



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**CENTRO DE TECNOLOGIA**  
**DEPARTAMENTO DE ENGENHARIA DE TELEINFORMÁTICA**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE TELEINFORMÁTICA**  
**DOUTORADO EM ENGENHARIA DE TELEINFORMÁTICA**

**RENAN BESSA**

**CONTRIBUIÇÕES AO PROBLEMA DE PREDIÇÃO DAS PROPRIEDADES  
MECÂNICAS DE VERGALHÕES DE AÇO USANDO REGRESSÃO POR MÁQUINAS  
DE KERNEL**

**FORTALEZA**

**2024**

RENAN BESSA

CONTRIBUIÇÕES AO PROBLEMA DE PREDIÇÃO DAS PROPRIEDADES MECÂNICAS  
DE VERGALHÕES DE AÇO USANDO REGRESSÃO POR MÁQUINAS DE KERNEL

Tese apresentada ao Programa de Pós-Graduação em Engenharia de Teleinformática do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do título de doutor em Engenharia de Teleinformática. Área de Concentração: Sinais e Sistemas.

Orientador: Prof. Dr. Guilherme de Alencar Barreto.

Coorientador: Prof. Dr. Elineudo Pinho de Moura.

FORTALEZA

2024

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Sistema de Bibliotecas  
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

---

B465c Bessa, Renan.  
Contribuições ao Problema de Predição das Propriedades Mecânicas de Vergalhões de Aço Usando Regressão por Máquinas de Kernel / Renan Bessa. – 2024.  
75 f. : il. color.

Tese (doutorado) – Universidade Federal do Ceará, Centro de Tecnologia, Programa de Pós-Graduação em Engenharia de Teleinformática, Fortaleza, 2024.  
Orientação: Prof. Dr. Guilherme de Alencar Barreto.  
Coorientação: Prof. Dr. Elineudo Pinho de Moura.

1. Estimativa de propriedades mecânicas. 2. Limite de escoamento. 3. Limite de Resistência à Tração. 4. Alongamento Percentual. 5. Regressão por Vetores Suporte via Mínimos Quadrados. I. Título.

CDD 621.38

---

RENAN BESSA

CONTRIBUIÇÕES AO PROBLEMA DE PREDIÇÃO DAS PROPRIEDADES MECÂNICAS  
DE VERGALHÕES DE AÇO USANDO REGRESSÃO POR MÁQUINAS DE KERNEL

Tese apresentada ao Programa de Pós-Graduação em Engenharia de Teleinformática do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do título de doutor em Engenharia de Teleinformática. Área de Concentração: Sinais e Sistemas.

Aprovada em: 30/07/2024.

BANCA EXAMINADORA

---

Prof. Dr. Guilherme de Alencar  
Barreto (Orientador)  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Elineudo Pinho de  
Moura (Coorientador)  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Ivan Nunes da Silva  
Universidade de São Paulo (USP)

---

Prof. Dr. Petronio Cândido de Lima e Silva  
Instituto Federal do Norte de Minas Gerais  
(IFNMG)

---

Prof. Dr. Ajalmar Rêgo da Rocha Neto  
Instituto Federal do Ceará (IFCE)

## AGRADECIMENTOS

Gostaria de expressar minha sincera gratidão a todos que contribuíram, direta ou indiretamente, para a realização desta tese de doutorado.

Ao meu orientador, Professor Guilherme de Alencar Barreto, pelo constante apoio, orientação e paciência ao longo desta jornada. Suas sugestões e críticas foram fundamentais para o desenvolvimento deste trabalho.

Agradeço também ao Professor Elineudo Pinho de Moura pela disponibilização do conjunto de dados e pela colaboração na pesquisa.

Aos amigos do laboratório SPIRAL — Igor Sousa, Renan Fonteles, Diego Perdigão, Kelvin Sales, Rafael Eufrázio, Patricia Tavares, Michael Duarte e Professor Charles Cavalcante —, deixo meu agradecimento pelas colaborações, pelas discussões produtivas e pelo ambiente acolhedor. Em especial, ao David Coelho, pela parceria e pelo apoio essenciais à concretização deste trabalho.

Por fim, agradeço à minha família, em especial à minha mãe, Silvana, pelo carinho e incentivo, e à minha namorada, Duda, pelo amor e apoio ao longo desta trajetória.

A todos, o meu sincero muito obrigado.

## RESUMO

Visando garantir a qualidade do produto e reduzir o custo da fabricação do aço, desenvolvem-se modelos de regressão não linear para prever as propriedades mecânicas de vergalhões de aço usando técnicas de aprendizado de máquina. Tendo isso em mente, revisita-se esse problema desenvolvendo uma metodologia de arquitetura que combina dois conceitos chaves na construção de modelos parcimoniosos: (i) esparsidade, propriedade na qual um modelo preditivo é construído com menos elementos relevantes (e.g., vetores suporte (VSs)) e, conseqüente, com menos parâmetros associados, e (ii) localidade, propriedade na qual vários modelos mais simples que são ajustados a partições menores dos dados. Nesse contexto, desenvolvem-se dois modelos de regressão baseados no modelo regressão por vetores suporte via mínimos quadrados (least squares support vector regression, LSSVR). O primeiro é uma versão esparsa aprimorada e o segundo é um novo modelo de regressão local baseado em LSSVR. A tarefa de interesse é a previsão de quatro variáveis que são usadas para avaliar as propriedades mecânicas do vergalhão de aço produzido, a saber: o limite de escoamento (LE), o limite de resistência à tração (LRT), a razão LRT/LE e o alongamento percentual (AP) com base em informações sobre sua composição química (12 variáveis) e os parâmetros do tratamento térmico de laminação (6 variáveis). Avaliam-se os modelos propostos usando dados reais coletados da fabricação de vergalhões de aço e comparam-se com o modelo global de LSSVR em termos de acurácia e eficiência. A abordagem local e esparsa do modelo LSSVR com o kernel de Cauchy consegue consistentemente superar a abordagem padrão de modelo de regressão único com o kernel gaussiano na tarefa de interesse, alcançando melhorias médias no  $R^2$  em relação aos estudos anteriores: 6,49% para LE, 6,47% para LRT, 2,72% para LRT/LE e 4,95% para AP. Além disso, a esparsificação do modelo e a abordagem de regressão local reduzem significativamente o número de operações com VSs, utilizando em média 36,25% do total de VSs disponíveis para a estimativa de LE, 38,75% para LRT, 31,83% para LRT/LE e 36,73% para AP, reduzindo aproximadamente 52% do tempo de cálculo de previsão das saídas do modelo.

**Palavras-chave:** Estimacão de propriedades mecânicas; Limite de Escoamento; Limite de Resistência à Tração; Razão LRT/LE; Alongamento Percentual; Regressão por Vetores Suporte via Mínimos Quadrados.

## ABSTRACT

Aiming to ensure product quality and reduce the cost of steel manufacturing, nonlinear regression models are developed to predict the mechanical properties of steel rebar using machine learning techniques. Bearing this in mind, this problem is revisited by developing an architectural methodology that combines two key concepts in the construction of parsimonious models: (i) sparsity, a property in which a predictive model is built with fewer relevant elements (e.g., support vectors) and consequently with fewer associated parameters, and (ii) locality, a property in which several simpler models are fitted to smaller partitions of the data. In this context, two regression models based on the (least squares support vector regression, LSSVR) model are developed. The first is an enhanced sparse version, and the second is a new local regression model based on LSSVR. The task of interest is the prediction of four variables used to evaluate the mechanical properties of the produced steel rebar, namely: yield strength (YS), ultimate tensile strength (UTS), the UTS/YS ratio, and percent elongation (PE) based on information about its chemical composition (12 variables) and the parameters of the rolling heat treatment (6 variables). The proposed models are evaluated using real data collected from steel rebar manufacturing and compared with the global LSSVR model in terms of accuracy and efficiency. The local and sparse approach of the LSSVR model with the Cauchy kernel consistently outperforms the standard single regression model approach with the Gaussian kernel in the task of interest, achieving average improvements in  $R^2$  compared to previous studies: 6.49% for YS, 6.47% for UTS, 2.72% for UTS/YS, and 4.95% for PE. Additionally, the model sparsification and local regression approach significantly reduce the number of operations with support vectors, using an average of 36.25% of the total available support vectors for the estimation of YS, 38.75% for UTS, 31.83% for UTS/YS, and 36.73% for PE, reducing approximately 52% of the calculation time for predicting the model outputs.

**Keywords:** Mechanical property estimation; Yield Strength; Ultimate Tensile Strength; UTS/YS ratio; Percent Elongation; Least Square Support Vector Regression.

## LISTA DE FIGURAS

Figura 1 – Ilustração do processo de fabricação de vergalhões de aço e aquisição do conjunto de dados. . . . .	23
Figura 2 – Curva tensão-deformação de ensaio de tração até a fratura. . . . .	25
Figura 3 – Comparação geral de desempenho dos modelos avaliados em valores de $R^2$ médio e de desvio padrão para 100 experimentos. . . . .	27
Figura 4 – <i>Boxplot</i> de desempenho utilizando a <i>LS-SVMlab Toolbox</i> na sintonia dos hiperparâmetros do kernel gaussiano para diferentes proporções do conjunto de treinamento. . . . .	33
Figura 5 – <i>Boxplot</i> de desempenho para diferentes tipos de kernel utilizando a <i>LS-SVMlab Toolbox</i> . . . . .	34
Figura 6 – <i>Boxplot</i> de desempenho para diferentes formas de sintonia de hiperparâmetros. . . . .	36
Figura 7 – <i>Boxplot</i> de desempenho para diferentes ordens da função logarítmica. . . . .	38
Figura 8 – <i>Boxplot</i> para comparação de desempenho entre diferentes funções de kernel implementação própria do modelo LSSVR. . . . .	39
Figura 9 – <i>Boxplot</i> dos tempos de execução dos diferentes kernels durante a previsão do modelo. Os pontos verdes no gráfico de enxame mostram as observações individuais do tempo necessário para a previsão das amostras do conjunto de teste para cada saída, ajustados horizontalmente para evitar sobreposição e melhorar a visualização da densidade dos dados. . . . .	40
Figura 10 – Comparação entre os algoritmos <i>k-means</i> e <i>k-medoids</i> . Os círculos representam os protótipos (centróides ou medóides), e as linhas tracejadas indicam as fronteiras de Voronoi que delimitam as partições do espaço de dados. . . . .	45
Figura 11 – Representação do método ALD4P para situação inicial com três vetores de suporte no espaço de características $H$ . . . . .	47
Figura 12 – Diagrama de resumo da otimização dos hiperparâmetros do modelo LSSVR. . . . .	48
Figura 13 – Diagrama de resumo de metodologia de esparsificação. . . . .	49
Figura 14 – Barra de erro dos resultados dos métodos de esparsificação para saída LE. . . . .	50
Figura 15 – Barra de erro dos resultados dos métodos de esparsificação para saída LRT. . . . .	50
Figura 16 – Barra de erro dos resultados dos métodos de esparsificação para saída LRT/LE. . . . .	51
Figura 17 – Barra de erro dos resultados dos métodos de esparsificação para saída AP. . . . .	51
Figura 18 – Diagrama de resumo de modelagem local. . . . .	56

Figura 19 – Desempenho dos modelos locais juntamente com as respectivas quantidades de VSs utilizados e o tempo gasto durante a previsão das saídas. . . . .	58
Figura 20 – <i>Boxplot</i> de desempenho para diferentes modelos LSSVR. . . . .	59
Figura 21 – <i>Boxplot</i> dos tempos de execução dos diferentes modelos LSSVR durante a previsão das suas saídas. . . . .	60
Figura 22 – Gráficos de dispersão comparando os valores preditos e observados das 100 rodadas. . . . .	61
Figura 23 – Comparação geral de desempenho dos modelos avaliados em valores de $R^2$ médio e de desvio padrão para 100 experimentos com novos resultados. . .	61

## LISTA DE TABELAS

Tabela 1 – Descrição estatística da composição química das amostras [%]. . . . .	24
Tabela 2 – Descrição estatística dos parâmetros do tratamento térmico. . . . .	24
Tabela 3 – Sumário estatístico das saídas observadas referentes à propriedades mecânicas dos vergalhões produzidos. . . . .	26
Tabela 4 – Comparação de desempenho entre diferentes funções de kernel da <i>LS-SVMlab Toolbox</i> . . . . .	34
Tabela 5 – Comparação de desempenho entre diferentes funções de kernel implementação própria do modelo LSSVR. . . . .	38
Tabela 6 – Comparação de desempenho entre diferentes com foco na esparsificação do modelos. . . . .	52
Tabela 7 – Comparação de desempenho entre diferentes com foco na modelagem local.	57
Tabela 8 – Comparação de desempenho entre diferentes modelos de AM. . . . .	60

## LISTA DE ALGORITMOS

Algoritmo 1	– Pseudocódigo do $ \alpha $ - <i>pruning</i> . . . . .	42
Algoritmo 2	– Pseudocódigo do <i>k-means</i> para redução do volume de dados. . . . .	43
Algoritmo 3	– Pseudocódigo do <i>k-medoids</i> para redução do volume de dados. . . . .	44
Algoritmo 4	– Pseudocódigo do ALD4P. . . . .	46

## LISTA DE ABREVIATURAS E SIGLAS

ALD	<i>approximate linear dependence</i>
ALD4P	<i>ALD for pruning</i>
ALD4P-IO	<i>ALD4P with input and output</i>
AM	aprendizado de máquina
AP	alongamento percentual
CNN	<i>convolutional neural network</i>
CSA	<i>coupled simulated annealing</i>
DL	<i>deep learning</i>
ELM	<i>extreme learning machine</i>
KMOD	<i>kernel with moderate decreasing</i>
LE	limite de escoamento
LR	<i>linear regression</i>
LRT	limite de resistência à tração
LS-LSSVR	<i>local sparse LSSVR</i>
LSSVR	<i>least squares support vector regression</i>
MLM	<i>minimum learning machine</i>
MLP	<i>multilayer perceptron</i>
PE	percent elongation
PSO	<i>particle swarm optimization</i>
QV	quantização vetorial
RF	<i>random forest</i>
S-LSSVR	<i>sparse LSSVR</i>
SVR	<i>support vector regression</i>
UTS	ultimate tensile strength
VS	vetor suporte
YS	yield strength

## LISTA DE SÍMBOLOS

$R^2$	Coefficiente de determinação
$\mathbb{R}$	Conjunto dos números reais
$y_n$	$n$ -ésimo elemento do conjunto de dados
$S_{tt}$	Conjunto de dados de teste
$N_{tt}$	Quantidade total de elementos do conjunto de teste
$\hat{y}_n$	$n$ -ésimo valor previsto pelo modelo
$\bar{y}_n$	Média aritmética dos valores da saída desejada do conjunto de teste
$f(\cdot)$	Função de um modelo de regressão.
$S_{tr}$	Conjunto de dados de treinamento
$N_{tr}$	Quantidade total elementos do conjunto de treinamento
$\mathbf{x}_n$	Vetor de variáveis de entrada do $n$ -ésima amostra
$N_x$	Número de dimensões de $\mathbf{x}_n$
$\mathbf{y}_n$	Vetor de variáveis de saída do $n$ -ésima amostra.
$N_y$	Número de dimensões de $\mathbf{y}_n$
$\phi(\cdot)$	Função de uma mapeamento não linear de alta dimensão
$H$	Número de dimensões do mapeamento de $\phi(\cdot)$
$\mathbf{w}$	Vetor de parâmetros de $f(\cdot)$
$b$	Intercepto de $f(\cdot)$
$e_n$	Erro associado ao valor observado na $n$ -ésima amostra
$\gamma$	Constante de regularização do modelo
$J_p(\cdot)$	Função perda de um modelo de regressão
$\mathcal{L}(\cdot)$	Função de Lagrange

