presentes no *dataset*. Fonte: do Autor

Elemento	# de Pontos	Porcentagem do Total	Raridade no <i>Dataset</i>
Carbono	2995	100,0%	Comum
Silício	2995	100,0%	Comum
Manganês	2995	100,0%	Comum
Fósforo	2995	100,0%	Comum
Enxofre	2995	100,0%	Comum
Alumínio	480	16,0%	Raro
Cromo	2415	80,6%	Comum
Níquel	1310	43,7%	Raro
Molibdênio	1405	46,9%	Raro
Titânio	275	9,2%	Raro
Vanádio	750	25,0%	Raro
Berílio	60	2,0%	Raro
Nitrogênio	385	12,9%	Raro
Cobre	410	13,7%	Raro
Nióbio	180	6,9%	Raro

treino.

Para finalizar essa etapa, foi feita a validação das curvas separadas. Cada curva foi analisada para garantir que possuía os 8 pontos, evitando a sua quebra e perda de informação. Também foi confirmado a presença dos materiais raros com base nos critérios citados acima e checado se não havia nenhum ponto presente tanto no *dataset* de validação quanto no que iríamos utilizar para treino e teste. Lembrando que foram testados 2 formas de separar os dados, utilizando `train_test_split` e validação cruzada. Ao final, foi feito a seleção com a validação cruzada por trazer melhores resultados para os experimentos.

5.3.1 Separação em Treino e Teste

A separação em dados de treino e teste é feita para que os modelos de inteligência artificial tirem as métricas em cima de dados que não foram usados para ajustar seus parâmetros.

Nessa etapa já foram utilizados os dados que foram separados na etapa anterior. Uma forma comum para separar os dados, é utilizando a função `train_test_split` do `scikit-learn`. Ela cria um conjunto de dados de treino e teste estratificado, separando os pontos de forma aleatória. Para esse trabalho, isso poderia ser um problema pois cada 8 pontos sequenciais representa uma curva, e queríamos evitar de separar seus pontos. Para resolver esse problema, foi realizado a separação de forma aleatória por curva, ou seja, os pontos não seriam divididos.

Essa estratégia foi feita junto com a validação cruzada nos testes iniciais, porém não fazia sentido usá-la sendo que tínhamos já uma implementação de validação cruzada,

sendo abandonada.

5.4 Estratégia de Avaliação

Para avaliar os modelos, adotamos a estratégia de validação cruzada, também chamada de *k-fold*. Assim como na divisão de treino, validação e teste, precisávamos ter cuidado em relação aos pontos que formam uma curva.

Criamos um *k-fold* customizado que respeitava os limites das curvas, ao contrário do tradicional que não iria levar em conta esse caso. Essa estratégia garantiu que as curvas inteiras ficassem em cada *fold*. Como configuração, usamos um *k* de valor 5.

Como resumo, foi feita uma avaliação da seguinte forma: para a avaliação do modelo, os dados de treino e validação e, para a otimização dos hiperparâmetros foram separados, foi então utilizado a validação cruzada em cima dos dados de treino que foram separados na etapa anterior.

5.5 Seleção de Modelos

Para selecionar os modelos, foi definido um espaço de busca tanto com modelos tradicionais quanto com modelos profundos. Para este trabalho, foram utilizados *Random Forest*, *XGBoost*, *Gradient Boosting*, *SVR* e redes neurais rasas e profundas. A ideia foi utilizar diferentes paradigmas de aprendizado, passando por regressores, ensembles até redes neurais.

Para o treinamento de redes neurais, foi adotada duas estratégias. Na primeira, usamos o próprio *scikit-learn* para treinar um MLP com alguns hiperparâmetros simples e ver os resultados. Porém, o *scikit-learn* não permite uma grande liberdade para personalização da arquitetura, por exemplo, a definição de camadas e neurônios eram feitas por tuplas, no estilo (2, 25). Já no *Tensorflow*, conseguimos separar o número de camadas do número de neurônios, com isso o espaço de busca aumenta. Também conseguimos, pelo *Tensorflow*, fazer com que o número de épocas fosse um hiperparâmetro a ser buscado, junto com o *Dropout*.

5.6 Otimização dos Hiperparâmetros

Para selecionar o melhor conjunto de hiperparâmetros, foi necessário adotar duas abordagens diferentes. Para os modelos clássicos, treinados utilizando a biblioteca do *scikit-learn*, foi usado o *Optuna* para achar o melhor conjunto possível. O *Optuna* é uma biblioteca que utiliza diversos algoritmos de busca para encontrar a melhor combinação de hiperparâmetros dentro do espaço de busca definido. Nela, foram definidas sugestões

de valores de busca e a quantidade de buscas a serem feitas. Com base em modelos estatísticos, ele vai selecionando as melhores combinações para o modelo.

Já para os modelos de redes neurais rasos e profundos que foram treinados utilizando a biblioteca to Tensorflow, foi adotada uma estratégia clássica de busca conhecida como busca aleatória. Nela, definimos uma quantidade de valores para busca, um número de vezes que iremos rodar essa busca e a busca aleatória vai, como diz o nome, aleatoriamente selecionar os conjuntos, treinar o modelo, verificar o resultado e salvar os que forem melhor.

Essa divisão foi necessária devido a uma limitação de hardware. A busca via Optuna para os modelos treinados usando o Tensorflow estava estourando a memória, fazendo o processo ser morto pelo sistema operacional. Isso foi uma limitação desse projeto. Como configuração dos experimentos para os experimentos com scikit-learn, foram:

- **Decision Tree Regressor (DecisionTreeRegressor):**

- **Espaço de hiperparâmetros:**

- * max_depth: [5, 10, 15, 20, 25, 30]
 - * min_samples_split: [2, 5, 10, 15]
 - * min_samples_leaf: [1, 2, 4, 8, 12]
 - * max_features: [0.3, 0.5, 0.7, "sqrt", "log2"]
 - * criterion: ["squared_error", "friedman_mse", "absolute_error"]
 - * splitter: ["best", "random"]
 - * ccp_alpha: [0.0, 0.01, 0.1]

- **Hiperparâmetros base:**

- * max_depth: 10
 - * min_samples_split: 2
 - * min_samples_leaf: 1
 - * max_features: "sqrt"
 - * criterion: "squared_error"
 - * splitter: "best"
 - * ccp_alpha: 0.0

- **Random Forest Regressor (RandomForestRegressor):**

- **Espaço de hiperparâmetros:**

- * n_estimators: [100, 200, 500, 1000]
 - * max_depth: [10, 20, 30, 50]
 - * min_samples_split: [2, 5, 10]

- * min_samples_leaf: [1, 2, 4, 8]
- * max_features: [0.3, 0.5, 0.7]
- * bootstrap: [True, False]

– **Hiperparâmetros base:**

- * n_estimators: 200
- * max_depth: 20
- * min_samples_split: 5
- * min_samples_leaf: 2
- * max_features: "sqrt"
- * bootstrap: True
- * n_jobs: -1

• **XGBoost Regressor (XGBRegressor):**

– **Espaço de hiperparâmetros:**

- * n_estimators: [100, 200, 500, 1000]
- * max_depth: [3, 5, 7, 9, 12]
- * learning_rate: [0.001, 0.01, 0.05, 0.1, 0.3]
- * subsample: [0.6, 0.8, 1.0]
- * colsample_bytree: [0.6, 0.8, 1.0]
- * gamma: [0, 0.1, 0.3, 0.5]
- * min_child_weight: [1, 3, 5, 7]
- * reg_alpha: [0, 0.01, 0.1, 1]
- * reg_lambda: [0, 0.01, 0.1, 1]

– **Hiperparâmetros base:**

- * n_estimators: 200
- * max_depth: 5
- * learning_rate: 0.1
- * subsample: 0.8
- * colsample_bytree: 0.8
- * gamma: 0
- * min_child_weight: 3
- * reg_alpha: 0
- * reg_lambda: 0.1
- * n_jobs: -1
- * tree_method: "hist"

- * device: "cuda"

- **Support Vector Regressor (SVR):**

- **Espaço de hiperparâmetros:**

- * kernel: ["rbf", "poly"]
 - * C: [0.1, 1, 10, 100, 1000]
 - * gamma: [0.001, 0.01, 0.1, 1, "scale", "auto"]
 - * epsilon: [0.001, 0.01, 0.1, 0.5, 1.0]
 - * degree: [2, 3, 4]

- **Hiperparâmetros base:**

- * kernel: "rbf"
 - * C: 100
 - * gamma: "scale"
 - * epsilon: 0.1

- **Gradient Boosting Regressor (GradientBoostingRegressor):**

- **Espaço de hiperparâmetros:**

- * n_estimators: [100, 200, 500]
 - * learning_rate: [0.01, 0.05, 0.1, 0.2]
 - * max_depth: [3, 5, 7, 9]
 - * min_samples_split: [2, 5, 10]
 - * min_samples_leaf: [1, 2, 4]
 - * subsample: [0.6, 0.8, 1.0]
 - * max_features: [0.3, 0.5, 0.7, "sqrt"]

- **Hiperparâmetros base:**

- * n_estimators: 200
 - * learning_rate: 0.1
 - * max_depth: 5
 - * min_samples_split: 5
 - * min_samples_leaf: 2
 - * subsample: 0.8
 - * max_features: "sqrt"

- **Rede Neural Artificial (MLPRegressor):**

- **Espaço de hiperparâmetros:**

```

* hidden_layer_sizes: [ [64], [128], [64, 32], [128, 64] ]
* activation: ["relu"]
* solver: ["adam"]
* alpha: [0.0001, 0.001, 0.01]
* learning_rate: ["constant"]
* learning_rate_init: [0.001, 0.01]

```

– **Hiperparâmetros base:**

```

* hidden_layer_sizes: [128, 64]
* activation: "relu"
* solver: "adam"
* alpha: 0.001
* learning_rate: "constant"
* learning_rate_init: 0.01
* max_iter: 2000
* early_stopping: True

```

O espaço foi definido dessa forma para ser rodado tanto em *grid* ou *random search* quanto no Optuna. Para a busca de modelos usando Tensorflow, o espaço de busca definido foi:

- **Rede Neural Rasa (Shallow MLP):**

– Espaço de hiperparâmetros:

```

* hidden_layers: [1, 2]
* units: [4, 8, 16, 32]
* activation: ["relu", "tanh"]
* dropout_rate: [0.0, 0.1]
* learning_rate: [0.01, 0.001],
* epochs: [500, 1000, 2000]

```

– Hiperparâmetros base:

```

* hidden_layers=1
* units=4
* activation="relu"
* dropout_rate=0.0
* learning_rate=0.01

```

- **Rede Neural Profunda (Deep MLP):**

- Espaço de hiperparâmetros:
 - * `hidden_layers`: [3, 4, 5]
 - * `units`: [4, 8, 16, 32]
 - * `activation`: ["relu", "tanh"]
 - * `dropout_rate`: [0.0, 0.1]
 - * `learning_rate`: [0.01, 0.001]
 - * `epochs`: [500, 1000, 2000]
- Hiperparâmetros base:
 - * `hidden_layers=3`
 - * `units=4`
 - * `activation="relu"`
 - * `dropout_rate=0.0`
 - * `learning_rate=0.01`

Esses valores foram arbitrários e definidos após serem testados diversos valores diferente. Esses espaços de busca foram os que trouxeram melhores resultados tanto em eficiência de memória quanto em erro médio.