$\alpha$	Vetor de multiplicadores de Lagrange
$\partial(\cdot)$	Operador de derivação
$\Omega$	Matriz de kernel
$k(\cdot, \cdot)$	Função de kernel
$\ \cdot\ _2$	Norma euclidiana
$\sigma$	Largura da função Gaussiana
$c$	Intercepto da função polinomial
$d$	Ordem da função polinomial
$\mathbb{Z}_+^*$	Conjunto dos números inteiros não nulos
$\rho_h, \sigma_h$	Hiperparâmetros para otimização da função de kernel
$\mathbf{x}_n^*$	Vetor de entradas e saídas concatenadas.
$N_r$	Número de amostras no conjunto de treinamento esparsos
$\mathcal{P}_i$	$i$ -ésima partição
$\mathbf{c}_i$	Vetor centróide da $i$ -ésima partição
$\mathbf{m}_i$	Vetor medóide da $i$ -ésima partição
$\mathcal{D}$	Dicionário dos métodos baseados em <i>ALD</i> .
$\tilde{\mathbf{x}}_j$	$j$ -ésimo vetor do dicionário
$m$	Quantidade de vetores no dicionário
$\mathbf{a}$	Vetor de coeficientes dos métodos baseados em <i>ALD</i>
$\delta_t$	Erro de aproximação da $t$ -ésima amostra
$t$	Iteração em treinamento online
$\mu$	Parâmetro de nível de esparsidade
$n_s$	Semente aleatória

$K$	Quantidade de partições dos métodos de agrupamentos
$\mathcal{C}_i$	$i$ -ésima partição do conjunto de dados de treinamento
$h_i(\cdot)$	$i$ -ésima função de ponderação
$f_i(\cdot)$	Função do $i$ -ésimo modelo local

## SUMÁRIO

1	<b>INTRODUÇÃO</b>	16
1.1	<b>Objetivos da Tese</b>	19
1.2	<b>Produção Científica</b>	20
1.3	<b>Organização da Tese</b>	20
2	<b>SOBRE O CONJUNTO DE DADOS</b>	22
2.1	<b>Sobre a Produção dos Vergalhões de Aço</b>	22
2.2	<b>Sobre as Propriedades Mecânicas de Vergalhões de Aço</b>	25
2.3	<b>Estado da Arte</b>	26
2.4	<b>Conclusão do Capítulo</b>	28
3	<b>REGRESSÃO POR MÁQUINAS DE KERNEL</b>	29
3.1	<b>Regressão por Vetores Suporte via Mínimos Quadrados</b>	29
3.2	<b>Experimentos Preliminares com a <i>LS-SVMlab Toolbox</i></b>	31
3.3	<b>Implementação customizada do modelo LSSVR</b>	34
3.4	<b>Explorando Diferentes Funções de Kernel</b>	35
3.5	<b>Conclusão do Capítulo</b>	40
4	<b>MÉTODOS DE REDUÇÃO DE VETORES SUPORTE</b>	41
4.1	<b>Poda por Magnitude de <math>\alpha</math></b>	41
4.2	<b><i>K-Means</i> para Redução do Volume de Dados</b>	42
4.3	<b><i>K-Medoids</i> para Redução do Volume de Dados</b>	43
4.4	<b><i>ALD</i> para Poda de VSs</b>	44
4.5	<b>Sobre a Esparsificação do Modelo LSSVR</b>	48
4.6	<b>Conclusão do Capítulo</b>	53
5	<b>MODELAGEM LOCAL PARA REGRESSÃO KERNEL</b>	54
5.1	<b>Modelagem Local Baseada em Agrupamentos</b>	54
5.2	<b>Sobre a Modelagem Local da para Regressão Usando LSSVR</b>	55
5.3	<b>Conclusão do Capítulo</b>	60
6	<b>CONCLUSÃO E TRABALHOS FUTUROS</b>	62
6.1	<b>Contribuições da tese</b>	62
6.2	<b>Trabalhos Futuros</b>	63
	<b>REFERÊNCIAS</b>	65

<b>APÊNDICES</b> . . . . .	72
<b>APÊNDICE A</b> – Sintonia de Hiperpâmetros com <i>PSO+Simplex</i> . . . . .	72

## 1 INTRODUÇÃO

A identificação das propriedades mecânicas na fabricação do aço é de extrema importância para a indústria siderúrgica, possibilitando tanto uma relação custo-benefício favorável quanto a qualidade do produto. Em particular, as barras de aço laminado a quente, também conhecidas como barras de reforço (ou simplesmente vergalhões), têm sido utilizadas principalmente para reforçar estruturas de concreto, proporcionando resistência e durabilidade a edifícios, pontes e outros projetos de infraestrutura. Entre as principais propriedades mecânicas de interesse para os vergalhões de aço, podem-se mencionar o limite de escoamento (LE), o limite de resistência à tração (LRT), a razão LRT/LE e o alongamento percentual (AP) (Murta *et al.*, 2021; Murta *et al.*, 2022). As quantidades LE e LRT são importantes indicadores de desempenho dos materiais sob condições de carga, representando, respectivamente, o ponto inicial de deformação permanente e a carga máxima que o material pode suportar antes de falhar. A razão LRT/LE e AP são indicadores de ductilidade do material, ou seja, a capacidade do material de se deformar sem fraturar (Black e Kohser, 2021; Silva *et al.*, 2023).

É de amplo conhecimento que as propriedades mecânicas dos vergalhões de aço dependem fortemente de sua composição química e dos parâmetros relacionados aos processos de produção (Arumugam *et al.*, 2022; Sajid *et al.*, 2020). No entanto, essas propriedades ainda são determinadas por meio de testes laboratoriais custosos e demorados a partir de amostras retiradas do lote produzido. Tendo isso em mente, muitos autores têm utilizado alternativas baseadas em dados para antecipar (ou prever) tais propriedades mecânicas com a intenção de otimizar o processo de fabricação, viabilizando a criação de planos de produção e convertê-los em uma série de programas de controle que podem ser automaticamente transferidos para máquinas, reduzindo potenciais erros sistemáticos (Navon *et al.*, 1998). Devido à complexidade da tarefa, essas alternativas consistem basicamente em ajustar um modelo de regressão não linear aos dados disponíveis, utilizando como entradas informações da composição química e parâmetros do processo de produção (por exemplo, do tratamento térmico de laminação) e, como saídas, uma ou mais das quantidades de interesse (por exemplo, LE, LRT e AP).

Modelos de aprendizado de máquina (AM) têm sido frequentemente escolhidos para construir modelos de regressão não linear a fim de prever as propriedades mecânicas dos produtos na indústria siderúrgica. Exemplos que merecem destaque podem ser encontrados na fabricação de chapa de aço laminado a quente (Xie *et al.*, 2021), de bobina galvanizada de aço laminada a frio (Lalam *et al.*, 2019), de aços livres de intersticiais (Sgrott *et al.*, 2021), de

vergalhões de aço laminado a quente (Murta *et al.*, 2021; Murta *et al.*, 2022) e na siderurgia em geral (Sayed e Diab, 2019; Zhu *et al.*, 2020; Boto *et al.*, 2022; Wang *et al.*, 2023). Uma das primeiras aplicações bem-sucedidas de modelos de regressão baseados em AM para a previsão das propriedades mecânicas do aço é relatada por Wang *et al.* (2006). Nesse trabalho, os autores utilizaram a regressão por vetores suporte (*support vector regression*, SVR) com 29 entradas (19 extraídas da composição química e 10 dos parâmetros de laminação a quente) coletadas pelo supervisor desse processo para prever, com bom desempenho, as propriedades mecânicas do aço carbono simples laminado a quente Q235B, quantificadas como três variáveis de saída (LE, LRT e AP).

Chou *et al.* (2016) introduziram uma rede perceptron multicamadas (*multilayer perceptron*, MLP) com uma camada oculta para modelar e otimizar a composição química de barra de aço com o objetivo de prever suas propriedades mecânicas, considerando LRT e LE como variáveis de saída. Esse modelo de regressão baseado na MLP foi sintonizada por meio de otimização por enxame de partículas (*particle swarm optimization*, PSO) para explorar melhores soluções no espaço de busca através de iterações. De acordo com os autores, os resultados obtidos são consistentes com os dados reais. Seguindo um paradigma semelhante, um modelo de regressão baseado em MLP com uma camada oculta também é desenvolvido por Lalam *et al.* (2019) para prever as propriedades mecânicas de uma bobina, a saber, LE e LRT, a partir de sua composição química, espessura, largura e principais parâmetros do processo de galvanização, resultando em um total de 23 variáveis de entrada. Além disso, um sistema de monitoramento de qualidade online é desenvolvido com o objetivo de observar as propriedades mecânicas previstas e os parâmetros do processo de uma bobina galvanizada, auxiliando a equipe de qualidade na tomada de decisões.

O modelo preditivo desenvolvido por Murta *et al.* (2021) é construído com base em 18 entradas derivadas da composição química (12 entradas) e do tratamento térmico (6 entradas) de vergalhões de aço. Os métodos empregados incluem a regressão linear (*linear regression*, LR) múltipla e a rede MLP com duas camadas ocultas, cujo objetivo é prever as propriedades mecânicas dos vergalhões de aço, a saber, LE, LRT (ou alternativamente, a razão LRT/LE) e PE. O melhor desempenho, medido pelos valores de  $R^2$ , é obtido pelo modelo de regressão baseado em MLP. Uma MLP com duas camadas ocultas e 27 entradas é projetado por Xie *et al.* (2021) para prever as propriedades mecânicas (LE, LRT, AP e resistência ao impacto) de placas de aço industriais com base nos parâmetros do processo, bem como na composição do aço bruto. Os

autores aplicam esse modelo para uso online em uma planta de fabricação de aço real.

Quatro métodos de AM são avaliados por Hong *et al.* (2021) na tarefa de prever a ductilidade de aço fundido a quente a partir da composição química e das condições térmicas. Uma MLP com uma camada oculta obtém o melhor desempenho em comparação com a floresta aleatória (*random forest*, RF), o processos gaussianos (*Gaussian processes*) e o SVR. Uma MLP com uma camada oculta também é utilizado por Du *et al.* (2021) para prever as propriedades mecânicas de uma liga Fe-C de grãos ultrafinos, fundindo combinações experimentais da composição da liga, dos parâmetros do processo de laminação e dos parâmetros do processo de tratamento térmico. Nesse trabalho, a MLP supera o modelo SVR e uma variante da MLP treinada com um algoritmo genético. No trabalho de Dabiri *et al.* (2022b), o LRT de barras não emendadas e emendadas é previsto utilizando regressão não linear<sup>1</sup>, regressão de cumeeira (*ridge regression*) e MLP, alcançando resultados precisos. Em outro trabalho de Dabiri *et al.* (2022a), técnicas de AM baseadas em árvores, a saber, árvores de decisão e RF, são implementadas para analisar a deformação até o ponto de fratura de reforços de aço não emendados e emendados, alcançando resultados aceitáveis. Os autores destacam ainda que os modelos de regressão avaliados baseados em árvores foram econômicos e economizaram tempo em comparação com exames experimentais mais complicados, demorados e caros.

Um modelo de regressão baseado em máquina de aprendizado extremo (*extreme learning machine*, ELM) (Zhang *et al.*, 2020), uma arquitetura de rede neural cujos pesos da camada de entrada para a camada oculta são amostrados aleatoriamente e não podem ser modificados durante o treinamento, é desenvolvido por Xiong *et al.* (2023) para prever as propriedades mecânicas de tiras de liga de alumínio. Os resultados obtidos mostram que o modelo de regressão baseado em ELM proposto alcança alta precisão e estabilidade na previsão do LE, LRT e AP das tiras de liga de alumínio. A principal vantagem da ELM em relação a métodos de AM mais tradicionais, como a MLP e o SVR, reside na rapidez de treinamento desse modelo.

Apesar de sua popularidade entre os algoritmos de AM, as aplicações de modelos de aprendizado profundo (*deep learning*, DL) para prever as propriedades mecânicas do aço ainda estão em seus estágios iniciais. Há a hipótese que isso ocorre devido ao tipo de informação de entrada. A maioria das arquiteturas de DL, especialmente aquelas baseadas em rede neural convolucional (*convolutional neural network*, CNN), requer imagens como entradas. Em tarefas

<sup>1</sup> Os autores não detalham no artigo a que método de regressão não linear eles se referem. Muito provavelmente eles utilizaram para este fim a função do MATLAB® @fitnlm.

de inspeção visual, onde a aquisição de imagens é comum, a aplicação de CNN é atualmente dominante. Por exemplo, no trabalho de Li *et al.* (2021), uma abordagem de DL baseada em YOLOv3 (Terven *et al.*, 2023) é usada para a detecção e contagem automática de barras de aço por meio de imagens. Portanto, para usar uma CNN, é necessário converter os dados de produção em imagens 2D, como feito por Xu *et al.* (2019), para prever as propriedades mecânicas do aço laminado a quente usando composição química e parâmetros do processo.

De particular interesse para a presente tese está o trabalho de Murta *et al.* (2022), que utiliza o modelo regressão por vetores suporte via mínimos quadrados (*least squares support vector regression*, LSSVR) (Suykens e Vandewalle, 1999) para prever as propriedades mecânicas de vergalhões de aço. Esse modelo é construído com 18 entradas e 4 saídas (LE, LRT, razão LRT/LE e AP), com o modelo LSSVR superando o modelo de regressão baseado em MLP (Murta *et al.*, 2021) desenvolvido em seu trabalho anterior no mesmo conjunto de dados. Motivado por este resultado, apresentam-se a seguir os objetivos desta tese.

## 1.1 Objetivos da Tese

O objetivo geral desta pesquisa é revisitar o modelo LSSVR com o intuito de aprimorar seu desempenho na tarefa de previsão de propriedades mecânicas de vergalhões de aço. Nesse contexto, abordagens de esparsificação terão destaques ao lidar com modelos baseados em vetores suporte (VSs), já que a complexidade computacional geralmente cresce de forma cúbica com o tamanho do conjunto de treinamento (Oliveira *et al.*, 2018). Complementar a isso, utiliza-se o paradigma de regressão local para desenvolver uma abordagem inovadora para o modelo LSSVR. Diferentemente da abordagem de modelo de regressão, doravante chamada de regressão global, a modelagem por regressão local divide o conjunto de dados em regiões menores e constrói um modelo de regressão separado para cada região. Esta abordagem tem sido usada com sucesso em diversos problemas de regressão, como na identificação de modos de falha em paredes de cisalhamento de concreto armado (Liang e Xue, 2023), fabricação de compósitos em autoclave (Crawford *et al.*, 2021) e produção de polietileno (Abonyi *et al.*, 2003).

A seguir, são apresentados os objetivos específicos desta tese, que visam aprofundar a compreensão e validar a eficácia das técnicas desenvolvidas:

- Elaborar metodologias de experimentos e testar novas funções de kernel para o modelo LSSVR;
- Avaliar métodos de quantização vetorial (QV) (Ismail *et al.*, 2011; Rocha Neto e Barreto,

2012) e introduzir uma nova abordagem ao método de dependência linear aproximada (*approximate linear dependence*, ALD) (Engel *et al.*, 2004) na esparsificação do modelo LSSVR para dados não temporais;

- Avaliar a modelagem local baseada em agrupamentos (Alpaydin e Jordan, 1996), que se apresenta com uma alternativa interessante para reduzir os custos computacionais de previsão do modelo LSSVR;
- Analisar os métodos propostos, verificando avanços em eficácia e eficiência para o modelo LSSVR como contribuição desta pesquisa.

## 1.2 Produção Científica

As seguintes publicações são correlatas ao presente trabalho realizadas durante o período do doutorado:

- BESSA, R.; BARRETO, G. A. Sobre a aplicação de regressão kernel na predição das propriedades mecânicas de vergalhões de aço: Novos resultados. In: **Anais do XVI Congresso Brasileiro de Inteligência Computacional**. SBIC, 2023. (CBIC 2023). Disponível em: <http://dx.doi.org/10.21528/CBIC2023-125>.
- BESSA, R.; BARRETO, G. A.; COELHO, D. N.; MOURA, E. P. de; MURTA, R. H. F. On least squares support vector regression for predicting mechanical properties of steel rebars. **Metals**, MDPI AG, v. 14, n. 6, p. 695, jun. 2024. ISSN 2075-4701. Disponível em: <http://dx.doi.org/10.3390/met14060695>.

Em Bessa e Barreto (2023), avalia-se a metodologia do modelo LSSVR e emprega-se técnicas de QV para esparsificação do modelo. Já em Bessa *et al.* (2024), são introduzidos o método ALD modificado para poda de VSs e a modelagem local da LSSVR para previsão de propriedades mecânicas de vergalhões de aço.

## 1.3 Organização da Tese

A presente tese é estruturada em capítulos que abordam de maneira sequencial os aspectos fundamentais, metodológicos e experimentais do estudo. Cada capítulo é dedicado a uma etapa específica da pesquisa, proporcionando uma compreensão completa e detalhada do trabalho realizado. A seguir, é apresentada uma visão geral dos próximos capítulos:

- No Capítulo 2, revisa-se brevemente o conjunto de dados da fabricação de vergalhões de

ação e suas origens a partir de sua produção para modelagem das respectivas propriedades mecânicas;

- No Capítulo 3, descrevem-se os fundamentos do modelo LSSVR, além de avaliar métodos propostos anteriormente e sugerir novas abordagens;
- No Capítulo 4, apresentam-se os métodos de redução de vetores suporte usados pelos modelos de regressão baseados em LSSVR;
- No Capítulo 5, são apresentados os fundamentos da modelagem local baseada em agrupamentos;
- No Capítulo 6, conclui-se a tese revisando seus objetivos e propondo possíveis trabalhos futuros.

## 2 SOBRE O CONJUNTO DE DADOS

A importância e a singularidade do conjunto de dados utilizado nesta tese justificam a dedicação de um capítulo inteiro a ele. O conjunto de dados foi fornecido por uma siderúrgica brasileira semi-integrada, que produz aço a partir de sucata ferrosa.

O tratamento térmico utilizado é denominado THERMEX pela empresa, sendo conhecido na literatura como Processo Tempcore (Bahleda *et al.*, 2019). Este estudo aborda a previsão das propriedades do aço para vergalhões, uma tarefa específica de regressão baseada em informações sobre sua composição química (12 variáveis constitucionais) e parâmetros do tratamento térmico de laminação (6 variáveis de processo). Essas 18 variáveis de entrada são empregadas pelos modelos de regressão para prever as seguintes variáveis de saída: LE, LRT, razão LRT/LE e AP, totalizando 1300 instâncias no conjunto de dados.

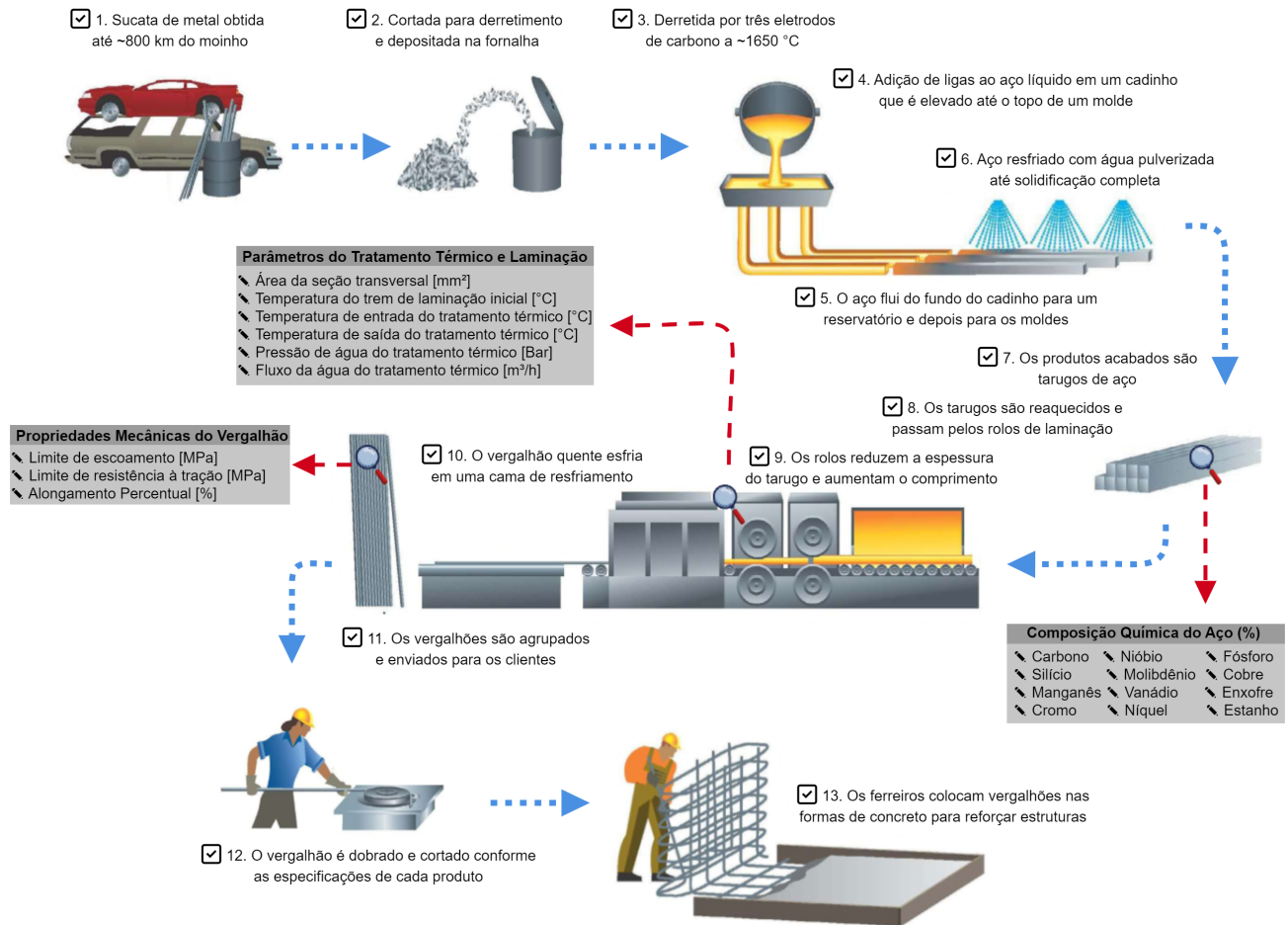
Ressalta-se que tais propriedades mecânicas são de fundamental importância para o setor siderúrgico e para a engenharia mecânica, uma vez que determinam a qualidade e a aplicabilidade final do produto. Isto posto, neste capítulo são discutidas as principais características do conjunto e do processo siderúrgico a partir do qual as medidas foram obtidas.

### 2.1 Sobre a Produção dos Vergalhões de Aço

O vergalhão de aço carbono pode ser fabricado a partir de sucata que não contém material virgem. Essa matéria-prima possui diversas origens e fornecedores. Conseqüentemente, há grande variação em sua composição, o que exige a correção da composição do aço antes do vazamento e do envio para a laminação. Os metais reciclados, juntamente com adições de ligas, são fundidos em um forno de arco elétrico para obter aço líquido, que é então solidificado em tarugos de aço. Em geral, o aço contém menos de 97% de ferro reciclado, menos de 2% de manganês, menos de 1,5% de cobre, menos de 0,9% de carbono e um total de 1,5% ou menos de níquel, silício, enxofre, estanho, fósforo e vanádio. Posteriormente, os tarugos são reaquecidos em um forno de reaquecimento e entram no laminador para formar vergalhões. Por fim, os vergalhões são enviados a oficinas onde são cortados e dobrados conforme solicitação do cliente (Nieves, 2016). Esse processo é resumido na Figura 1.

Convém destacar que o processo siderúrgico envolve uma ampla gama de informações, tanto relativas à composição química quanto às variáveis de processo. Nesse contexto, realizou-se uma etapa preliminar de avaliação, seleção e redução de variáveis, com o objetivo

Figura 1 – Ilustração do processo de fabricação de vergalhões de aço e aquisição do conjunto de dados.



Fonte: Adaptado de Nieves (2016).

de tornar o modelo mais eficiente (detalhado em Murta (2022)). Assim, o conjunto de dados da fabricação de vergalhões de aço abrange dois conjuntos distintos de variáveis de entrada, totalizando 18 entradas, que são usadas para construir os modelos de regressão.

O primeiro conjunto compreende as variáveis de entrada relacionadas à composição química do aço, a saber, a concentração de carbono, nióbio, fósforo, silício, molibdênio, cobre, manganês, vanádio, enxofre, cromo, níquel e estanho, que são elementos que podem influenciar as propriedades mecânicas das ligas. A Tabela 1 apresenta a descrição estatística da composição química do conjunto de dados de aço utilizado nesta pesquisa em termos de valor médio, máximo, mínimo e desvio padrão. O segundo grupo engloba os parâmetros do processo de laminação a quente. Desse modo, as seis variáveis de entrada desse grupo são a área da seção transversal da barra de aço após a laminação; a temperatura final de laminação; as temperaturas inicial e final

Tabela 1 – Descrição estatística da composição química das amostras [%].

Variáveis de Entrada	Mínimo	Máximo	Média	Desvio Padrão
Carbono	0,156	0,474	0,3216	0,039
Nióbio	0,000	0,008	0,0026	0,001
Fósforo	0,002	0,013	0,0064	0,002
Silício	0,099	0,208	0,1567	0,021
Molibdênio	0,006	0,029	0,0111	0,002
Cobre	0,187	0,445	0,2814	0,040
Manganês	0,456	1,149	0,9001	0,229
Vanádio	0,002	0,019	0,0081	0,006
Enxofre	0,020	0,050	0,0370	0,006
Cromo	0,056	0,271	0,1314	0,036
Níquel	0,049	0,143	0,0745	0,011
Estanho	0,001	0,040	0,0193	0,003

Fonte: Elaborado por Murta *et al.* (2021).

Tabela 2 – Descrição estatística dos parâmetros do tratamento térmico.

Variáveis de Entrada	Mínimo	Máximo	Média	Desvio Padrão
Área da seção transversal [ $mm^2$ ]	50	199	93,2	46,5
Temperatura de laminação final [ $^{\circ}C$ ]	742,7	1081,5	1031,9	21,9
Temperatura inicial de tratamento térmico [ $^{\circ}C$ ]	654,7	961,8	920,7	21,1
Temperatura final de tratamento térmico [ $^{\circ}C$ ]	337,6	584,1	436,8	47,5
Pressão da água de tratamento térmico [ $Bar$ ]:	0	15,5	5,8	5,0
Fluxo da água de tratamento térmico [ $m^3/h$ ]:	0	68,2	29,2	23,7

Fonte: Elaborado por Murta *et al.* (2021).

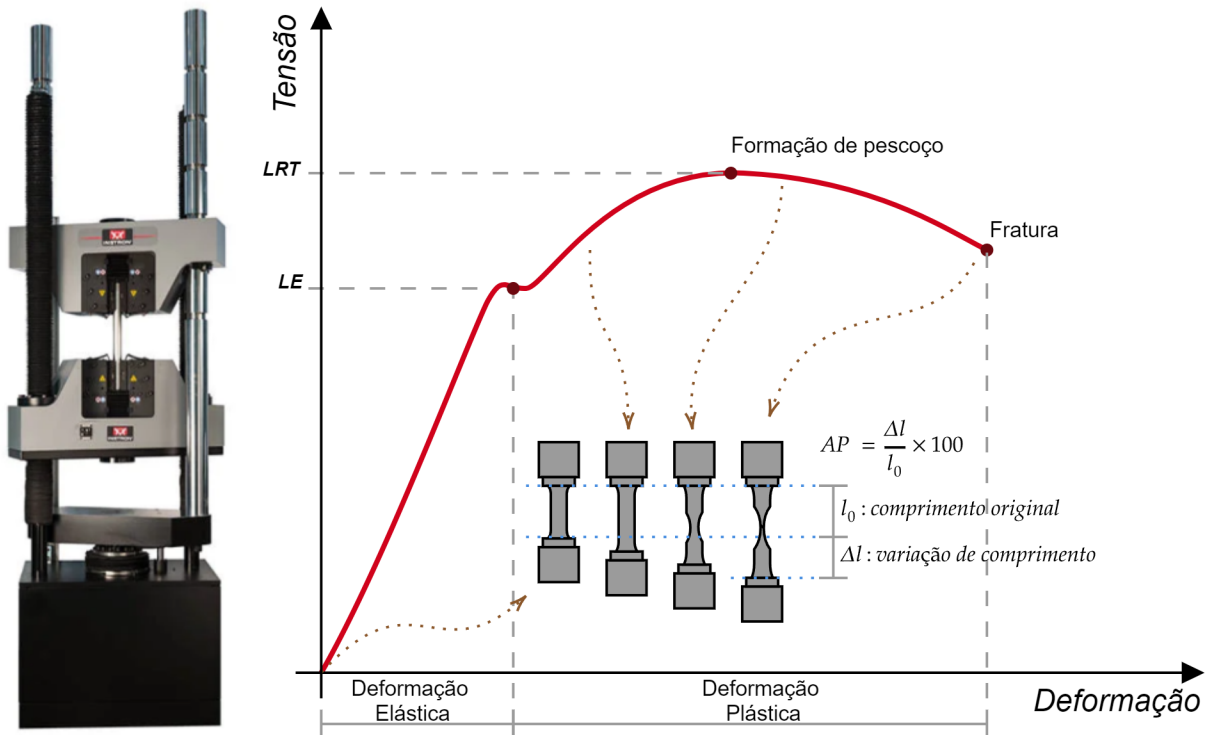
do tratamento térmico; e a pressão e fluxo da água de tratamento térmico. A Tabela 2 fornece informações sobre a faixa dos parâmetros de laminação a quente e tratamento térmico aplicados.

O vergalhão é produzido a partir da laminação de um tarugo com seção transversal de aproximadamente  $100 \times 100 \text{ mm}$ , ou seja,  $10.000 \text{ mm}^2$ . Durante o processo de laminação, esse tarugo é progressivamente deformado até atingir as dimensões finais especificadas para cada bitola. Após a laminação, as áreas das seções transversais dos vergalhões analisados correspondem a  $50 \text{ mm}^2$ ,  $71 \text{ mm}^2$ ,  $78 \text{ mm}^2$ ,  $113 \text{ mm}^2$ ,  $129 \text{ mm}^2$  e  $199 \text{ mm}^2$ . Os respectivos diâmetros foram de  $8 \text{ mm}$ ,  $3/8''$  ( $9,525 \text{ mm}$ ),  $10 \text{ mm}$ ,  $12 \text{ mm}$ ,  $1/2''$  ( $12,7 \text{ mm}$ ) e  $5/8''$  ( $15,875 \text{ mm}$ ).