5.7 Métricas de Avaliação

Todos os modelos, independente se foram treinados pelo scikit-learn ou pelo Tensorflow, utilizaram as mesmas métricas de avaliação para o treino. Foram analisadas as médias do erro absoluto e a raiz do erro médio quadrado (MAE e RMSE), o coeficiente de determinação R^2 para a análise dos pontos e o Erro Percentual Absoluto Médio (MAPE). O MAE diz o erro absoluto, ou seja a diferença entre o que foi predito e o valor real. Já o RMSE, ignora se os erros são positivos ou negativos. O R^2 diz a proporção da variância dos dados, variando de 0 a 1, nesse caso, podemos ler que quanto mais perto de 1, melhor é o modelo. Já o MAPE é o percentual do MAE.

Para a avaliação em cima dos dados de validação, foram utilizados as mesmas métricas, porém agora em cima das curvas. Essa estratégia foi utilizada pois, apesar de termos uma predição de pontos, o que queremos ao final é saber o quão uma curva se aproxima e, para este trabalho, uma curva é um conjunto de 8 pontos. Utilizar essas duas abordagens garante não somente que estamos acertando os pontos, mas que também o sistema gere curvas que tenham um formato que seja fisicamente aceitável e parecidas com as curvas reais.

Para o erro, foi feito uma análise a nível geral e também uma análise por temperatura e taxa de tensão. As mesmas métricas citadas acima foram utilizadas para gerar gráficos a fim de entender o comportamento do modelo para esses 2 parâmetros.

5.8 Apresentação do Sistema ao Usuário

Como forma do usuário interagir com o sistema, foi criado uma API usando a biblioteca do Python FastAPI. Ela é uma API simples que conta com 2 *endpoints*, um para prever pontos com base em um vetor e outra que recebe os 8 vetores e prediz a curva em si, retornando os 8 pontos que formam a curva e um gráfico da curva de escoamento em formato png. Os *endpoints* são acessados, se rodado em máquina local, pelos caminhos localhost/predict e localhost/predict_curve, para prever os pontos e a curva respectivamente.

A API precisa basicamente de duas coisas para funcionar, o modelo, que pode ser tanto um modelo scikit-learn quanto um modelo em Tensorflow, e um binário no formato pickle contendo o *scaler* dos dados. Como utilizamos a técnica de normalização na hora do treinamento, o modelo normalizador precisa ser o mesmo que foi utilizado para o treino dos modelos. Neste trabalho, ambos são salvos durante o *fine-tuning* dos hiperparâmetros. Essa API foi consumida por uma aplicação em Streamlit.

6 Resultados e Avaliações

Este presente capítulo é dedicado a análise dos dados coletados durante todo experimento, a busca de hiperparâmetros com base no espaço de busca apresentado no Capítulo 5. Será feita a análise dos resultados gerados com base nas métricas definidas e selecionando o melhor par de modelo e hiperparâmetros para a geração de uma curva de escoamento. Para isso, o trabalho apresentará duas seções, uma chamada de "Modelos Tradicionais", onde será comparado os modelos do scikit-learn, e outra de "Modelos Rasos e Profundos", para os modelos treinados pelo Tensorflow. Ao final, uma comparação do melhor de ambos os casos para determinar o comportamento de cada um para a tarefa apresentada.

Foi realizado uma análise descritiva com a MAE (erro absoluto médio), RMSE (raíz do erro quadrático médio) e R^2 . Com isso, iremos determinar os melhores modelos pelo erro médio em cima dos dados de validação. Também foi feito uma análise descritiva utilizando a média do erro, desvio padrão e erro máximo e mínimo para os melhores modelos. Uma análise gráfica foi utilizada para entender o comportamento do modelo, calculando a distribuição do erro por temperatura e velocidade de deformação.

6.1 Modelos Tradicionais

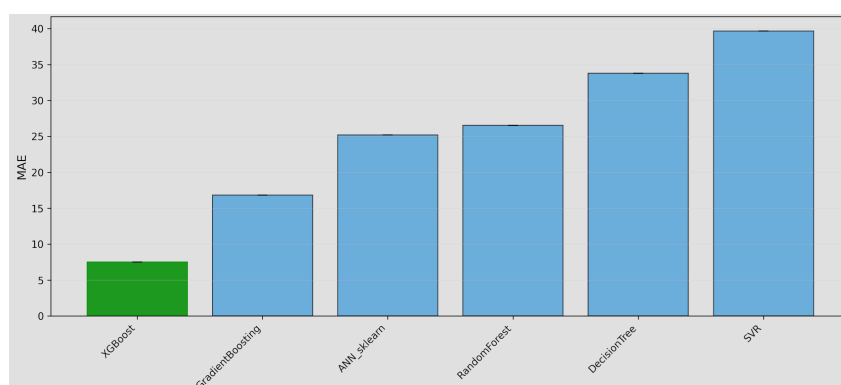
Nesta seção, foi feito uma análise para os modelos de Árvore de Decisão (Decision Trees), *Random Forest*, GradientBoosting, XGBoost e modelos simples de MLP (chamados aqui de ANN_sklearn) e SVR. A figura 8 mostra a comparação do resultado desses modelos pela MAE. Podemos ver por ela que o XGBoost apresentou um erro médio absoluto muito menor que os outros modelos. A tabela 3 mostra as métricas para cada modelo. Podemos ver que o melhor modelo, XGBoost, foi aproximadamente 2,4 vezes melhor do que o segundo colocado, com um erro de 7,5291 contra 16,8115. O pior modelo avaliado foi o SVR, com erro absoluto médio de 39,6908, um erro médio de 5,27 vezes maior que o melhor modelo. As figuras 8 e 9 mostram os modelos lado a lado utilizando as métricas de MSE, RMSE e R^2 .

Tabela 3 – Comparativo dos modelos.

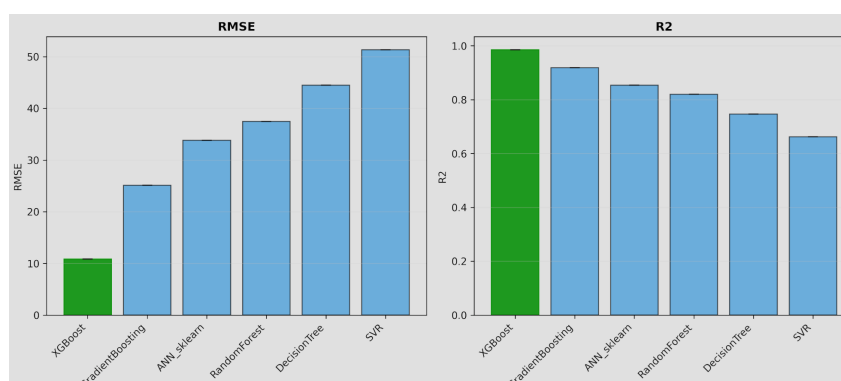
Modelo	Rank	MAE	RMSE	R ²
XGBoost	1	7,5291	10,8481	0,9849
GradientBoosting	2	16,8115	25,1401	0,9190
ANN_sklearn	3	25,2146	33,7989	0,8537
RandomForest	4	26,5544	37,4806	0,8200
DecisionTree	5	33,7834	44,4706	0,7467
SVR	6	39,6908	51,3035	0,6628

Fonte: Do Autor.

Figura 8 – Comparação do erro médio absoluto entre os modelos. Em verde, o melhor modelo.



Fonte: Do Autor.

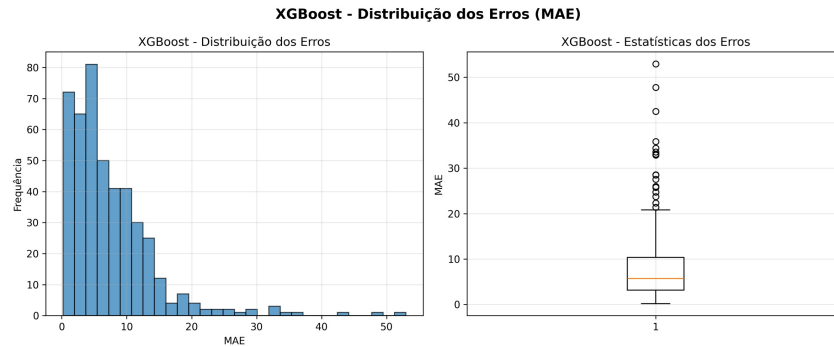
Figura 9 – Comparação da raiz do erro médio quadrático e do R² entre os modelos. Em verde, o melhor modelo.

Fonte: Do Autor.

6.1.1 Distribuição de erros

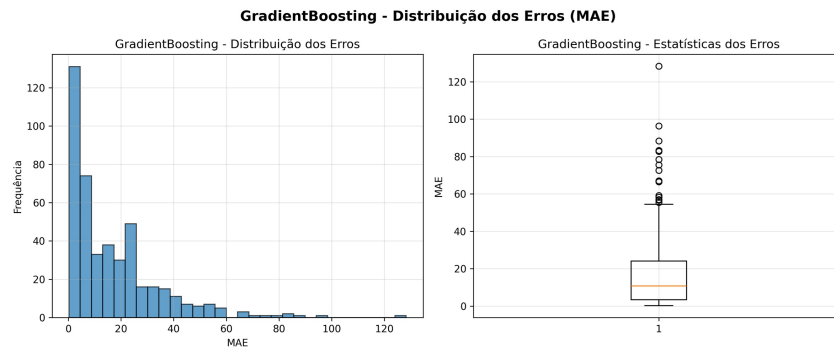
A distribuição de erros mostra onde se concentra os maiores erros. Para tal tarefa, foi feito um histograma com 30 bins, ou "colunas". Também foi feito um diagrama de caixa

Figura 10 – Distribuição e Diagrama de Caixa do erro médio absoluto para o modelo de XGBoost.



Fonte: Do Autor.

Figura 11 – Distribuição e Diagrama de Caixa do erro médio absoluto para o modelo de GradientBoosting.



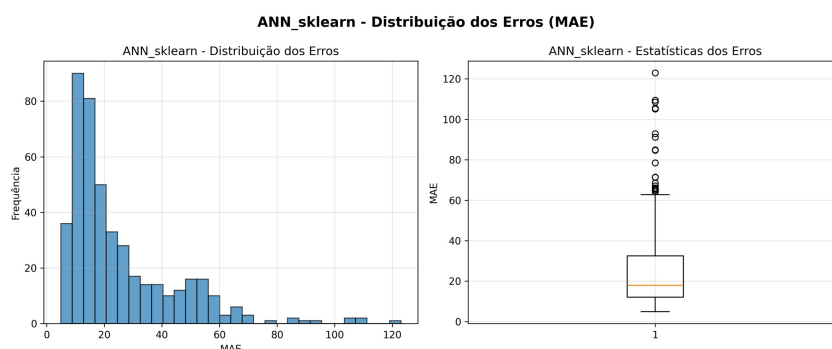
Fonte: Do Autor.

para cada modelo para entender a concentração dos erros e possíveis pontos fora da curva.

Foi possível notar que todos os modelos são assimétricos a direita, concentrando a maior quantidade de erros nos erros menores. Isso é mais presente nos melhores modelos, já que os modelos de Árvore de Decisão e SVR acabaram errando também nos valores médios. Pelo diagrama de caixa, também vemos que todos os erros maiores dos modelos acabaram se enquadrando como pontos fora da curva, tendo a média sendo próximo dos valores de erro mais baixos.

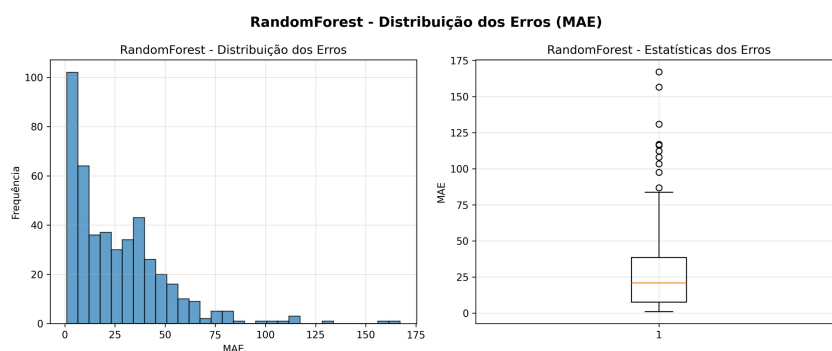
Outra análise feita, foi utilizando o RMSE e o MAPE para cada modelo. Em ambos os casos, o mesmo comportamento ocorre, como esperado, com erros bem concentrados nos menores valores, principalmente para o XGBoost, exceto para os modelos de SVR e Árvore de Decisão, que possuem bastante erros na parte intermediária da curva. Todos os gráficos confirmam que o XGBoost acabou sendo superior aos outros modelos comparados com uma boa margem.

Figura 12 – Distribuição e Diagrama de Caixa do erro médio absoluto para o modelo de MLP do scikit-learn.



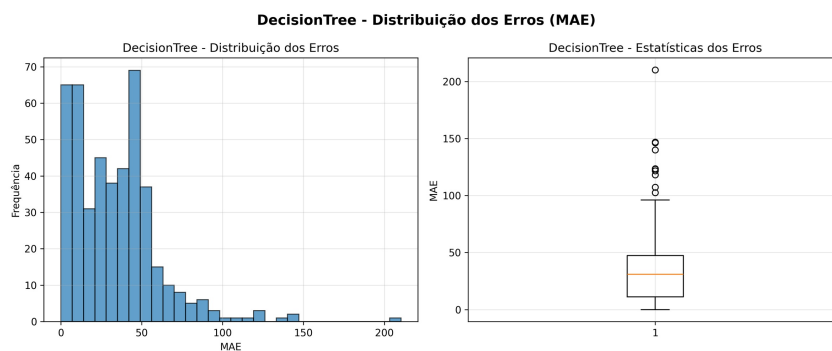
Fonte: Do Autor.

Figura 13 – Distribuição e Diagrama de Caixa do erro médio absoluto para o modelo de *Random Forest*.



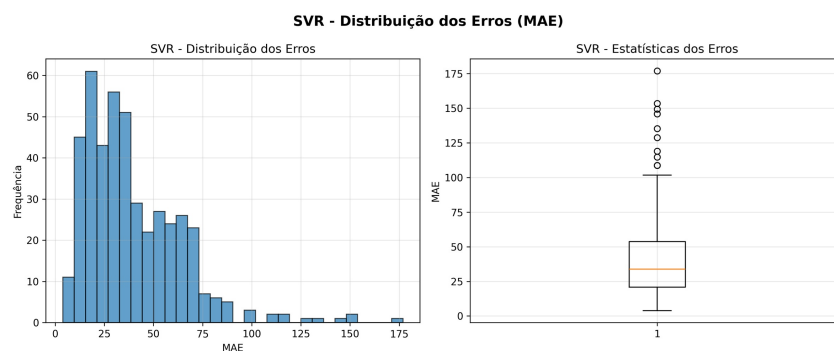
Fonte: Do Autor.

Figura 14 – Distribuição e Diagrama de Caixa do erro médio absoluto para o modelo de Árvore de Decisão.



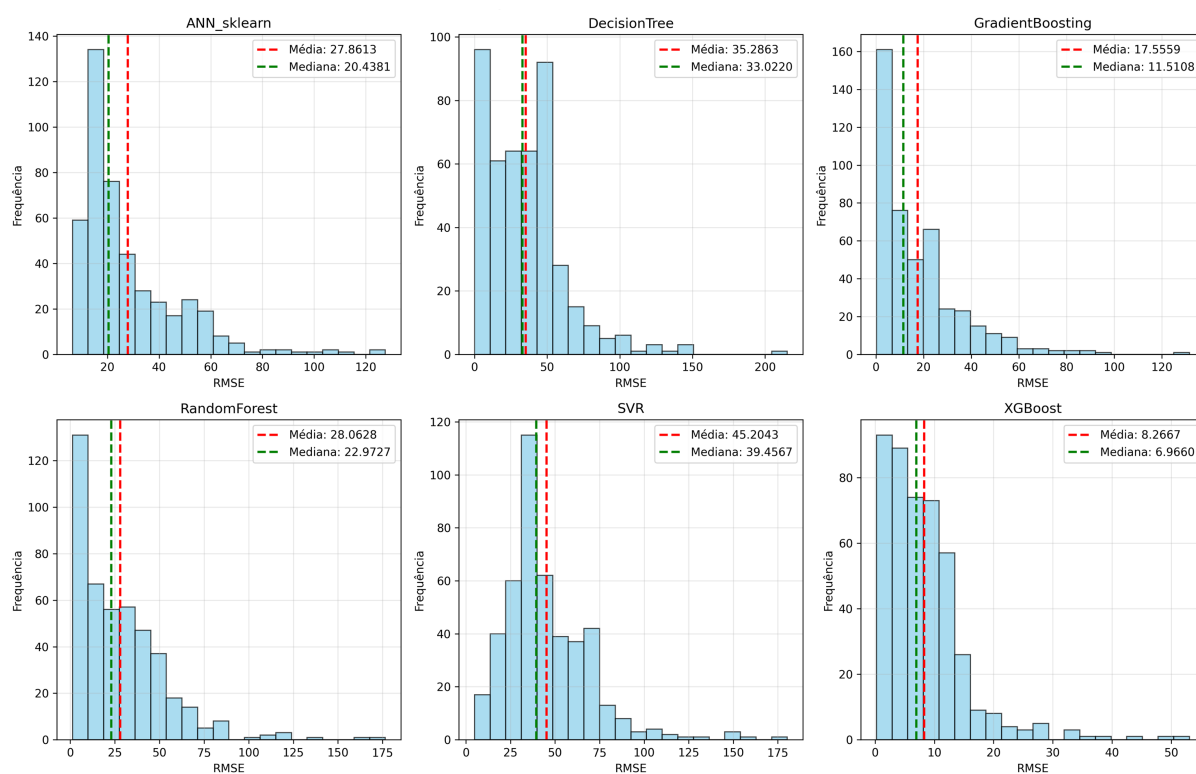
Fonte: Do Autor.

Figura 15 – Distribuição e Diagrama de Caixa do erro médio absoluto para o modelo de SVR.