Quanto menor a seção transversal do vergalhão obtido, maior é o grau de deformação plástica sofrido pelo material durante a laminação. Esse fator exerce forte influência na microestrutura e, conseqüentemente, nas propriedades mecânicas do produto final, especialmente no limite de resistência.

Além disso, a temperatura final de laminação, assim como as temperaturas inicial e final do tratamento térmico, desempenham papel decisivo na definição da microestrutura do aço. Já a pressão da água utilizada no processo de têmpera influencia diretamente a taxa de resfriamento, o que afeta tanto o gradiente de temperatura entre a superfície e o núcleo quanto a

Figura 2 – Curva tensão-deformação de ensaio de tração até a fratura.



Fonte: Adaptado de Callister e Rethwisch (2015).

espessura da camada externa de martensita formada.

## 2.2 Sobre as Propriedades Mecânicas de Vergalhões de Aço

Para determinar essas propriedades mecânicas, utiliza-se uma máquina de ensaio de tração. Esse equipamento aplica uma carga axial crescente ao vergalhão até a ruptura. Durante o ensaio, registram-se as forças aplicadas e as deformações correspondentes, gerando uma curva tensão-deformação apresentada na Figura 2. Esse ensaio é fundamental para garantir a qualidade e a segurança dos vergalhões utilizados em obras de engenharia civil, proporcionando dados precisos sobre o comportamento mecânico do material sob diferentes condições de carga (Callister e Rethwisch, 2015).

A medida LE é a tensão máxima que o vergalhão pode suportar antes de sofrer deformação plástica. Esse parâmetro é crucial para determinar a capacidade de carga do material em situações de serviço normal. Após ultrapassar esse limiar, a tensão necessária para continuar a deformação plástica nos metais aumenta até um valor máximo, o ponto de LRT, e então diminui

Tabela 3 – Sumário estatístico das saídas observadas referentes à propriedades mecânicas dos vergalhões produzidos.

Variáveis de Saída	Mínimo	Máximo	Média	Desvio Padrão
LE [MPa]	421	661	521,7	41,6
LRT [MPa]	596	914	682,8	35,1
Razão LRT/LE	1,12	1,55	1,314	0,072
AP [%]	8	20	15,1	1,897

Fonte: Elaborado por Murta *et al.* (2021).

até a eventual fratura do material. Toda deformação até esse ponto distribui-se uniformemente por toda a região estreita do corpo de prova. Quando a tensão atinge seu valor máximo, uma pequena constrição, ou "pescoço", começa a se formar em algum ponto, concentrando toda a deformação subsequente. Esse fenômeno, chamado estrição, resulta na fratura do material nesse local específico (Black e Kohser, 2021; Callister e Rethwisch, 2015).

A ductilidade é uma propriedade mecânica crucial que mede o grau de deformação plástica até a fratura. Um metal que apresenta pouca ou nenhuma deformação plástica antes de fraturar é considerado frágil. A ductilidade pode ser expressa quantitativamente com a medida AP. A razão LRT/LE também indica a ductilidade do vergalhão. Valores mais altos dessa razão sugerem um material capaz de absorver mais energia antes da ruptura, aumentando sua segurança e resistência a falhas catastróficas (Murta, 2022). A Tabela 3 apresenta a descrição estatística das propriedades mecânicas das amostras analisadas neste estudo.

### 2.3 Estado da Arte

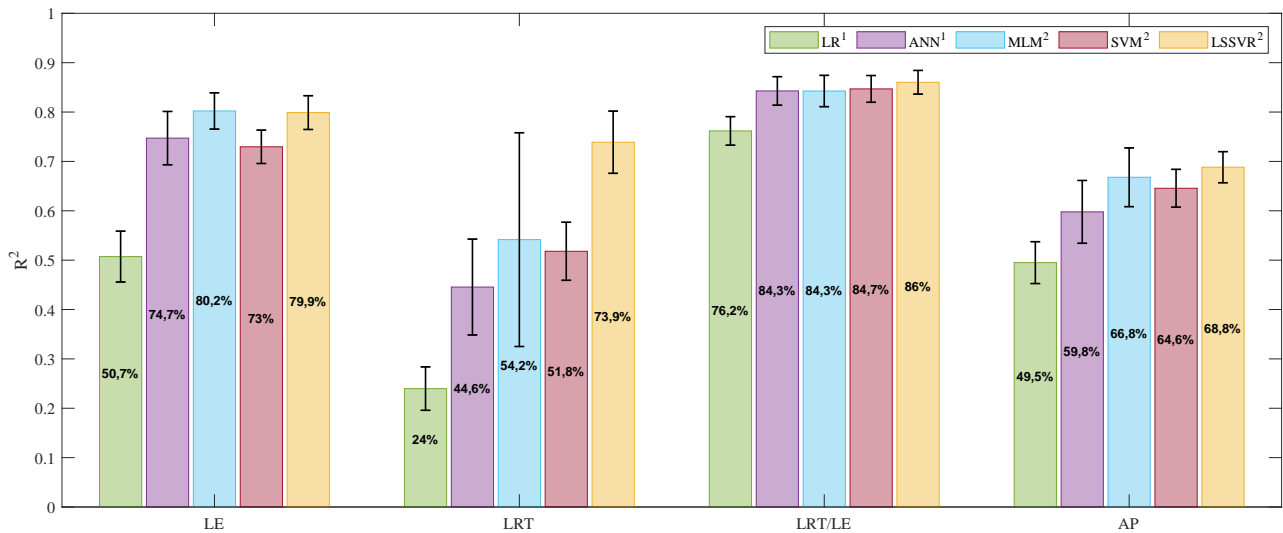
Até o momento, dois estudos apresentam resultados para esse conjunto de dados empregando diferentes modelos de regressão com abordagem em AM. No primeiro, Murta *et al.* (2021) utilizam os modelos de LR e baseado em MLP. Em publicação seguinte, Murta *et al.* (2022) avaliam os modelos baseados na máquina de aprendizado mínimo (*minimum learning machine*, MLM) (Souza *et al.*, 2015), SVR e LSSVR que apresentam progressos em termo do coeficiente de determinação ( $R^2$ ).

A figura de mérito  $R^2$ , com objetivo de comparar os desempenhos dos modelos de regressão, apresenta-se como umas das métricas mais informativas nessa tarefa (Chicco *et al.*, 2021). Sua formulação é dada por

$$R^2 = 1 - \frac{\sum_{n=1}^{N_{it}} (y_n - \hat{y}_n)^2}{\sum_{n=1}^{N_{it}} (y_n - \bar{y})^2}, \quad (2.1)$$

em que  $y_n \in \mathbb{R}$  é o valor observado do  $n$ -ésimo elemento do conjunto de teste  $S_{tt}$ , com o total de

Figura 3 – Comparação geral de desempenho dos modelos avaliados em valores de  $R^2$  médio e de desvio padrão para 100 experimentos.



<sup>1</sup> Murta *et al.* (2021); <sup>2</sup> Murta *et al.* (2022)

Fonte: Adaptado de Murta *et al.* (2022).

$N_{it}$  elementos, e  $\hat{y}_n$  o respectivo valor previsto pelo modelo. Ademais,  $\bar{y} = \frac{1}{N_{it}} \sum_{n=1}^{N_{it}} y_n$  representa a média aritmética dos valores das saídas observadas do conjunto de teste. Pode-se mostrar que o coeficiente de determinação  $R^2$  é o quadrado da correlação entre os valores observados  $\{y_n\}$  e os valores preditos  $\{\hat{y}_n\}$ .

O coeficiente de determinação pode ser interpretado como a proporção da variância na variável dependente que é previsível a partir das variáveis independentes. Além disso, pode-se assumir valores de  $R^2$  na faixa de  $(-\infty, 1]$  de acordo com a relação mútua entre os dados observados e o modelo de previsão, onde o melhor valor possível é  $R^2 = +1$ . Entretanto, uma limitação surge no caso de valores negativos de  $R^2$ : isso indica um desempenho muito ruim do modelo, mas não fornece uma medida direta de quão ruim é esse desempenho (Chicco *et al.*, 2021).

A Figura 3 reúne os resultados alcançados pelos modelos testados em 100 divisões aleatórias e independentes de 80% para treinamento e 20% para teste do conjunto de dados (80%/20%), que são apresentados em valor médio e desvio padrão de  $R^2$ . Como mencionado anteriormente, o modelo LSSVR se destaca dos demais devido ao seu desempenho superior na previsão das propriedades mecânicas.

## **2.4 Conclusão do Capítulo**

O presente capítulo detalha a fabricação dos vergalhões de aço e suas propriedades, com foco na curva tensão-deformação. São apresentadas métricas e resultados de diversas técnicas de modelagem testadas. A partir do próximo capítulo, esta tese se concentra em aprofundar o entendimento do modelo LSSVR e em aprimorá-lo. No Capítulo 3, serão discutidos os fundamentos desse modelo e novas propostas metodológicas para seu desenvolvimento.

### 3 REGRESSÃO POR MÁQUINAS DE KERNEL

Neste capítulo, descrevem-se os fundamentos do modelo LSSVR, além de avaliar métodos propostos anteriormente e sugerir novas abordagens. Inicialmente, realiza-se uma análise detalhada da *toolbox* utilizada em trabalhos anteriores, examinando suas funcionalidades e limitações. Em seguida, compara-se a implementação própria do modelo LSSVR com a *toolbox*. Além disso, testam-se funções de kernel que não haviam sido previamente exploradas para este conjunto de dados, buscando identificar potenciais melhorias das previsões. Apresentam-se, também os resultados obtidos, descrevendo detalhadamente o desempenho do modelo em termos de acurácia e eficiência.

#### 3.1 Regressão por Vetores Suporte via Mínimos Quadrados

O modelo LSSVR é uma extensão do modelo SVR, que reformula seu problema de otimização via método dos mínimos quadrados. A construção desse modelo de regressão, representado por  $f(\cdot)$ , pressupõe a disponibilidade de um conjunto de dados do processo  $\mathcal{S}_{tr} := \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^{N_{tr}}$  com  $N_{tr}$  amostras para seu treinamento, sendo  $\mathbf{x}_n \in \mathbb{R}^{N_x}$  o  $n$ -ésimo vetor de variáveis preditoras de dimensão  $N_x$  e  $\mathbf{y}_n \in \mathbb{R}^{N_y}$  o  $n$ -ésimo vetor das correspondentes saídas observadas de dimensão  $N_y$ . Sem perda de generalidade, pode-se considerar uma única saída ( $N_y = 1$ ) para descrever a abordagem para métodos baseados em kernel, de modo que

$$f(\mathbf{x}) = \mathbf{w}^\top \phi(\mathbf{x}) + b, \quad (3.1)$$

onde  $\phi(\cdot) : \mathbb{R}^{N_x} \rightarrow \mathbb{R}^H$  é um mapeamento não linear em um espaço de alta dimensão  $H$ ,  $\mathbf{w} \in \mathbb{R}^H$  é o vetor de parâmetros e  $b \in \mathbb{R}$  é o intercepto da função.

Especificamente, os parâmetros do modelo LSSVR são estimados minimizando a função de perda

$$J_p(\mathbf{w}, \mathbf{e}) = \frac{1}{2} \mathbf{w}^\top \mathbf{w} + \frac{\gamma}{2} \sum_{n=1}^{N_{tr}} e_n^2, \quad (3.2)$$

sujeito a

$$y_n = \mathbf{w}^\top \phi(\mathbf{x}_n) + b + e_n, \quad (3.3)$$

onde  $e_n = y_n - f(\mathbf{x}_n)$  é o erro associado à  $n$ -ésima amostra, e  $\gamma > 0$  é uma constante de regularização a ser avaliada, considerando tanto o erro de treinamento quanto a suavidade da função de regressão (a norma do vetor  $\mathbf{w}$ ).

Esse problema de otimização pode ter sua formulação no espaço dual descrita pelo lagrangiano

$$\begin{aligned}\mathcal{L}(\mathbf{w}, b, \mathbf{e}, \boldsymbol{\alpha}) &= J_p(\mathbf{w}, \mathbf{e}) - \sum_{n=1}^{N_{tr}} \alpha_n \left[ \mathbf{w}^\top \phi(\mathbf{x}_n) + b + e_n - y_n \right] \\ &= \frac{1}{2} \mathbf{w}^\top \mathbf{w} + \frac{\gamma}{2} \sum_{n=1}^{N_{tr}} e_n^2 - \sum_{n=1}^{N_{tr}} \alpha_n \left[ \mathbf{w}^\top \phi(\mathbf{x}_n) + b + e_n - y_n \right],\end{aligned}\quad (3.4)$$

em que  $\boldsymbol{\alpha} \in \mathbb{R}^{N_{tr}}$  representa o vetor de multiplicadores de Lagrange. Portanto, as condições de otimalidade para (3.4) são fornecidas por

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial \mathbf{w}} = 0 \rightarrow \mathbf{w} = \sum_{n=1}^{N_{tr}} \alpha_n \phi(\mathbf{x}_n), \\ \frac{\partial \mathcal{L}}{\partial b} = 0 \rightarrow \sum_{n=1}^{N_{tr}} \alpha_n = 0, & n = 1, \dots, N_{tr}, \\ \frac{\partial \mathcal{L}}{\partial e_n} = 0 \rightarrow \alpha_n = \gamma e_n, \\ \frac{\partial \mathcal{L}}{\partial \alpha_k} = 0 \rightarrow \mathbf{w}^\top \phi(\mathbf{x}_n) + b + e_n - y_n = 0, & k = 1, \dots, N_{tr}. \end{cases}\quad (3.5)$$

Ao eliminar  $e_n$  e  $\mathbf{w}$  em (3.5), obtém-se a solução a partir do seguinte sistema de Karush-Kuhn-Tucker, reescrevendo o problema em um sistema linear

$$\begin{bmatrix} 0 & \mathbf{1}_{N_{tr}}^\top \\ \mathbf{1}_{N_{tr}} & \boldsymbol{\Omega} + \gamma^{-1} \mathbf{I}_{N_{tr}} \end{bmatrix} \begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{y} \end{bmatrix},\quad (3.6)$$

onde  $\mathbf{y} = [y_1, \dots, y_{N_{tr}}]^\top$ ,  $\mathbf{1}_{N_{tr}} \in \mathbb{R}^{N_{tr}}$  é um vetor de 1's,  $\mathbf{I}_N \in \mathbb{R}^{N_{tr} \times N_{tr}}$  é uma matriz identidade, e  $\boldsymbol{\Omega} \in \mathbb{R}^{N_{tr} \times N_{tr}}$  é a matriz de kernel com elementos  $\Omega_{i,j} = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j) \quad \forall i, j \in \{1, \dots, N_{tr}\}$ .

Uma forma de emular o produto interno de um mapeamento não linear em um espaço de alta dimensão é usando o truque do kernel, em que se emprega uma função que satisfaz as condições de Mercer<sup>1</sup> (Vapnik, 1999). Por exemplo, a função gaussiana escolhida no trabalho de Murta *et al.* (2022) para os experimentos, dada por

$$\Omega_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j) = \exp \left\{ -\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma^2} \right\},\quad (3.7)$$

onde  $k(\cdot, \cdot)$  é a função de kernel,  $\|\cdot\|_2$  é a norma euclidiana e  $\sigma \in \mathbb{R}$  denota a largura da função gaussiana (um hiperparâmetro a ser ajustado). Com esta solução, o modelo de regressão resultante é expresso como

$$f(\mathbf{x}) = \sum_{n=1}^{N_{tr}} \alpha_n k(\mathbf{x}_n, \mathbf{x}) + b.\quad (3.8)$$

<sup>1</sup> Garantir que a matriz  $\boldsymbol{\Omega}$  seja simétrica e positiva definida.