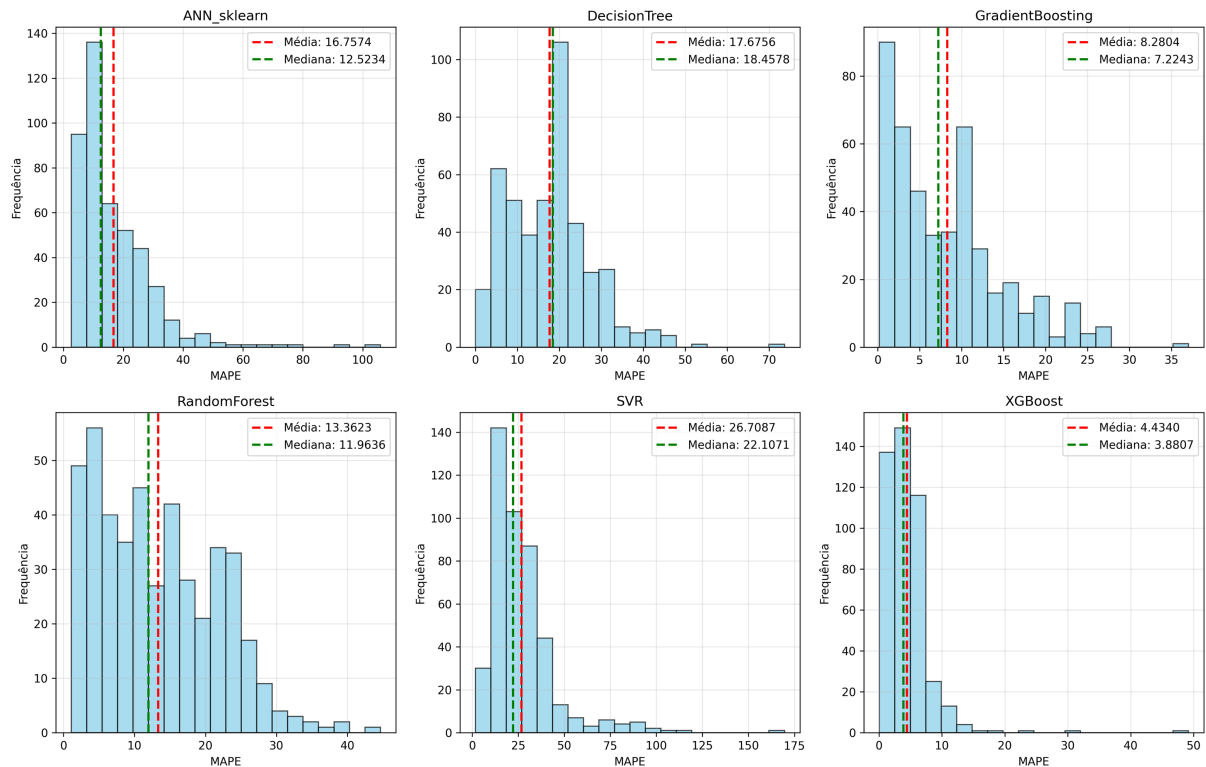
Fonte: Do Autor.

Figura 16 – Distribuição do RMSE dos modelos.



Fonte: Do Autor.

Figura 17 – Distribuição do MAPE dos modelos.



Fonte: Do Autor.

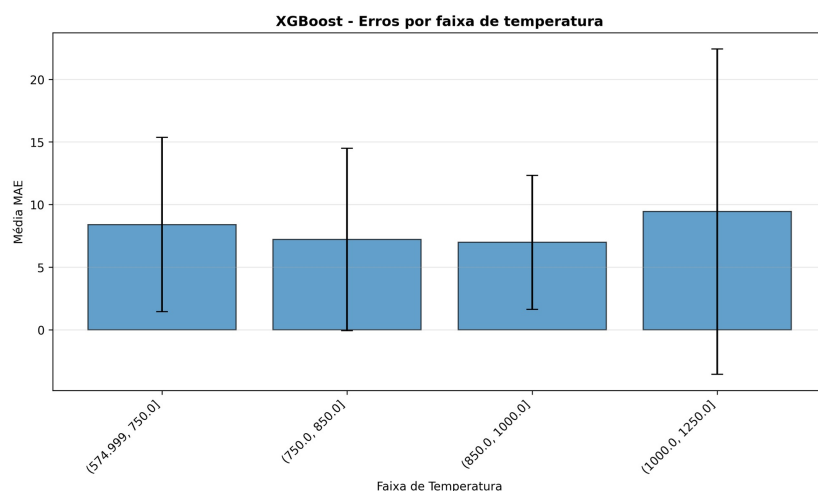
Apesar da média dar uma boa noção de como o modelo se comporta, foi preciso fazer uma análise mais detalhada de cada modelo por temperatura e velocidade de deformação para cada modelo para tentar identificar os casos que os modelos melhor conseguem prever e os que pior conseguem. Eles fornecem uma ideia de que faixa de trabalho o modelo tem uma maior confiança de acerto.

6.1.2 Erro por Temperatura

Para a análise de temperatura, foi utilizado um gráfico de barras para o erro médio e adicionado o desvio padrão para cada faixa de temperatura para cada modelo. Com isso, foi possível buscar informações para responder "se a temperatura afetava o modelo" e "o quão consistente cada modelo é para cada faixa de temperatura".

A figura 18 mostra que para as 3 primeiras faixas de temperatura, o modelo se comporta de forma muito similar, porém, para a temperatura de 1250, ele costuma errar mais e possui um desvio padrão também maior.

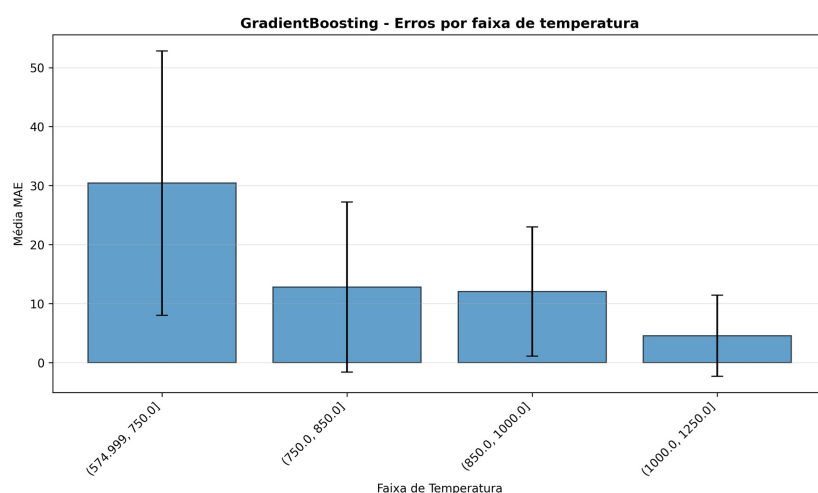
Figura 18 – Diagrama de barra mostrando a média do erro absoluto com o desvio padrão para o modelo XGBoost.



Fonte: Do Autor.

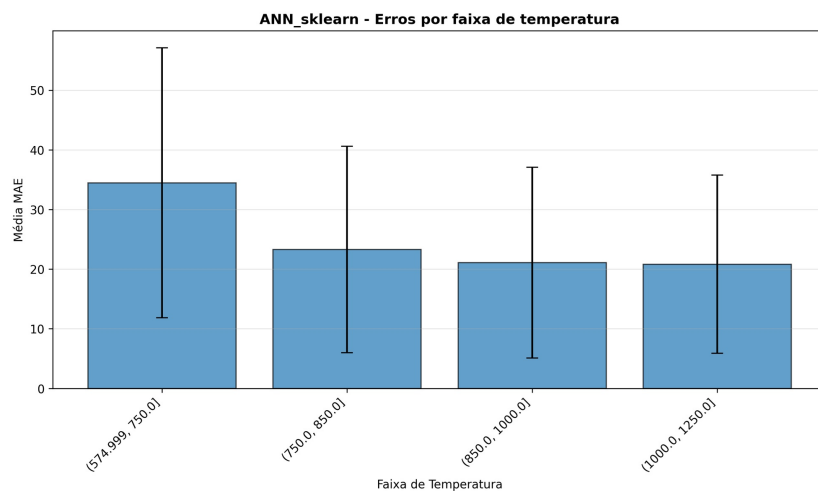
Já a figura 19 mostra que o modelo de *Gradient Boosting* foi muito melhor na faixa de temperatura em que o XGBoost mais teve dificuldade, tendo também uma boa performance nas faixas intermediárias, porém para temperaturas baixas, o modelo acabou errando muito. Esse comportamento foi muito similar ao MLP do scikit-learn, a *Random Forest* e Árvore de decisão, como mostram, respectivamente, as figuras 20, 21 e 22.

Figura 19 – Diagrama de barra mostrando a média do erro absoluto com o desvio padrão para o modelo *Gradient Boosting*.



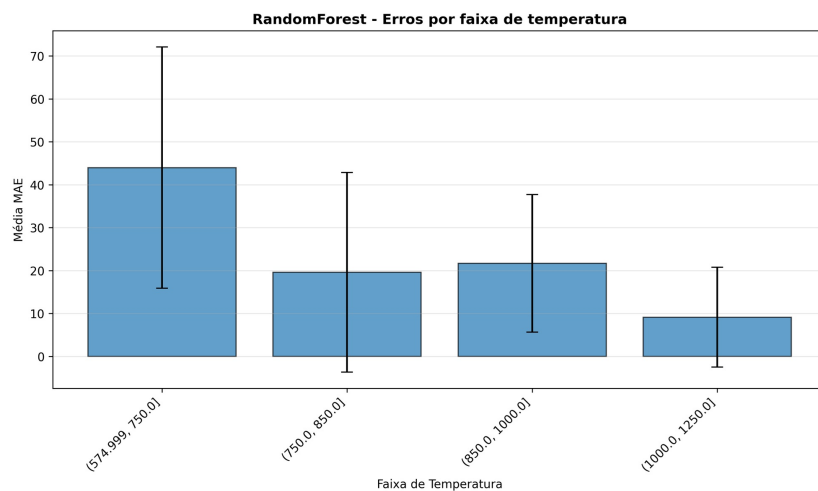
Fonte: Do Autor.

Figura 20 – Diagrama de barra mostrando a média do erro absoluto com o desvio padrão para o modelo MLP do scikit-learn.