Vale ressaltar que a técnica regressão de cumeieira kernel (*kernel ridge regression*) (Saunders *et al.*, 1998; Murphy, 2012) pode ser entendida como uma versão simplificada do LSSVR, que consiste operacionalmente na desconsideração do intercepto  $b$  na Equação 3.6 e da restrição  $\sum_{n=1}^N \alpha_n = 0$ . Dessa maneira, a solução do sistema linear é dada por

$$\boldsymbol{\alpha} = (\boldsymbol{\Omega} + \gamma \mathbf{I})^{-1} \mathbf{y}. \quad (3.9)$$

Ao contrário do modelo SVR, que produz uma solução esparsa na sua construção, utilizando apenas alguns vetores de suporte do conjunto de treinamento (denominados VSs), o modelo LSSVR frequentemente utiliza todos os dados disponíveis. Em outras palavras, todos os exemplos do conjunto de treinamento, representado por  $\mathcal{X}_{tr} := \{\mathbf{x}_n\}_{n=1}^{N_{tr}}$ , devem ser armazenados, o que equivale a considerar como VSs todos os exemplos de treinamento ao construir o modelo preditivo. Assim, diversas técnicas são empregadas para reduzir o número de VSs e simplificar a estrutura interna do modelo LSSVR (Suykens *et al.*, 2002; Rocha Neto e Barreto, 2012; Zheng *et al.*, 2024). No Capítulo 4, algumas dessas técnicas são apresentadas para comparação de desempenho.

### 3.2 Experimentos Preliminares com a *LS-SVMlab Toolbox*

A otimização de hiperparâmetros é uma etapa crítica na construção de um modelo baseado em kernel. Em Murta *et al.* (2022), a *LS-SVMlab Toolbox* (versão 1.8), escrita em MATLAB<sup>®</sup>, é utilizada para implementar o modelo LSSVR com o kernel gaussiano. Na versão atual dessa *toolbox*, os hiperparâmetros ( $\gamma$  e  $\sigma$ ) são ajustados utilizando o método de recozimento simulado acoplado (*coupled simulated annealing*, CSA) (Souza *et al.*, 2010) e o método *Simplex*. Inicialmente, o algoritmo CSA é responsável por explorar o espaço de busca (*explore phase*). Em seguida, o método *Simplex* refina a solução (*exploit phase*), reduzindo o número de avaliações da função objetivo e melhorando a qualidade da solução em comparação com o uso isolado desses algoritmos (Brabanter *et al.*, 2011).

Nos experimentos descritos por Murta *et al.* (2022), apesar da divisão entre treinamento e teste do conjunto de dados ser de 80%/20%, apenas 20% das amostras (ou seja, 25% das amostras de  $\mathcal{S}_{tr}$ ) são utilizadas na etapa de otimização dos hiperparâmetros. Essa escolha proporciona rapidez no treinamento, mas conforme notado em experimentos preliminares levados a cabo durante a presente pesquisa variando a proporção dos dados de treinamento para ajuste do modelo (25%, 50%, 75% e 100%), observa-se uma melhoria significativa nos valores de  $R^2$

para as quatro saídas avaliadas. Essa comparação é apresentada na Figura 4 por meio de gráficos *boxplot* para 100 execuções independentes de experimentos, cada uma com uma divisão aleatória dos dados. Adicionalmente ao *boxplot*, os pontos em azul superpostos ao *boxplot* formam um gráfico de enxame (*swarm plot*) e representam as observações individuais dos valores de  $R^2$ , em que a posição horizontal de cada ponto é ajustada para evitar sobreposição com outros pontos, permitindo uma melhor visualização da densidade de dados em diferentes níveis.

Em termos de  $R^2$  médio, observa-se um aumento de 5,03% (de 0,7989 para 0,8391) para a saída LE, de 5,14% (de 0,7389 para 0,7769) para a saída LRT, de 1,97% (de 0,8603 para 0,8772) para a saída da razão LRT/LE e de 3,36% (de 0,6883 para 0,7114) para a saída da AP em comparação aos resultados apresentado por Murta *et al.* (2022). Portanto, torna-se interessante utilizar todos os dados do conjunto  $\mathcal{S}_{tr}$  na sintonia dos hiperparâmetros, mesmo diante do aumento no custo computacional. Por isso, adota-se o emprego de todo o conjunto de treinamento para a otimização dos hiperparâmetros e a mesma proporção de 80%/20% para a construção de todos os modelos de regressão baseados em LSSVR propostos nesta tese.

A *LS-SVMlab Toolbox* conta com a implementação de duas outras funções de kernel, a saber, o kernel linear expresso por

$$k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^\top \mathbf{x}_j, \quad (3.10)$$

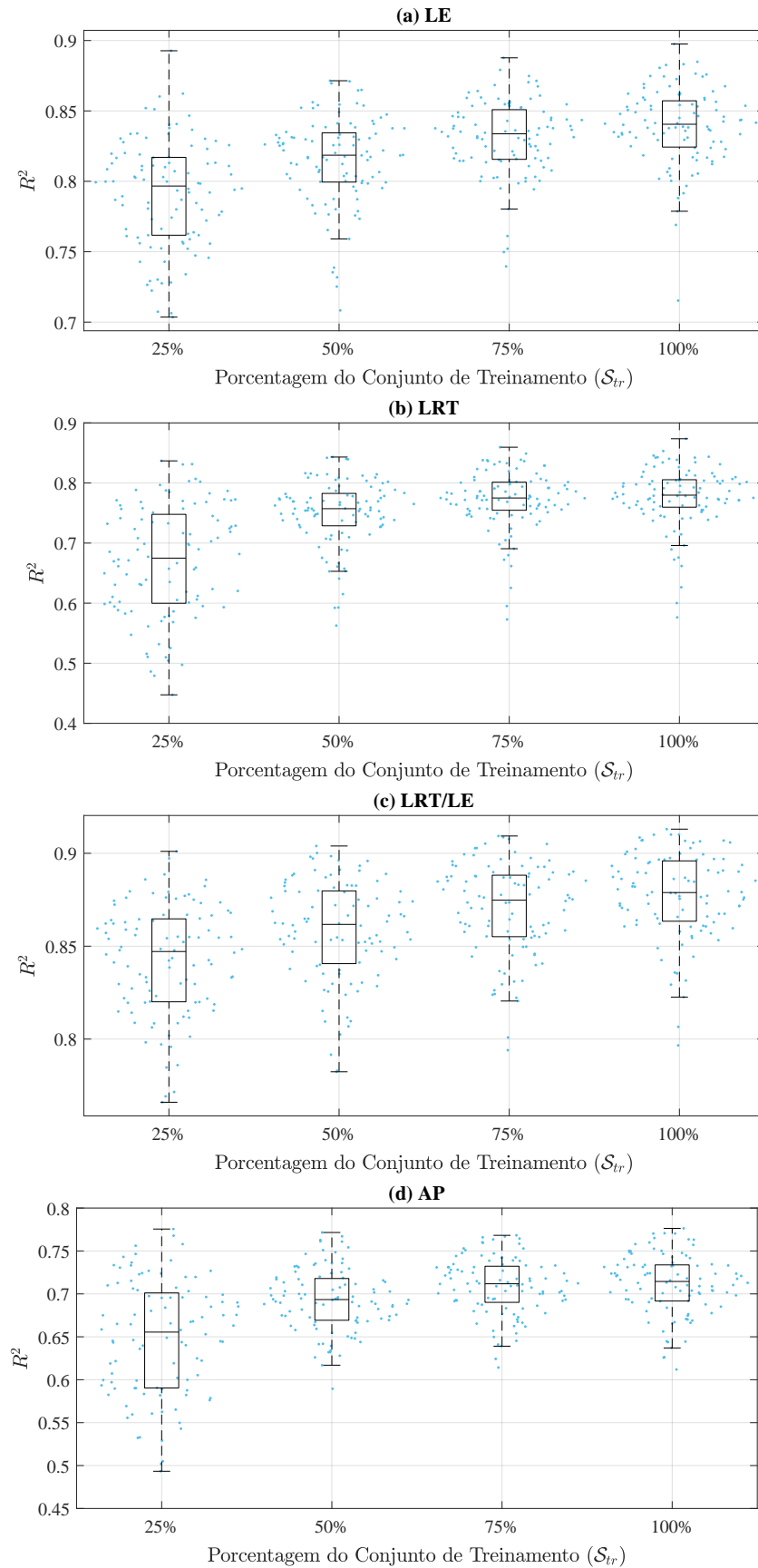
e pelo kernel polinomial dado por

$$k(\mathbf{x}_i, \mathbf{x}_j) = \left( \mathbf{x}_i^\top \mathbf{x}_j + c \right)^d, \quad (3.11)$$

em que  $c \in \mathbb{R}$  é o intercepto da função e  $d \in \mathbb{Z}_+^*$  é o grau do polinômio. Com isso, a *LS-SVMlab Toolbox* ajusta apenas o hiperparâmetro  $\gamma$  para o modelo com kernel linear e, para o com kernel polinomial, os valores de  $\gamma$ ,  $c$  e  $d$ . Portanto, o grau do polinômio é selecionado pelo algoritmo de otimização em cada teste.

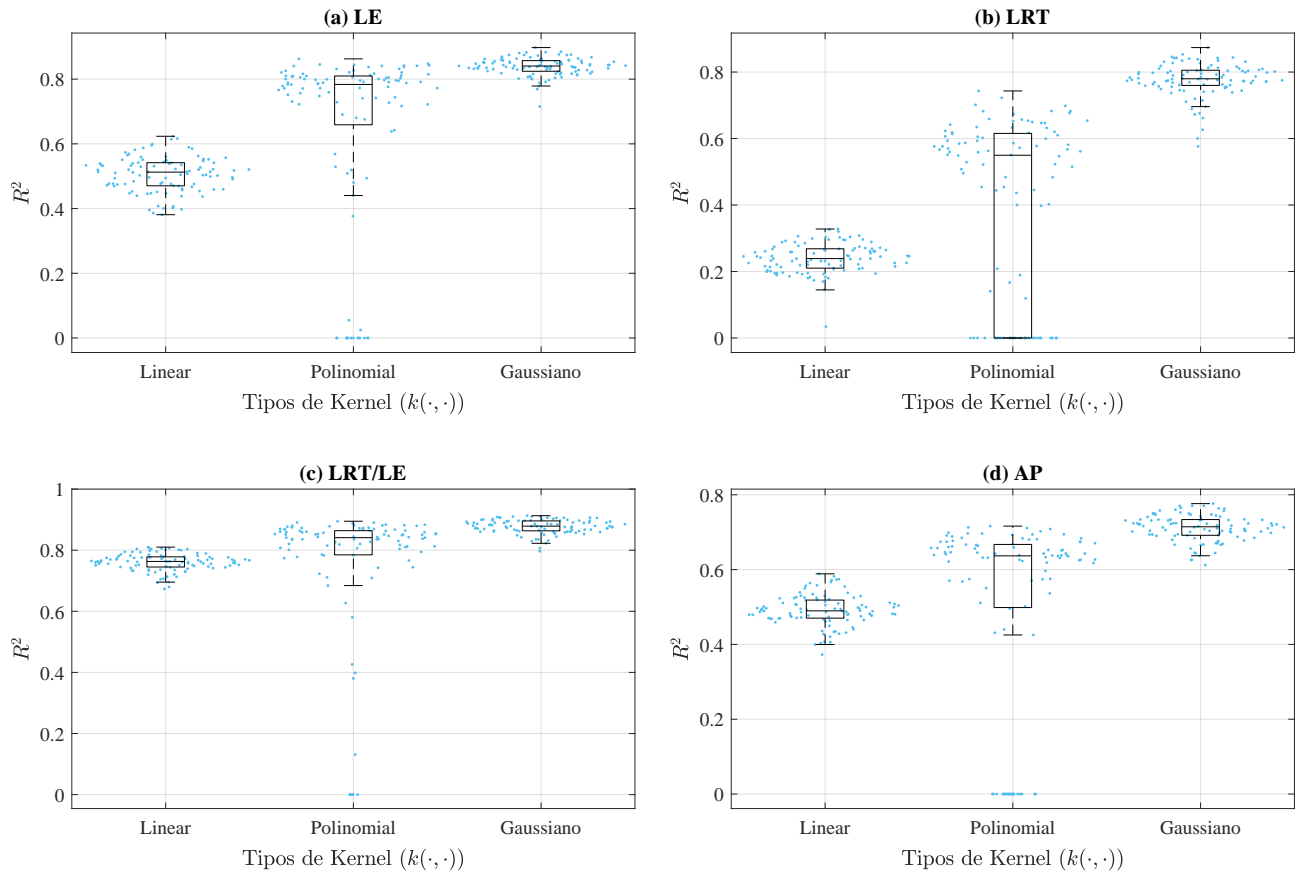
Os experimentos realizados para esses kernels são apresentados na Figura 5. Vale ressaltar que os resultados para o kernel polinomial registram valores de  $R^2$  negativos, os quais são saturados em zero. Os valores numéricos de média e de desvio padrão de  $R^2$  são exibidos na Tabela 4, indicando a superioridade do kernel gaussiano em modelar relações não lineares entre as variáveis.

Figura 4 – *Boxplot* de desempenho utilizando a *LS-SVMlab Toolbox* na sintonia dos hiperparâmetros do kernel gaussiano para diferentes proporções do conjunto de treinamento.



Fonte: Autoria própria.

Figura 5 – *Boxplot* de desempenho para diferentes tipos de kernel utilizando a *LS-SVMlab Toolbox*.



Fonte: Autoria própria.

Tabela 4 – Comparação de desempenho entre diferentes funções de kernel da *LS-SVMlab Toolbox*.

Tipos de Kernel	LE		LRT		LRT/LE		AP	
	$R^2_{mean}$	$R^2_{std}$	$R^2_{mean}$	$R^2_{std}$	$R^2_{mean}$	$R^2_{std}$	$R^2_{mean}$	$R^2_{std}$
Linear	0,5058	0,0539	0,2396	0,0447	0,7605	0,0285	0,4917	0,0421
Polinomial	0,6455	0,2854	0,4043	0,2714	0,7450	0,2483	0,5018	0,2659
Gaussiano	0,8391	0,0275	0,7769	0,0506	0,8772	0,0231	0,7114	0,0352

Fonte: Autoria própria.

### 3.3 Implementação customizada do modelo LSSVR

O intuito de desenvolver uma implementação própria do modelo LSSVR é obter mais liberdade e facilidade para propor melhorias. Além de ser um exercício de aprendizado da técnica, resulta em um código mais legível e aplicável para fins científicos com o foco em regressão. Diferentemente da *LS-SVMlab Toolbox*, opta-se por combinar a metaheurística PSO (Kennedy e Eberhart, ; Rustam e Kintandani, 2019) com o método *Simplex* de Nelder-Mead (Nelder e Mead, 1965; Lagarias *et al.*, 1998) na tarefa de sintonização dos hiperparâmetros do

modelo. Todo o código é desenvolvido em MATLAB<sup>®</sup> (R2023b) e está disponível no repositório GitHub<sup>2</sup>.

Para a validação da implementação, um algoritmo PSO global (descrito no Apêndice A) é construído com um enxame composto por cinco partículas, cada uma com duas posições para a sintonia do modelo LSSVR com kernel gaussiano. O intervalo de busca para os dois hiperparâmetros desse kernel é  $\gamma^{-1}, \sigma \in [e^{-7.5}, e^{7.5}]$ . Após a inicialização aleatória do enxame, cinco iterações são computadas e o melhor resultado obtido serve como ponto inicial para o método *Simplex*. A função `@fminsearch` do MATLAB<sup>®</sup> é empregada para refinar a solução encontrada pela melhor partícula do PSO, com um limite de 100 iterações em caso de não convergência. Para essa otimização, é aplicada uma validação cruzada de 10 partes (*10-fold cross-validation*) ao conjunto de dados de treinamento, utilizando o erro quadrático médio como função de custo a ser minimizada.

São realizadas 100 rodadas de validação, mantendo a proporção treinamento/teste estabelecida na seção anterior (80%/20%). Todos os modelos utilizam as mesmas 100 divisões em todos os experimentos. Além disso, a normalização *z-score* é aplicada a ambos os grupos usando apenas as estatísticas do conjunto de treinamento, e as saídas previstas dos modelos são desnormalizadas para a avaliação de desempenho.

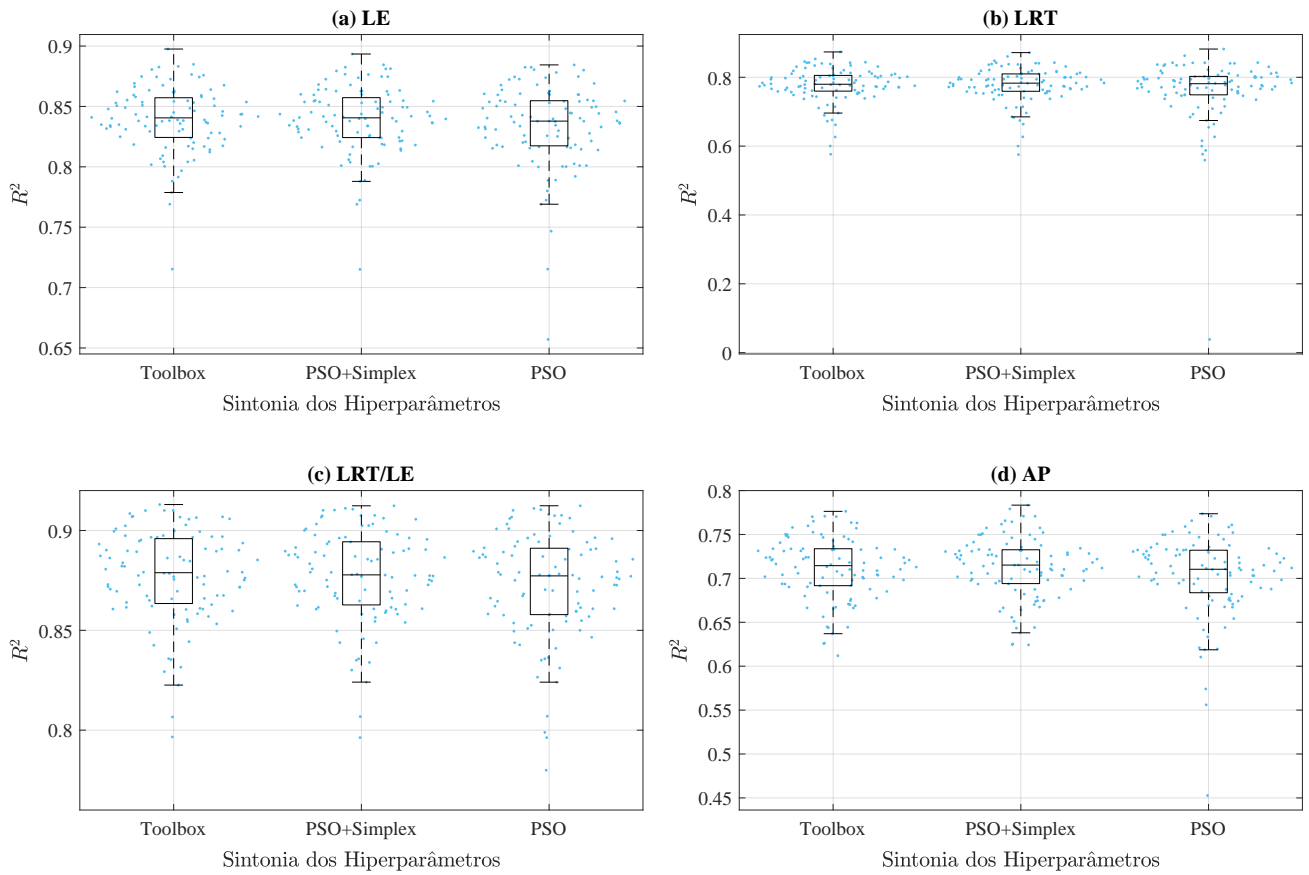
Os *boxplots* apresentados na Figura 6 comparam os resultados obtidos entre a *toolbox* e a implementação proposta. Além disso, avalia-se o comportamento ao utilizar apenas o PSO com 25 rodadas (5 partículas  $\times$  20 iterações extras = 100 avaliações da função custo) na busca dos hiperparâmetros, em comparação com o limite de 100 avaliações da função custo do método *Simplex*, que não é atingido nenhuma vez nesses experimentos. Dessa forma, observa-se que a combinação do PSO com o *Simplex* apresenta desempenho semelhante ao da *toolbox*, conciliando boa precisão e estabilidade. Além disso, o método híbrido supera o uso isolado do PSO, que mostra maior variabilidade nos resultados.

### 3.4 Explorando Diferentes Funções de Kernel

Além das funções de kernel presentes na *toolbox*, outras funções também são amplamente utilizadas e preconizadas na literatura. Cada uma dessas funções possui características específicas que podem ser vantajosas dependendo da natureza dos dados e do problema em questão. A escolha adequada do kernel pode impactar significativamente o desempenho do

<sup>2</sup> [https://github.com/renan-bessa/tese\\_ufc\\_2024](https://github.com/renan-bessa/tese_ufc_2024)

Figura 6 – *Boxplot* de desempenho para diferentes formas de sintonia de hiperparâmetros.



Fonte: Autoria própria.

modelo, proporcionando maior exatidão e precisão nas previsões (Ayat *et al.*, 2002; Coelho *et al.*, 2017; Khozani *et al.*, 2019; Zhou *et al.*, 2023; Malang *et al.*, 2023; Ngu *et al.*, 2024).

Por esse motivo, alguns kernels alternativos são escolhidos para análise sob a hipótese de que podem produzir resultados superiores àqueles apresentados pelo uso do kernel gaussiano. As seguintes funções contém até dois hiperparâmetros para otimização, a saber,  $\rho_h, \sigma_h \in \mathbb{R}_+^*$ . A primeira delas é a função sigmoide denotada por

$$k(\mathbf{x}_i, \mathbf{x}_j) = \tanh\left(\rho_h \mathbf{x}_i^\top \mathbf{x}_j - 1\right); \quad (3.12)$$

em seguida, a função exponencial, expressa por

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left\{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2}{\sigma_h}\right\}; \quad (3.13)$$

e a função logarítmica, dada por

$$k(\mathbf{x}_i, \mathbf{x}_j) = \log\left\{1 + \frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^l}{\sigma_h^2}\right\}, \quad (3.14)$$

onde  $l \in \mathbb{Z}_+^*$  representa a ordem da norma da diferença dos vetores. Além destas, investiga-se também a função de Cauchy, representado como

$$k(\mathbf{x}_i, \mathbf{x}_j) = \left\{ 1 + \frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{\sigma_h^2} \right\}^{-1}. \quad (3.15)$$

Por fim, a função de kernel com decaimento moderado (*kernel with moderate decreasing*, KMOD) (Ayat *et al.*,) é dada por

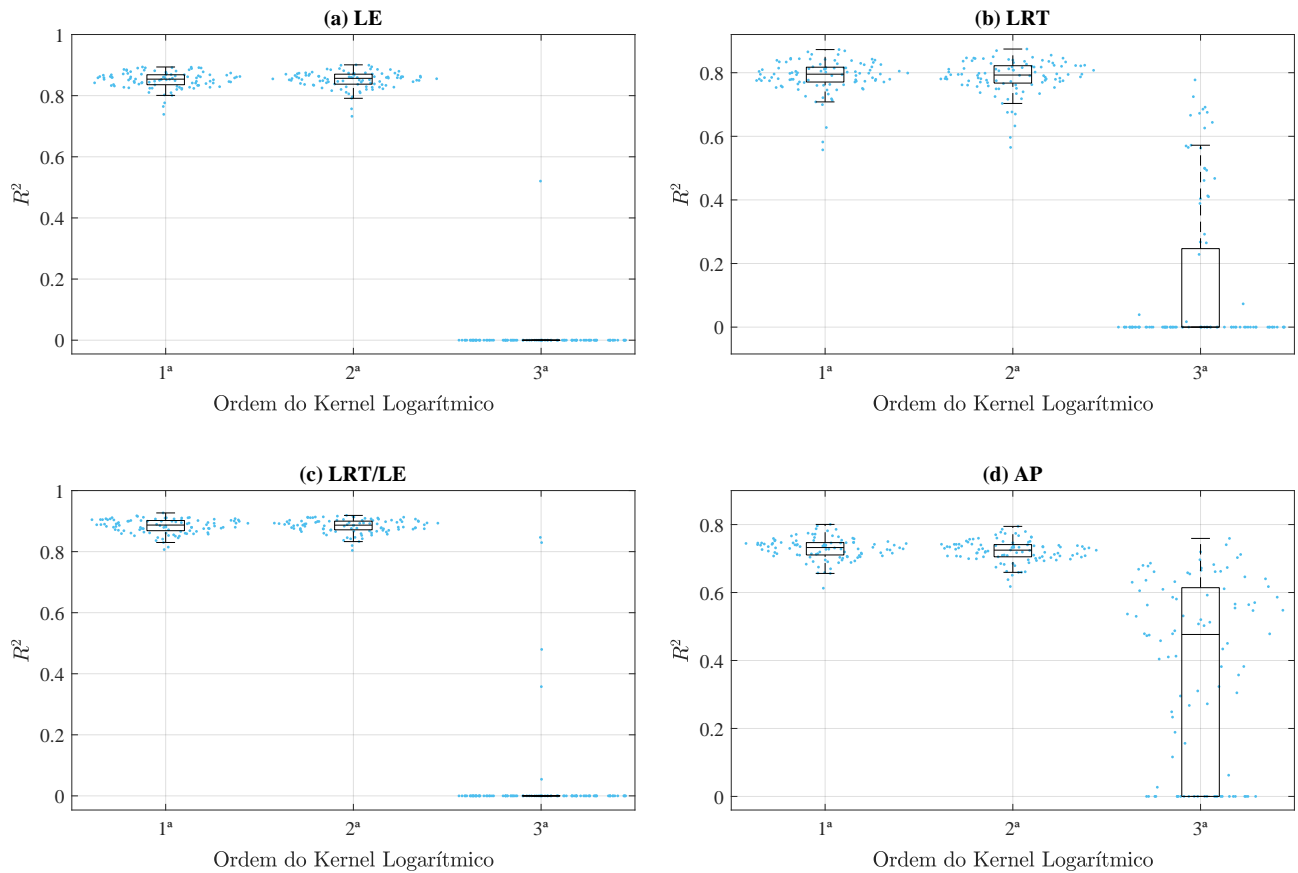
$$k(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{\exp(-\rho_h/\sigma_h^2) - 1} \left[ \exp \left\{ \frac{-\rho_h}{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 + \sigma_h^2} \right\} - 1 \right]. \quad (3.16)$$

Na etapa de sintonização dos hiperparâmetros, emprega-se a mesma metodologia discutida na seção anterior para as novas funções de kernel, ou seja, PSO com cinco partículas em um espaço de busca de  $\gamma^{-1}, \rho_h, \sigma_h \in [e^{-7.5}, e^{7.5}]$  e o método *Simplex* para refinar a solução. Em relação ao kernel logarítmico, são testadas as ordens 1, 2 e 3, cujos resultados são apresentados na Figura 7. Observa-se que não há diferença significativa entre as ordens 1 e 2; porém, a ordem 3 não conduziu a resultados consistentes. Por isso, opta-se por escolher a ordem 1 para a comparação com as demais funções de kernel.

Devido ao desempenho insatisfatório do kernel sigmoide, a metodologia do PSO foi alterada para utilizar 12 partículas em um espaço de busca expandido de  $\gamma^{-1}, \rho_h \in [e^{-15}, e^{7.5}]$ . Entretanto, essa alteração não resultou em melhoria no desempenho do modelo com esse kernel. A comparação de desempenho das funções de kernel é apresentada na Figura 8 por meio de *boxplots* e em valores numéricos (média e desvio padrão) de  $R^2$  na Tabela 5.

Diante desses resultados, pode-se inferir que os três melhores kernels testados são o exponencial, o logarítmico e o de Cauchy. Esses kernels apresentaram desempenho superior, tanto em termos de média quanto de desvio padrão de  $R^2$ , quando comparados ao kernel gaussiano, considerado o melhor avaliado até o momento. Outra análise interessante refere-se ao custo computacional. Praticamente, o tempo necessário para que o modelo calcule as previsões para as quatro saídas desejadas durante os testes do modelo é medido para cada kernel (100 medições  $\times$  4 saídas = 400 medições de tempo). A Figura 9 apresenta o *boxplot* desse tempo, incluindo o kernel linear como base de referência. Os resultados corroboram o menor custo computacional do kernel de Cauchy, devido ao cálculo mais simples, que é a função inversa de um valor escalar, em comparação às funções logarítmica e exponencial.

Figura 7 – *Boxplot* de desempenho para diferentes ordens da função logarítmica.



Fonte: Autoria própria.

Tabela 5 – Comparação de desempenho entre diferentes funções de kernel implementação própria do modelo LSSVR.

Tipos de Kernel	LE		LRT		LRT/LE		AP	
	$R^2_{mean}$	$R^2_{std}$	$R^2_{mean}$	$R^2_{std}$	$R^2_{mean}$	$R^2_{std}$	$R^2_{mean}$	$R^2_{std}$
Sigmoide	0,6907	0,1896	0,4849	0,1480	0,7887	0,1843	0,5617	0,1639
Gaussiano	0,8392	0,0276	0,7773	0,0506	0,8772	0,0230	0,7118	0,0346
Exponencial	0,8511	0,0271	0,7876	0,0506	0,8832	0,0232	0,7290	0,0315
Logarítmico	0,8513	0,0273	0,7881	0,0510	0,8833	0,0232	0,7293	0,0316
Cauchy	0,8515	0,0272	0,7885	0,0508	0,8839	0,0228	0,7244	0,0325
KMOD	0,8476	0,0368	0,7833	0,0613	0,8830	0,0232	0,7232	0,0332

Fonte: Autoria própria.

Em uma análise sobre a eficiência dos modelos, a aplicação do modelo com o kernel de Cauchy reduz o tempo médio de processamento em aproximadamente 17,8% (de 30,9 *ms* para 25,4 *ms*) em comparação ao logarítmico, moderadamente superior em termos de  $R^2$ . Ademais, a redução em comparação com o kernel gaussiano é de aproximadamente 24,4% (de 33,6 *ms* para 25,4 *ms*). Por esses motivos, a escolha do kernel de Cauchy é justificada como a mais adequada das testadas até o momento para a tarefa de previsão de propriedades mecânicas de vergalhões de aço.

Figura 8 – *Boxplot* para comparação de desempenho entre diferentes funções de kernel implementação própria do modelo LSSVR.