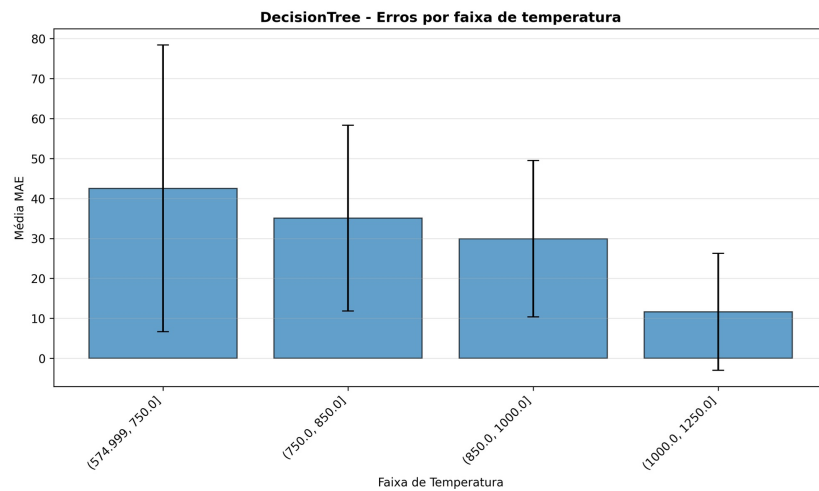
Fonte: Do Autor.

Figura 21 – Diagrama de barra mostrando a média do erro absoluto com o desvio padrão para o modelo de *Random Forest*.



Fonte: Do Autor.

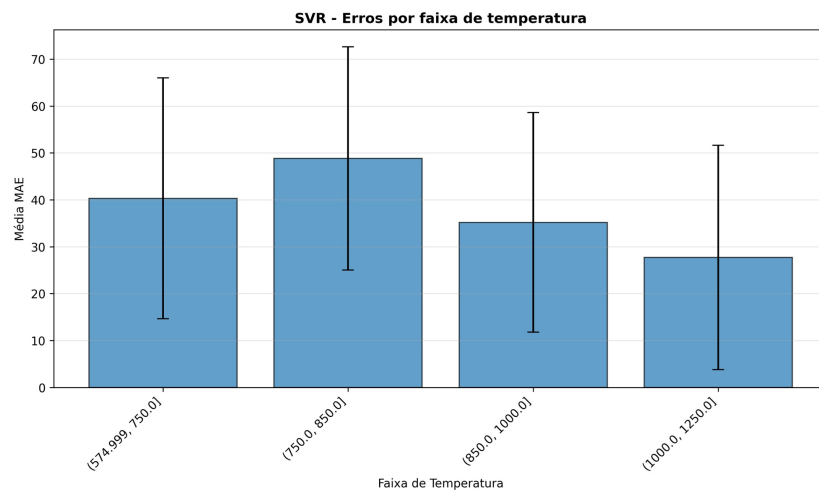
Figura 22 – Diagrama de barra mostrando a média do erro absoluto com o desvio padrão para o modelo de Árvore de Decisão.



Fonte: Do Autor.

Por final, o SVR se comportou muito mal em todas as faixas de temperatura. Tanto seu erro médio quanto o desvio padrão foi superior aos outros modelos e muito superior ao XGBoost.

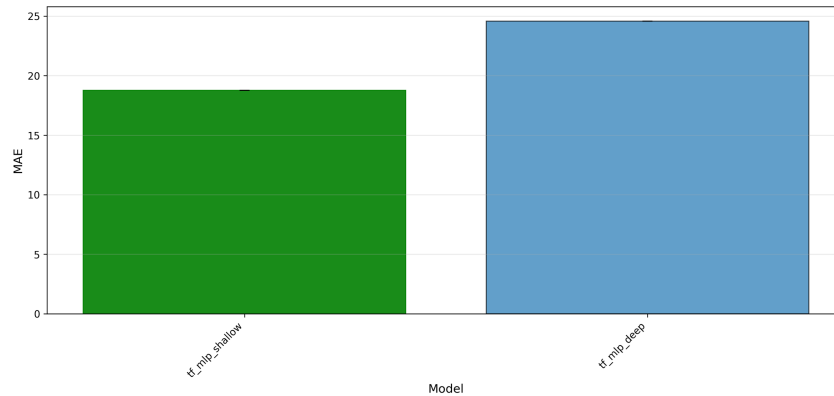
Figura 23 – Diagrama de barra mostrando a média do erro absoluto com o desvio padrão para o modelo SVR.



Fonte: Do Autor.

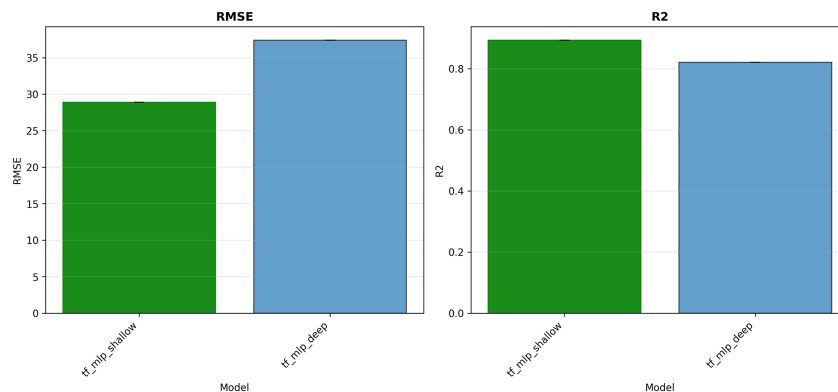
Como conclusão dessa análise, temos que, primeiro, o XGBoost foi muito consistente em todas as faixas de temperatura. Apesar de um desvio padrão alto para a faixa mais alta, ainda assim ficou dentro quando comparado aos outros modelos. Podemos também ver que a temperatura pode influenciar alguns modelos, mas essa conclusão carece de mais estudos somente sobre esse ponto.

Figura 30 – Comparação do erro absoluto médio entre o modelo raso (*tf_mlp_shallow* e profundo *tf_mlp_deep*).



Fonte: Do Autor.

Figura 31 – Comparação do RMSE e R^2 entre o modelo raso (*tf_mlp_shallow*) e profundo (*tf_mlp_deep*)

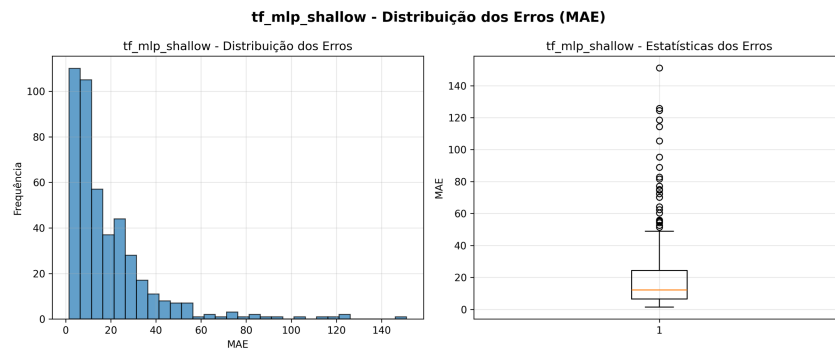


Fonte: Do Autor.

6.2.1 Distribuição de Erros

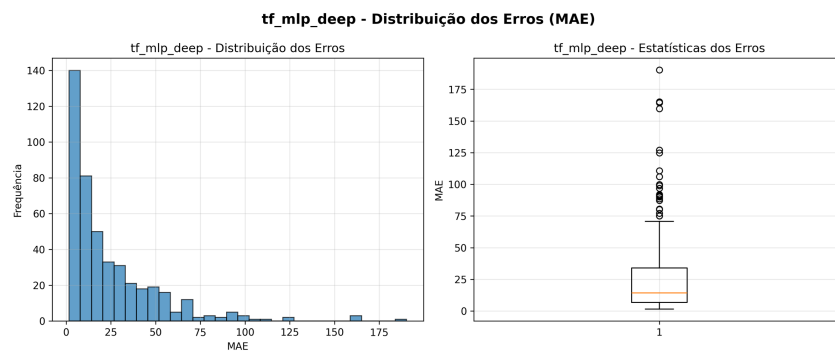
Os modelos de redes neurais do Tensorflow se comportaram de forma muito similar aos modelos intermediários do scikit-learn. Os erros se concentraram bastante nos menores valores. Quando comparado ao modelo MLP do scikit-learn, obtivemos uma melhora considerável mesmo utilizando uma técnica menos eficaz de busca de hiperparâmetros. A figura 32 mostra a distribuição de erros, com um *bin* de 30 e o diagrama de caixa para o modelo raso e a figura 33 para o modelo profundo.

Figura 32 – Distribuição e Diagrama de Caixa do erro médio absoluto para o modelo raso de rede neural.



Fonte: Do Autor.

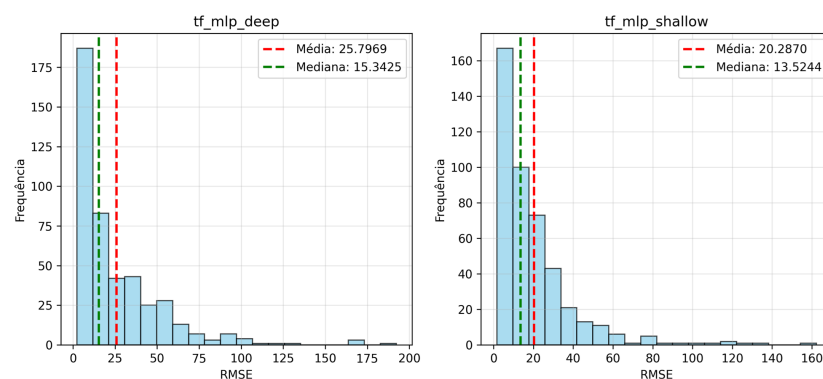
Figura 33 – Distribuição e Diagrama de Caixa do erro médio absoluto para o modelo profundo de rede neural.



Fonte: Do Autor.

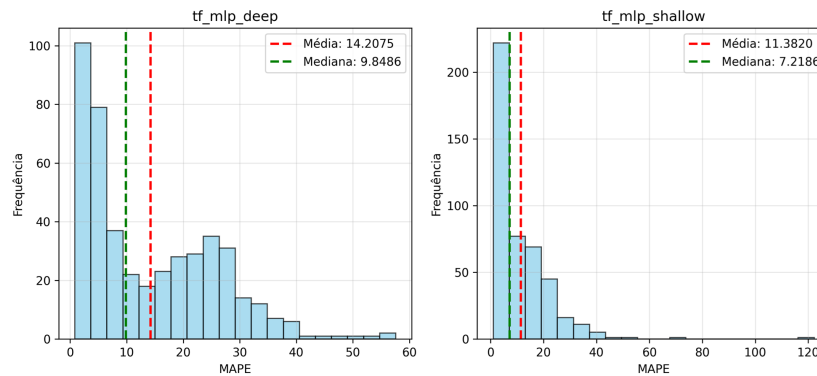
Por fim, temos a análise do RMSE e do MAPE lado a lado para os modelos. Os resultados podem ser vistos nas figuras 34 e 35 respectivamente.

Figura 34 – Distribuição do RMSE dos modelos raso (direita) e profundo (esquerda).



Fonte: Do Autor.

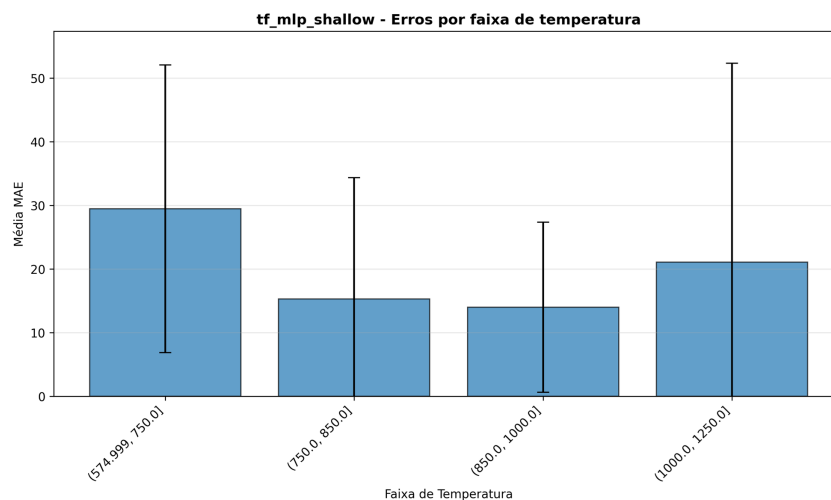
Figura 35 – Distribuição do MAPE dos modelos raso (direita) e profundo (esquerda).



6.2.2 Erro por Temperatura

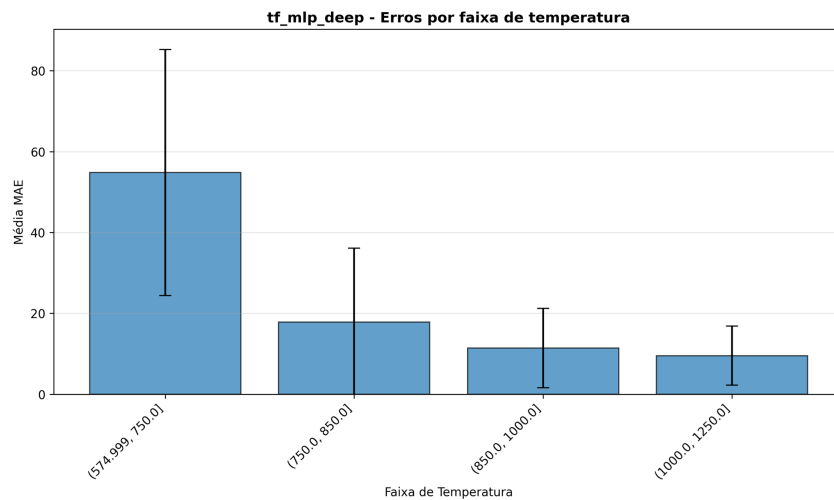
Assim como para os modelos do scikit-learn, uma análise por gráfico de barras para o erro médio foi realizada junto com o desvio padrão para cada faixa de temperatura para o modelo raso e o profundo. A figura 36 mostra que o modelo raso se comporta melhor para as faixas intermediárias de temperatura. Para temperaturas de 750° C, temos o maior erro absoluto médio e para as temperaturas de 1250° C, temos um desvio padrão muito grande o que pode trazer dúvidas sobre a capacidade do modelo em generalizar algo nessa faixa.

Figura 36 – Diagrama de barra mostrando a média do erro absoluto com o desvio padrão para o modelo raso.



A Figura 37 mostra um comportamento um pouco diferente. Nela, podemos ver uma grande concentração de erros na faixa de 750° C, enquanto para as faixas de 850, 1000 e 1250° C, o modelo se comporta relativamente bem, sendo as duas últimas comparáveis com ao XGBoost.

Figura 37 – Diagrama de barra mostrando a média do erro absoluto com o desvio padrão para o modelo profundo.

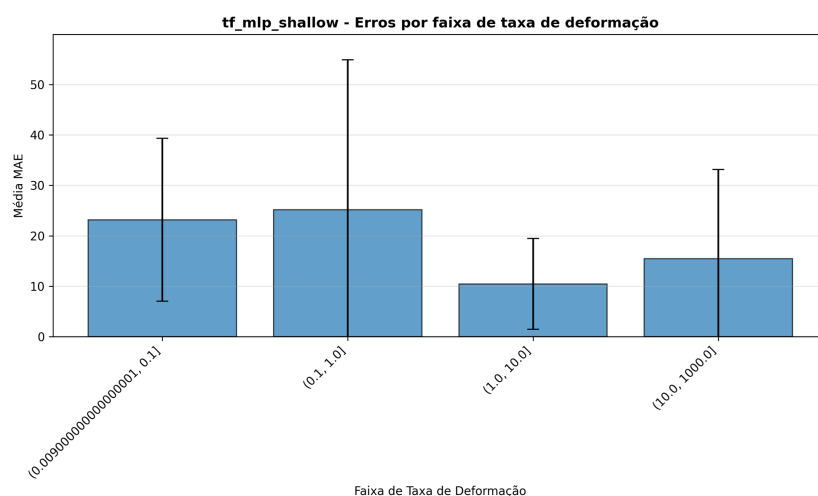


Fonte: Do Autor.

6.2.3 Erro por velocidade de deformação

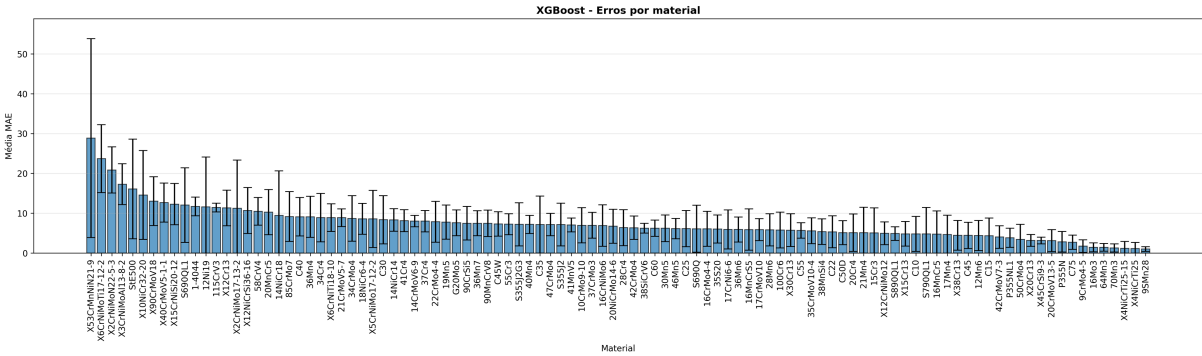
Pela velocidade de deformação, os modelos rasos e profundos possuíram comportamentos muito similares tendo a faixa de 1 como a mais dificultosa para predizer, sendo essa faixa a que possui os maiores valores de média e de desvio padrão. Como melhor faixa, ambos os modelos tiveram a faixa de 10. Foi possível ver também que a faixa de 1 pode ter jogado a média geral do modelo profundo para cima, assim como a temperatura de 750° C. É possível que removendo essas faixas, o modelo profundo fique melhor.

Figura 38 – Diagrama de barra mostrando a média do erro absoluto com o desvio padrão para o modelo raso para a velocidade de deformação.



Fonte: Do Autor.

Figura 40 – XGBoost - Erro por material.



Fonte: Do Autor.

Tabela 6 – Melhores Predições do XGBoost

Material	Temperatura	velocidade de deformação	MAE	RMSE	Maior Erro
X4NiCrTi25	950	100.0	0.303	0.415	0.990
X4NiCrTi25-15	1000	100.0	0.504	0.622	1.373
X4NiCrTi25	900	50.0	0.645	0.859	1.731
X4NiCrTi25-15	900	1.0	0.435	0.537	1.001
X4NiCrTi25-15	950	10.0	0.486	0.769	1.968
X4NiCrTi25	950	0.1	0.293	0.450	1.138
20Cr4	900	10.0	0.393	0.536	1.267
17CrNi6-6	1000	10.0	0.365	0.505	1.168
X4NiCrTi25	900	10.0	0.794	1.153	2.419
9CrMo4-5	1000	10.0	0.361	0.435	0.697

Fonte: Do Autor.

Tabela 7 – Piores predições do XGBoost.

Nome	Temperatura (°C)	velocidade de deformação	MAE	RMSE	Max Error
X53CrMnNi21-9	700	1.0	52.98	53.05	58.56
X53CrMnNi21-9	700	0.1	47.80	48.24	56.94
StE500	1250	100.0	42.52	42.97	46.52
StE500	1250	500.0	35.85	36.77	41.73
X6CrNiMoTi17-12-2	800	1.0	34.39	37.93	53.71
12Ni19	850	10.0	33.55	33.60	36.46
X10NiCr32-20	800	0.1	33.10	33.39	35.16
X2CrNiMo17-13-2	800	1.0	32.88	34.06	38.84
StE500	1100	100.0	28.53	28.59	30.70
14NiCr18	850	10.0	28.49	28.50	30.38

Fonte: Do Autor.