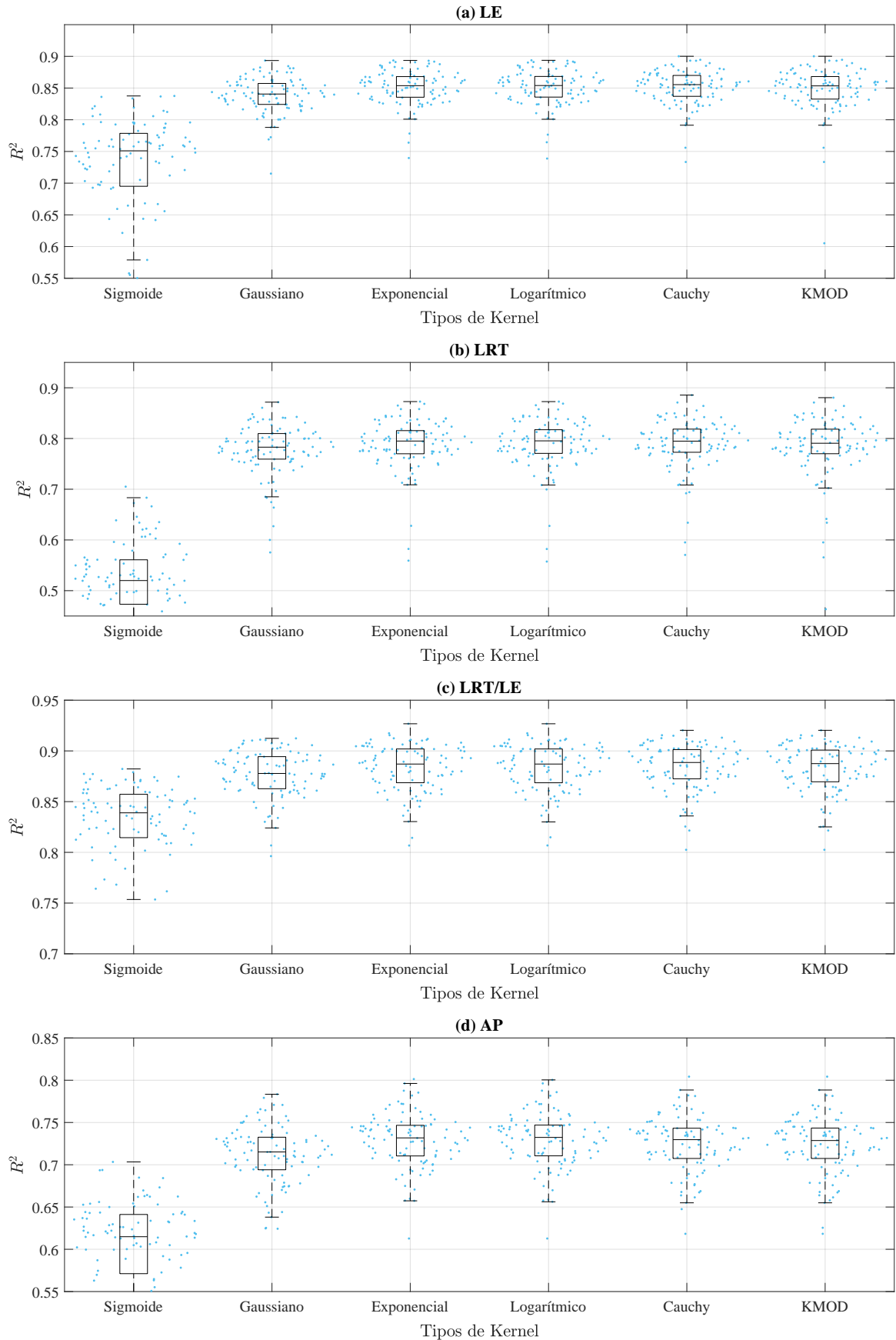
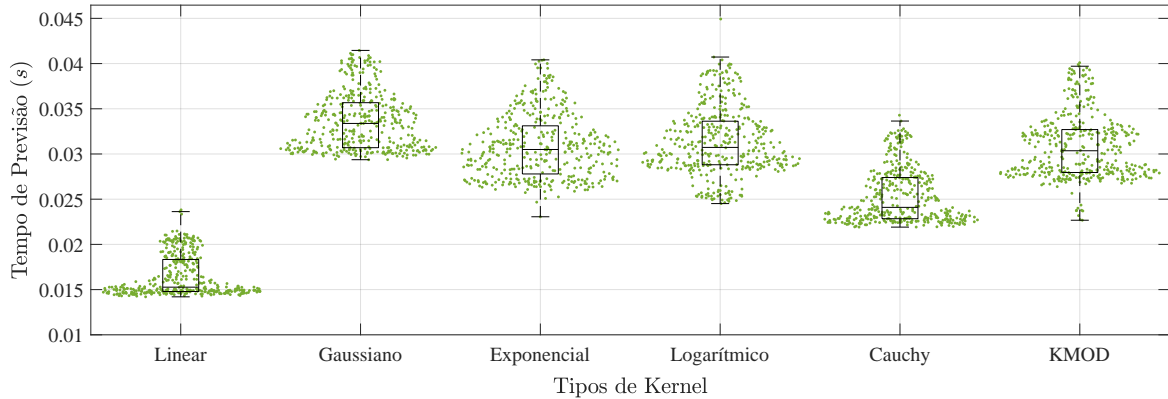


Figura 9 – *Boxplot* dos tempos de execução dos diferentes kernels durante a previsão do modelo. Os pontos verdes no gráfico de enxame mostram as observações individuais do tempo necessário para a previsão das amostras do conjunto de teste para cada saída, ajustados horizontalmente para evitar sobreposição e melhorar a visualização da densidade dos dados.



Fonte: Autoria própria.

### 3.5 Conclusão do Capítulo

Neste capítulo, foram apresentados os fundamentos do modelo LSSVR, juntamente com a metodologia aplicada na sintonia dos seus hiperparâmetros na implementação desenvolvida. Foram avaliadas diversas funções de kernel encontradas na literatura em termos de acurácia e tempo de processamento. Além disso, foi realizado um estudo preliminar com a implementação do modelo LSSVR via *toolbox* e a versão customizada. Ressalta-se que a versão customizada permitiu a investigação de diferentes kernels, que apresentaram resultados melhores do que o kernel gaussiano, este usado por padrão em várias aplicações de LSSVR reportadas na literatura. Com base nos resultados alcançados, observou-se que a função de Cauchy se destacou como a mais eficiente para a tarefa em questão. No próximo capítulo, são apresentadas ferramentas destinadas à esparsificação do modelo, visando reduzir o número de VSs para uma predição mais ágil e com menor uso de memória.

## 4 MÉTODOS DE REDUÇÃO DE VETORES SUPORTE

Para reduzir o número de VSs do modelo LSSVR, empregam-se nesta tese seis procedimentos de esparsificação. O primeiro método envolve uma simples seleção aleatória de VSs dos dados de treinamento, servindo como linha de base para avaliar o desempenho das outras abordagens. Em Murta *et al.* (2022), utiliza-se a poda por magnitude de  $\alpha$  (multiplicadores de Lagrange calculados pela Equação 3.6), denominado nesta tese como  $|\alpha|$ -*pruning*, baseada na metodologia descrita por Suykens *et al.* (2002). Esta técnica é revisitada aqui com a introdução de algumas mudanças e é a segunda abordagem testada.

O terceiro método utiliza o algoritmo *k-means* (MacQueen, 1967), uma técnica de QV comumente empregada para redução do volume de dados (Rocha Neto e Barreto, 2012). Neste contexto, o objetivo é representar um conjunto de dados original por um conjunto reduzido formado por protótipos, sem perda significativa de informação. Portanto, a seleção de VSs baseia-se na dissimilaridade, onde protótipos substituem amostras semelhantes de  $\mathcal{S}_{tr}$  na construção do modelo esparso. No entanto, o algoritmo *k-means* pode ser sensível a amostras atípicas (*outliers*). Para resolver essa questão, o algoritmo *k-medoids* (Kaufman e Rousseeuw, 1990) é aplicado como quarto método, uma abordagem robusta quando a média dos pontos não representa adequadamente o conjunto de dados.

A quinta abordagem é uma modificação proposta do método ALD, denominada ALD para poda (*ALD for pruning*, ALD4P), que visa encontrar um subconjunto dos VSs mais independentes entre si, dado o mapeamento não-linear induzido por uma função kernel. Por fim, propõe-se adicionar informação da saída do conjunto de dados nesse método, denominado ALD4P com entradas e saídas (*ALD4P with input and output*, ALD4P-IO), resultando no sexto método de esparsificação deste estudo.

### 4.1 Poda por Magnitude de $\alpha$

Para a realização da poda com base nos valores absolutos das magnitudes dos coeficientes de Lagrange, nomeada de  $|\alpha|$ -*pruning*, estipula-se a quantidade final de amostras no conjunto esparso, representada por  $N_r$ . O procedimento segue os seguintes passos descritos no Algoritmo 1.

Este procedimento assegura que o conjunto esparso final seja composto pelos VSs mais relevantes, removendo gradativamente os elementos com menor impacto no modelo,

---

**Algoritmo 1:** Pseudocódigo do  $|\alpha|$ -*pruning*.

---

**entrada:**  $\mathcal{S}_{tr}, N_r$   
**saída:**  $\mathcal{S}_s$   
**início**  
    # Verifique a quantidade de amostras:  
     $N_{tr} \leftarrow \mathcal{S}_{tr};$   
    # Atribua o conjunto esparsos:  
     $\mathcal{S}_s := \mathcal{S}_{tr};$   
    **para**  $i = 1 : (N_{tr} - N_r)$  **faça**  
        # Treine o modelo LSSVR para estimação de  $\alpha$ :  
         $\alpha \leftarrow LSSVR(\mathcal{S}_s);$   
        # Encontre o índice referente ao menor elemento de  $\alpha$  em valor absoluto:  
         $j_{min} = \arg \min(\text{abs}(\alpha));$   
        # Exclua a amostra  $(\mathbf{x}_{j_{min}}, \mathbf{y}_{j_{min}})$  do conjunto esparsos:  
         $\mathcal{S}_s := \mathcal{S}_s \setminus \{(\mathbf{x}_{j_{min}}, \mathbf{y}_{j_{min}})\};$   
    **fim**  
**fim**

---

conforme a abordagem de Suykens *et al.* (2002). Na abordagem canônica, a exclusão é feita em blocos de aproximadamente 5% dos menores valores absolutos de  $\alpha$  do conjunto de treinamento. No entanto, a retirada de um elemento por vez é mais eficaz, pois permite uma avaliação mais precisa do impacto individual de cada vetor no modelo.

Esta abordagem incremental minimiza o risco de remover elementos que, embora possuam valores absolutos de  $\alpha$  baixos, podem ser essenciais para a precisão do modelo. Além disso, em uma nova estimação dos coeficientes de Lagrange, aqueles valores inicialmente baixos podem não permanecer entre os menores após a remoção do elemento de menor valor, o que destaca a necessidade de uma abordagem mais criteriosa. Assim, a remoção gradual e individual dos elementos garante uma seleção mais precisa e potencialmente melhora a qualidade do conjunto esparsos final.

## 4.2 *K-Means* para Redução do Volume de Dados

No contexto da QV, o algoritmo *k-means* é aplicado para particionar o conjunto de dados em um número fixo de partições, onde cada grupo é representado por um centróide. Nessa abordagem que visa reduzir o conjunto de dados original, concatena-se as entradas e saídas correspondentes, isto é,  $\{\mathbf{x}_n^* := [\mathbf{x}_n; \mathbf{y}_n]\}_{n=1}^{N_{tr}}$ , para representar eficientemente as diferentes regiões do espaço de dados. Utiliza-se a distância euclidiana quadrática como função objetivo, definindo

---

**Algoritmo 2:** Pseudocódigo do *k-means* para redução do volume de dados.

---

**entrada:**  $\mathcal{S}_{tr}, N_r$

**saída:**  $\mathcal{S}_s$

**início**

# Verifique a quantidade de amostras:

$N_{tr} \leftarrow \mathcal{S}_{tr};$

# Concatene os pares de entrada-saída do conjunto  $\mathcal{S}_{tr}$ :

$\{\mathbf{x}_n^* := [\mathbf{x}_n; \mathbf{y}_n]\}_{n=1}^{N_{tr}};$

# Atribua aleatoriamente os valores dos centróides a partir de  $\{\mathbf{x}_n^*\}_{n=1}^{N_{tr}}:$

$\{\mathbf{c}_i\}_{i=1}^{N_r} \leftarrow \{\mathbf{x}_n^*\}_{n=1}^{N_{tr}};$

**repita**

# Atribua cada vetor  $\mathbf{x}_n^*$  ao centróide  $\mathbf{c}_i$  mais próximo:

Encontre o centróide  $\mathbf{c}_i$  mais próximo de  $\mathbf{x}_n^*$ ;

Atribua  $\mathbf{x}_n^*$  à partição  $\mathcal{P}_i$  correspondente;

# Atualize cada centróide  $\mathbf{c}_i$  calculando sua nova posição:

$\mathbf{c}_i :=$  média aritmética dos vetores em  $\mathcal{P}_i$ ;

**até** que não haja mudanças nas posições dos centróides;

# Separe os elementos de cada centróide  $\mathbf{c}_i$  em pares de entrada-saída:

$(\mathbf{x}'_i, \mathbf{y}'_i) \leftarrow \mathbf{c}_i$

# Defina o conjunto esparso:

$\mathcal{S}_s := \{(\mathbf{x}'_i, \mathbf{y}'_i)\}_{i=1}^{N_r}$

**fim**

---

o problema de minimização como

$$\min \sum_{i=1}^{N_r} \sum_{\forall \mathbf{x}_n^* \in \mathcal{P}_i} \|\mathbf{x}_n^* - \mathbf{c}_i\|_2^2, \quad (4.1)$$

em que  $N_r$  representa a quantidade de partições, ou seja, o número de protótipos que formam o conjunto de dados esparso por QV, e  $\mathbf{c}_i \in \mathbb{R}^{N_x + N_y}$  é o vetor centróide (protótipo) da partição  $\mathcal{P}_i$ . Esse algoritmo opera de modo iterativo descrito no Algoritmo 2. Após esse processo, cada centróide é dividido em pares de entrada-saída:  $\mathbf{c}_i \rightarrow (\mathbf{x}'_i, \mathbf{y}'_i)$ . Isso resulta em um conjunto de treinamento esparso, denotado como  $\mathcal{S}_s := \{(\mathbf{x}'_i, \mathbf{y}'_i)\}_{i=1}^{N_r}$ .

### 4.3 *K-Medoids* para Redução do Volume de Dados

O algoritmo *k-medoids* substitui o centróide  $\mathbf{c}_i$  em (4.1) pelo medóide  $\mathbf{m}_i$ , de forma que  $\mathbf{m}_i$  é necessariamente um vetor do conjunto  $\{\mathbf{x}_n^*\}_{n=1}^{N_r}$ . Esse algoritmo segue as operações apresentadas no Algoritmo 3. Ao final dessa técnica, assim como no *k-means*, cada medóide é dividido em pares de entrada-saída:  $\mathbf{m}_i \rightarrow (\mathbf{x}'_i, \mathbf{y}'_i)$ , resultando em um conjunto de treinamento esparso, denotado como  $\mathcal{S}_s := \{(\mathbf{x}'_i, \mathbf{y}'_i)\}_{i=1}^{N_r}$ . No entanto,  $(\mathbf{x}'_i, \mathbf{y}'_i) \in \mathcal{S}_{tr}$  não é apenas uma representação dos dados originais, pois os medóides são pontos reais do conjunto de dados.