Tabela 8 – Estatísticas de erro por categoria de material.

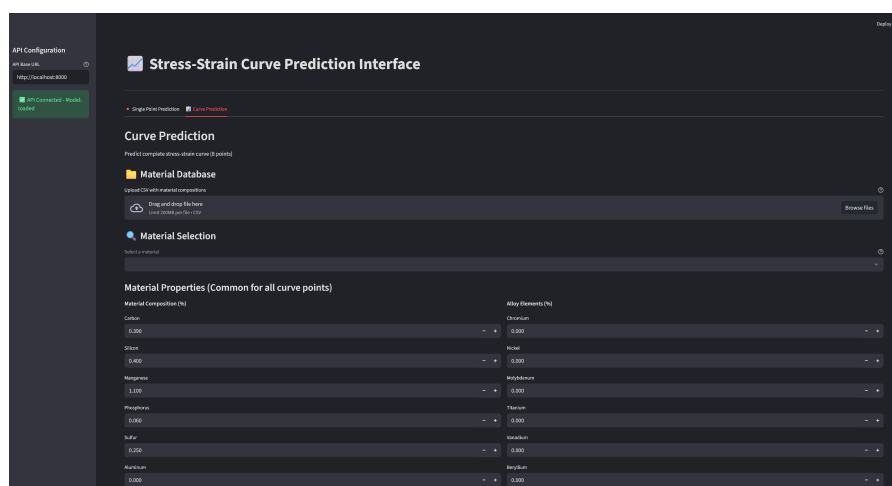
Categoria	Erro médio	Desvio padrão	Materiais	Curvas	Exemplo típico	Descrição
Aços carbono / baixa liga	6.319	2.228	69	294	C45, 42CrMo4, 20MnCr5	Aços ferríticos ou perlíticos com baixo teor de elementos de liga, comportamento previsível de deformação.
Aços estruturais / alta resistência	8.328	4.208	7	40	S355J2, S690QL1, P355N	Aços normalizados ou temperados e revenidos, usados em estruturas de alta resistência.
Aços inoxidáveis / alta liga	10.597	7.440	21	93	X6CrNiMoTi17-12-2, X3CrNiMoN22-5-3	Aços austeníticos, duplex ou endurecíveis por precipitação, com alta complexidade microestrutural e comportamento não linear.
Outros	7.204	4.219	5	22	65Mn28, 63Mn3	Ligas especiais ou difíceis de classificar, com comportamento variável de deformação.

Fonte: Do Autor.

6.4 Ferramenta para o Usuário

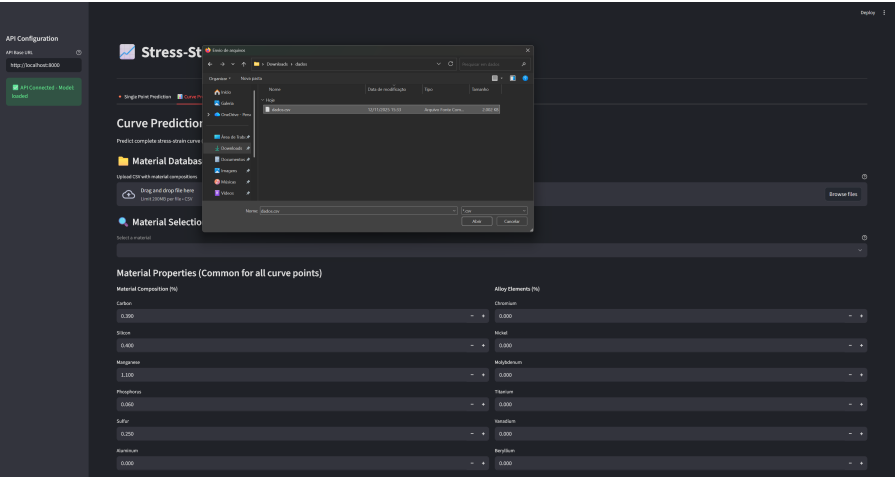
Para disponibilizar o acesso do modelo ao usuário, primeiro foi desenvolvido uma API capaz de receber os valores dos materiais, a temperatura do experimento e a velocidade de deformação. A API carrega o modelo e o normalizador. Com isso ela retorna os pontos da curva e uma imagem com a curva gerada. Para utilizar essa API, foi feito um sistema utilizando o Streamlit. Nele, o usuário pode colocar um CSV com uma lista de materiais com a sua composição. A imagem 41 mostra a imagem inicial da aplicação web, rodando localmente. Na imagem 42 vemos um CSV sendo carregado. Com isso, foi selecionado um material como mostrado na imagem 43 e o valores de sua composição são atualizados como visto na figura 44, onde também vemos que podemos colocar os valores para temperatura e velocidade de deformação. Vale lembrar que podemos colocar valores até mesmo não vistos nos conjuntos de treino, como por exemplo, temperatura de 900° C.

Figura 41 – Página Inicial da ferramenta web de predição.



Fonte: Do Autor.

Figura 42 – Carregamento do CSV com os dados dos materiais.



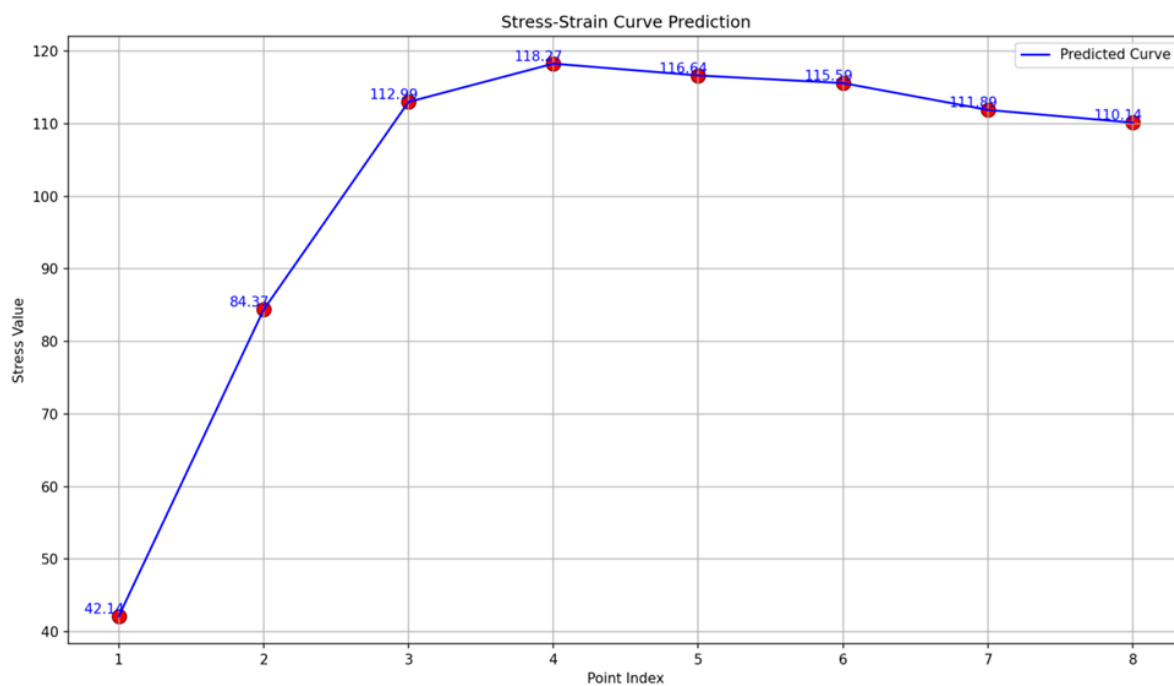
Fonte: Do Autor.

Figura 43 – Seleção de um material.



Fonte: Do Autor.

Figura 46 – Curva gerada pelo modelo via o sistema web em MPa para cada ponto da curva predito



Fonte: Do Autor.

7 Conclusão

O objetivo desse trabalho foi contextualizar o tema de conformação mecânica e como a inteligência artificial pode auxiliar na geração de curvas de escoamento. Também foram relacionados os desafios dessa área e os altos custos envolvidos na indústria para os experimentos envolvendo materiais reais e como foi possível contornar isso utilizando um modelo de aprendizado de máquina.

Durante o estudo, foi provado que alguns modelos de inteligência artificial se comportam melhor para a tarefa e que existe um modelo que se destaca dos demais quando há um conjunto de treino limitado e com materiais que são complexos, como aços inoxidáveis de alta liga. Também foi feita uma análise de erro mais profunda para tentar entender como a temperatura e velocidade de deformação influenciavam os modelos e como cada material se comportava em cima do melhor modelo encontrado nesse trabalho, o que nos mostrou que temos um futuro tema de pesquisa já que, para certas faixa de temperatura e velocidade de deformação, os modelos não iam tão bem quanto as outras.

Para isso, foi criada uma pipeline de treinamento e otimização de hiperparâmetros de modelos utilizando duas bibliotecas populares da área, o scikit-learn e o Tensorflow. Com eles, foram utilizadas técnicas como a busca aleatória e a biblioteca Optuna para achar a melhor combinação de hiperparâmetro de treinamento e, com isso, o melhor modelo. Também foi feita uma análise em nível de curva para os modelos.

Como melhor modelo, foi visto que o XGBoost se destacou para esse caso, sendo mais de duas vezes melhor que o segundo colocado e superando até mesmo modelos de aprendizado profundo. Cabe ressaltar o conjunto de treinamento limitado, o que pode ser uma pista do porque os modelos profundos não conseguiram superar o XGBoost. Apesar da boa performance geral, também foi visto que para materiais complexos, como os aços inoxidáveis de alta liga, o modelo teve dificuldades para gerar os pontos corretos, tendo a maior concentração de erros nessa classe de material. Isso levanta uma segunda proposta de pesquisa futura, uma geração de um dataset com foco em materiais dessa categoria.

No trabalho, também foi mostrado que é possível carregar o melhor modelo em uma API e servi-la usando um sistema web em que o usuário pode colocar as composições do material manualmente ou carregar um arquivo com diversos materiais e selecioná-los para gerar uma nova curva passando a temperatura e a velocidade de deformação que precisar. A API retorna para o usuário tanto os pontos gerados pelo modelo como também a curva em si.

Com isso, os objetivos propostos para essa pesquisa foram atingidos, demonstrando ser possível utilizar modelos de aprendizado de máquina para auxiliar profissionais na

geração de curvas com um erro razoável.

Sobre a evolução desse trabalho, pode-se levantar como pontos de melhoria, expandir o conjunto de treinamento, principalmente para materiais mais complexos e também um estudo aprofundado em como a temperatura e a velocidade de deformação afeta os modelos. Para isso, a sugestão é fazer uma análise desses parâmetros também a nível de material para tentar identificar se os erros encontrados foram realmente devido a temperatura e a velocidade de deformação ou se é algo relacionado a como o material se comporta nessas condições.

Referências

- AGGARWAL, C. C. *Neural Networks and Deep Learning*. 2. ed. [S.l.]: Springer, 2019.
- ALIBABA. *Aluminium Profile Surface Defect Dataset*. 2018.
- ATTAR, H. R.; FOSTER, A.; LI, N. Development of a deep learning platform for sheet stamping geometry optimisation under manufacturing constraints. *Engineering Applications of Artificial Intelligence*, Elsevier Ltd, v. 123, 8 2023. ISSN 09521976.
- ATTAR, H. R.; FOSTER, A.; LI, N. Implicit neural representations of sheet stamping geometries with small-scale features. *Engineering Applications of Artificial Intelligence*, Elsevier Ltd, v. 123, 8 2023. ISSN 09521976.
- BISHOP, C. M. *Pattern recognition and machine learning*. [S.l.]: Springer Science + Business Media, 2009. 738 p. ISBN 0387310738.
- CAO, J.; BAMBACH, M.; MERKLEIN, M.; MOZAFFAR, M.; XUE, T. Artificial intelligence in metal forming. *CIRP Annals*, Elsevier Inc., v. 73, p. 561–587, 1 2024. ISSN 17260604.
- CNI. *Mapa Estratégico da Indústria*. 2013. Disponível em: https://static.portaldaindustria.com.br/media/filer_public/06/47/0647c470-35c7-4bda-a4a6-424ef8059afc/20130828171714831765o.pdf.
- DING, J. A review of deep neuron network applications in extrusion die design. In: *Metal Forming 2024*. [S.l.]: Materials Research Forum LLC, 2024. v. 44, p. 511–518. ISBN 9781644903247. ISSN 2474395X.
- ERCOLANI, J. B.; KEHL, L. R. Avaliação dos métodos de tração e compressão para obtenção de curvas de escoamento de ligas de alumínio. 2021. ISSN 2176-7270.
- GERON, A. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow : concepts, tools, and techniques to build intelligent systems*. [S.l.]: O'Reilly Media, Inc., 2019. 819 p. ISBN 9781492032649.
- GROOVER, M. P. *Fundamentals of Modern Manufacturing Materials, Processes, and Systems Seventh Edition*. 2013.
- GUO, Q.; LI, C.; DENG, X.; DONG, X.; ZHANG, C. An adaptive incremental two-stage framework for crack defect detection. *Multimedia Tools and Applications*, Springer, v. 83, p. 69249–69271, 8 2024. ISSN 15737721.
- HOSFORD, W.; CADDELL, R. *Metal Forming : Mechanics and Metallurgy*. [S.l.]: Cambridge University Press, 2007. 328 p. ISBN 9780511354533.
- HU, P.; MA, N.; LIU, L. zhong; ZHU, Y. guo. *Theories, Methods and Numerical Technology of Sheet Metal Cold and Hot Forming*. [S.l.]: Springer London, 2013. ISBN 978-1-4471-4098-6.

KOCATÜRK, F.; TOPARLI, M. B.; TANRIKULU, B.; YURTDAG, S.; ZEREN, D.; KILIÇASLAN, C. Flow curve prediction of cold forging steel by artificial neural network model. In: *ESAFORM 2021 - 24th International Conference on Material Forming*. [S.l.]: PoPuPS (University of Liège Library), 2021. ISBN 9782870193020.

LESINSKI, G.; CORNS, S.; DAGLI, C. Application of an artificial neural network to predict graduation success at the united states military academy. *Procedia Computer Science*, v. 95, p. 375–382, 2016. ISSN 18770509.

LIU, S.; XIA, Y.; SHI, Z.; YU, H.; LI, Z.; LIN, J. Deep learning in sheet metal bending with a novel theory-guided deep neural network. *IEEE/CAA Journal of Automatica Sinica*, Institute of Electrical and Electronics Engineers Inc., v. 8, p. 565–581, 3 2021. ISSN 23299274.

LOW, D. W. W.; CHAUDHARI, A.; KUMAR, D.; KUMAR, A. S. Convolutional neural networks for prediction of geometrical errors in incremental sheet metal forming. *Journal of Intelligent Manufacturing*, Springer, v. 34, p. 2373–2386, 6 2023. ISSN 15728145.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, v. 5, p. 115–133, 12 1943. ISSN 0007-4985.

MEHTEDI, M. E.; FORCELESE, A.; GRECO, L.; PIERALISI, M.; SIMONCINI, M. Flow curve prediction of zam100 magnesium alloy sheets using artificial neural network-based models. In: *Procedia CIRP*. [S.l.]: Elsevier B.V., 2019. v. 79, p. 661–666. ISSN 22128271.

MICHALSKI, R.; CARBONELL, J.; MITCHELL, T. *Machine Learning: An Artificial Intelligence Approach*. Springer Berlin Heidelberg, 2013. (Symbolic Computation). ISBN 9783662124055. Disponível em: <<https://books.google.com.br/books?id=-eqpCAAQBAJ>>.

MORO, N.; AURAS, A. P. *CONFORMAÇÃO MECÂNICA I – Generalidades, Laminação e Forjamento*. [S.l.], 2007. Disponível em: <http://www.norbertocetesc.pro.br/downloads/pfb_conformacaoi.pdf>.

MURPHY, K. P. *Machine learning : a probabilistic perspective*. [S.l.: s.n.], 2012. 1067 p. ISBN 9780262018029.

OREIRO, J. L.; FEIJO, C. A. De-industrialization: concept, causes, effects and the brazilian case. *Brazilian Journal of Political Economy*, v. 30, n. 2, p. 219–232, Apr. 2010. Disponível em: <<https://centrodeeconomiapolitica.org/repos/index.php/journal/article/view/439>>.

ROCHA, O. F. L. da. *Conformação Mecânica*. [S.l.], 2012. Disponível em: <https://redeetec.mec.gov.br/images/stories/pdf/eixo_ctrl_proc_indust/tec_metal/conform_mec/161012_confor_mec.pdf>.

ROSIK, A. *Obtenção da curva de escoamento a quente do aço 22MnB5 a partir do método de empilhamento de chapas finas*. [S.l.], 2018.

RUSSELL, S.; NORVIG, P. *Artificial Intelligence: A Modern Approach (Pearson Series in Artificial Intelligence)*. [S.l.]: Pearson, 2020. ISBN 0134610997.

- SANTOS, L. H.; SCHAEFFER, L.; FORTUNATO, S. Q.; CÉZAR, J. L.; FLACH, M. A. Estudo da influência dos tratamentos térmicos de recozimento pleno e esferoidização nas curvas de escoamento a frio do aço abnt 1045. *Tecnologia em Metalurgia Materiais e Mineração*, Editora Cubo, v. 15, p. 75–79, 2018.
- SAUER, C.; SCHLEICH, B.; WARTZACK, S. Deep learning in sheet-bulk metal forming part design. In: *Proceedings of International Design Conference, DESIGN*. [S.l.]: Faculty of Mechanical Engineering and Naval Architecture, 2018. v. 6, p. 2999–3010. ISBN 9789537738594. ISSN 18479073.
- SCHAEFFER, L. *Conformação Mecânica*. 2. ed. [S.l.]: Imprensa Livre, 2004.
- SCHULER. *Metal forming handbook*. [S.l.]: Springer-Verlag, 1998. 563 p. ISBN 3540611851.
- SILVA, M. L. da; SILVA, R. A. da; CORONEL, D. A.; FILHO, P. J. M. O setor industrial brasileiro: Desafios e oportunidades. *RDE - Revista de Desenvolvimento Econômico*, v. 2, p. 28–54, 2019. ISSN 21788022.
- TAULLI, T. *Artificial Intelligence Basics*. [S.l.]: Apress, 2019.
- TURING, A. M. I.—COMPUTING MACHINERY AND INTELLIGENCE. *Mind*, LIX, n. 236, p. 433–460, 10 1950. ISSN 0026-4423. Disponível em: <<https://doi.org/10.1093/mind/LIX.236.433>>.
- WANG, M.; DU, M.; LU, H.; HAN, Y.; ZHENG, Y. Enhancing the high-temperature constitutive modeling of in706 superalloy using advanced artificial neural networks and attentive staged optimization algorithm. *Computational Materials Science*, Elsevier B.V., v. 237, 3 2024. ISSN 09270256.
- WARWICK, K.; SHAH, H. Can machines think? a report on turing test experiments at the royal society. *Journal of Experimental Theoretical Artificial Intelligence*, v. 28, p. 989–1007, 11 2016. ISSN 0952-813X.
- YE, L.; XIA, X.; CHAI, B.; WANG, S.; YANG, B. Application of deep learning in workpiece defect detection. In: *Procedia Computer Science*. [S.l.]: Elsevier B.V., 2021. v. 183, p. 267–273. ISSN 18770509.
- ZHANG, Z.; LIU, Q.; WANG, Y. Road extraction by deep residual u-net. *IEEE Geoscience and Remote Sensing Letters*, v. 15, n. 5, p. 749–753, 2018.
- ZHU, L.; LI, N. Springback prediction for sheet metal cold stamping using convolutional neural networks. In: . [S.l.]: SPIE-Intl Soc Optical Eng, 2023. p. 38. ISBN 9781510666634. ISSN 1996756X.