---

**Algoritmo 3:** Pseudocódigo do *k-medoids* para redução do volume de dados.
 

---

**entrada:**  $\mathcal{S}_{tr}, N_r$ 
**saída:**  $\mathcal{S}_s$ 
**início**

# Verifique a quantidade de amostras:

 $N_{tr} \leftarrow \mathcal{S}_{tr};$ 

 # Concatene os pares de entrada-saída do conjunto  $\mathcal{S}_{tr}$ :

 $\{\mathbf{x}_n^* := [\mathbf{x}_n; \mathbf{y}_n]\}_{n=1}^{N_{tr}};$ 

 # Atribua aleatoriamente os valores dos medoides a partir de  $\{\mathbf{x}_n^*\}_{n=1}^{N_{tr}}$ :

 $\{\mathbf{m}_i\}_{i=1}^{N_r} \leftarrow \{\mathbf{x}_n^*\}_{n=1}^{N_{tr}};$ 
**repita**

 # Atribua cada vetor  $\mathbf{x}_n^*$  ao medoide  $\mathbf{m}_i$  mais próximo:

 Encontre o medoide  $\mathbf{m}_i$  mais próximo de  $\mathbf{x}_n^*$ ;

 Atribua  $\mathbf{x}_n^*$  à partição  $\mathcal{P}_i$  correspondente;

 # Atualize cada medoide  $\mathbf{m}_i$  encontrando sua nova posição:

 Calcule a soma das distâncias de cada  $\mathbf{x}_j^* \in \mathcal{P}_i$  para todos os outros vetores em  $\mathcal{P}_i$ ;

 Atualize o medoide  $\mathbf{m}_i$  para o vetor  $\mathbf{x}_j^*$  com a menor soma das distâncias;

**até** que não haja mudanças nas posições dos medoides;

 # Separe os elementos de cada medoide  $\mathbf{m}_i$  em pares de entrada-saída:

 $(\mathbf{x}'_i, \mathbf{y}'_i) \leftarrow \mathbf{m}_i$ 

# Defina o conjunto esparso:

 $\mathcal{S}_s := \{(\mathbf{x}'_i, \mathbf{y}'_i)\}_{i=1}^{N_r}$ 
**fim**

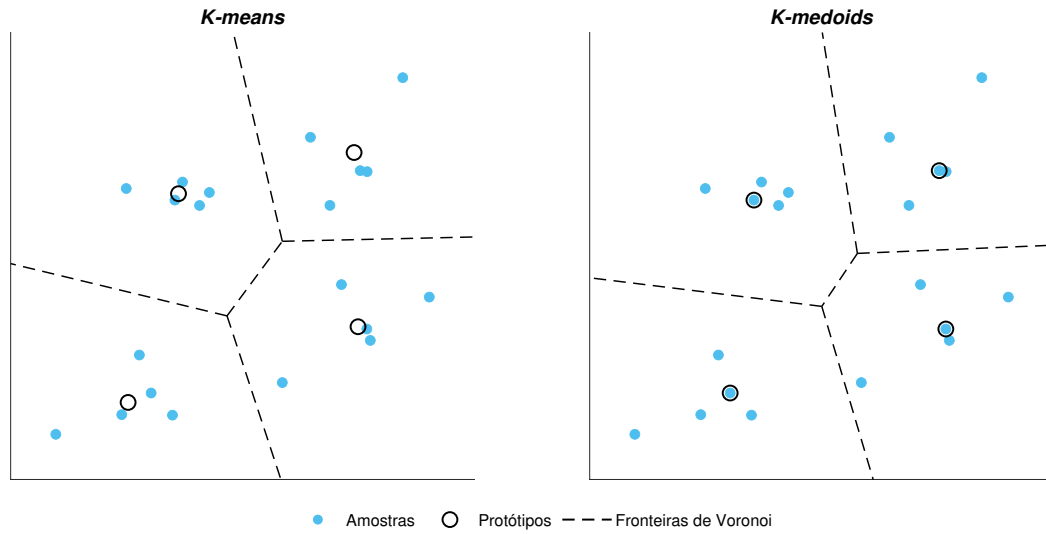

---

Uma representação esquemática do funcionamento do *k-means* e sua comparação com o método *k-medoids* é ilustrada na Figura 10. Nela, observa-se que, enquanto o *k-means* utiliza centróides que podem não coincidir com amostras reais, o *k-medoids* seleciona protótipos diretamente do conjunto de dados original, possibilitando representações mais robustas frente a valores discrepantes.

#### 4.4 ALD para Poda de VSs

Para remover sequencialmente amostras de um conjunto de dados, propõe-se uma variante do método ALD, uma técnica de esparsificação amplamente utilizada para selecionar os VSs relevantes enquanto modelos baseados em kernel são treinados online (Coelho e Barreto, 2019). A variante proposta, doravante referenciada pelo acrônimo ALD4P, realiza a poda dos VSs do conjunto completo de amostras de treinamento, em vez de inseri-los um a um, como é comum no processamento de dados temporais. Como se lida com dados estáticos (i.e., não temporais) nesta tese, a abordagem de poda se ajusta melhor aos requisitos da tarefa de regressão.

Figura 10 – Comparação entre os algoritmos *k-means* e *k-medoids*. Os círculos representam os protótipos (centróides ou medóides), e as linhas tracejadas indicam as fronteiras de Voronoi que delimitam as partições do espaço de dados.



Fonte: Autoria própria.

Originalmente, este método funciona da seguinte maneira. No passo de treinamento  $t$  ( $2 \leq t \leq N_{tr}$ ), após a observação de  $t - 1$  amostras de treinamento, o dicionário  $\mathcal{D}_{t-1}$  é composto por um subconjunto de  $m_{t-1}$  entradas de treinamento relevantes, denotadas por  $\{\tilde{\mathbf{x}}_j\}_{j=1}^{m_{t-1}}$ . Quando uma nova amostra de treinamento  $\mathbf{x}_t$  é disponibilizada, verifica-se se ela deve ser adicionada ao dicionário. Para isso, é necessário estimar o vetor de coeficientes  $\mathbf{a} = [a_1, \dots, a_{m_{t-1}}]^\top$ , que deve satisfazer o critério de ALD

$$\delta_t \stackrel{def}{=} \min_{\mathbf{a}} \left\| \sum_{j=1}^{m_{t-1}} a_j \phi(\tilde{\mathbf{x}}_j) - \phi(\mathbf{x}_t) \right\|^2 \leq \mu, \quad (4.2)$$

onde  $\delta_t \in \mathbb{R}_+$  é o erro de aproximação e  $\mu \in \mathbb{R}_+$  é o parâmetro de nível de esparsidade. Desenvolvendo o problema de minimização da Equação 4.2 e especificando uma função de kernel, pode-se escrever, em notação matricial:

$$\delta_t = \min_{\mathbf{a}} \left\{ \mathbf{a}^\top \tilde{\mathbf{\Omega}}_{t-1} \mathbf{a} - 2\mathbf{a}^\top \tilde{\mathbf{\kappa}}_{t-1} + \kappa_{tt} \right\}, \quad (4.3)$$

em que  $[\tilde{\mathbf{\Omega}}_{t-1}]_{i,j} = k(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j)$ ,  $[\tilde{\mathbf{\kappa}}_{t-1}]_i = k(\tilde{\mathbf{x}}_i, \mathbf{x}_t)$ ,  $\forall i, j \in \{1, \dots, m_{t-1}\}$ , e  $\kappa_{tt} = k(\mathbf{x}_t, \mathbf{x}_t)$ . Resolver a Equação 4.3 leva ao vetor ótimo  $\mathbf{a}_t$ , dado por

$$\mathbf{a}_t = \tilde{\mathbf{\Omega}}_{t-1}^{-1} \tilde{\mathbf{\kappa}}_{t-1}(\mathbf{x}_t), \quad (4.4)$$

permitindo que a condição ALD seja reescrita como

$$\delta_t = \kappa_{tt} - \tilde{\mathbf{\kappa}}_{t-1}^\top \mathbf{a}_t \leq \mu. \quad (4.5)$$

---

**Algoritmo 4:** Pseudocódigo do ALD4P.
 

---

**entrada:**  $\mathcal{S}_{tr}, N_r$   
**saída:**  $\mathcal{S}_s$   
**início**  
 # Verifique a quantidade de amostras:  
 $N_{tr} \leftarrow \mathcal{S}_{tr};$   
 # Monte o dicionário a partir do conjunto de treinamento:  
 $\mathcal{D} \leftarrow \mathcal{S}_{tr};$   
 # Atribua o conjunto esparso:  
 $\mathcal{S}_s := \mathcal{S}_{tr};$   
 $m = N_{tr};$   
**para**  $i = 1 : (N_{tr} - N_r)$  **faça**  
   **para**  $t = 1 : m$  **faça**  
     | Calcule  $\delta_t$  para  $\tilde{\mathbf{x}}_t$  com o dicionário  $\mathcal{D} \setminus \{\tilde{\mathbf{x}}_t\};$   
   **fim**  
   # Encontre o índice referente ao menor  $\delta_t$ :  
    $t_{min} = \arg \min \{\delta_t\}_{t=1}^m;$   
   # Exclua o elemento  $\tilde{\mathbf{x}}_{t_{min}}$  do dicionário:  
    $\mathcal{D} := \mathcal{D} \setminus \{\tilde{\mathbf{x}}_{t_{min}}\};$   
   # Exclua a amostra  $(\mathbf{x}_{t_{min}}, \mathbf{y}_{t_{min}})$  do conjunto esparso:  
    $\mathcal{S}_s := \mathcal{S}_s \setminus \{(\mathbf{x}_{t_{min}}, \mathbf{y}_{t_{min}})\};$   
    $m = m - 1;$   
**fim**  
**fim**

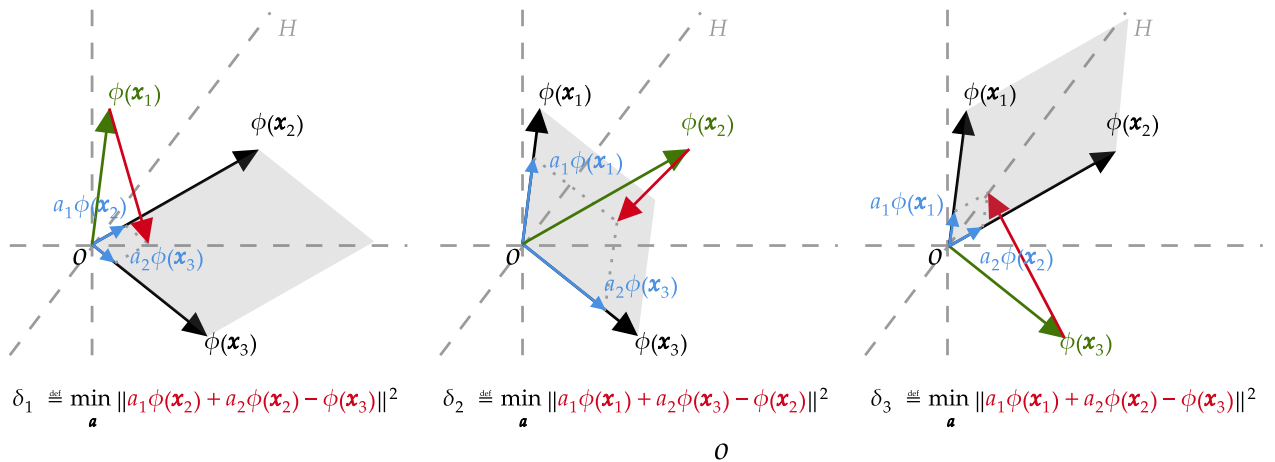
---

Se  $\delta_t > \mu$ , então a amostra  $\mathbf{x}_t$  deve ser adicionada ao dicionário, ou seja,  $\mathcal{D}_t = \mathcal{D}_{t-1} \cup \{\mathbf{x}_t\}$  e  $m_t = m_{t-1} + 1$ . No entanto, se  $\delta_t < \mu$ , a amostra é aproximadamente linearmente dependente e não deve ser adicionada ao dicionário.

Nesta tese, adapta-se o método ALD para remover sequencialmente a “amostra mais linearmente dependente” do conjunto de dados. Inicialmente, o dicionário é composto por  $N_{tr}$  amostras (todo o conjunto de dados de treinamento). Então, no passo de remoção 1,  $N_{tr}$  valores de  $\delta_n$  são gerados, considerando uma amostra específica removida  $\mathbf{x}_n$  como  $\mathbf{x}_t$  (ver equações de 4.2 a 4.5), e um dicionário composto pelas  $N_{tr} - 1$  amostras restantes. A amostra a ser removida, nesse passo, é aquela que gera o menor  $\delta_n$ , pois representa a amostra mais linearmente dependente das outras. No passo de remoção 2,  $N_{tr} - 1$  valores de  $\delta_n$  são gerados usando o conjunto de dados resultante do passo anterior. Este procedimento é repetido até que se atinja um número arbitrário  $N_r$  de amostras removidas sequencialmente. O Algoritmo 4 sintetiza a ideia do método ALD4P dada a escolha de uma quantidade  $N_r$  amostras para constituir o conjunto de dados de treinamento esparso.

Além disso, uma variante desse modelo é testada: a ALD4P-IO. Esta nova abordagem

Figura 11 – Representação do método ALD4P para situação inicial com três vetores de suporte no espaço de características  $H$ .



Fonte: Autoria própria.

concatena a informação da saída nos elementos do dicionário na forma  $\{\tilde{\mathbf{x}}_n^* := [\mathbf{x}_n; y_n]\}_{n=1}^{N_{tr}}$  (dicionário inicial), semelhante ao que ocorre na QV, e realiza o mesmo procedimento descrito do ALD4P.

### Exemplo ilustrativo

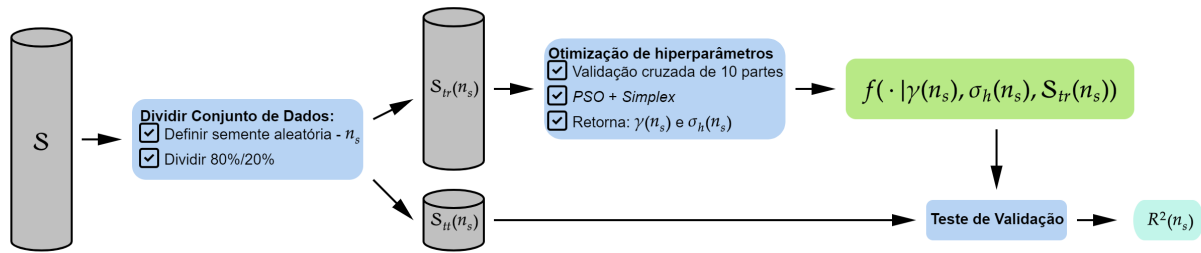
Para tornar mais clara a aplicação do método ALD4P, considere o caso em que o dicionário inicial possui três VSs,  $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$ . O objetivo é reduzir o dicionário para dois VSs, de modo a realizar a esparsificação mantendo a capacidade de representação.

**Passo 1.** Para cada vetor de suporte  $\mathbf{x}_t \in \mathcal{D}$ , calcula-se o erro de aproximação  $\delta_t$  correspondente à tentativa de representar  $\phi(\mathbf{x}_t)$  como combinação linear dos demais VSs. Assim, como ilustrado na Figura 11,  $\delta_1$  é obtido ao aproximar  $\phi(\mathbf{x}_1)$  por combinação de  $\phi(\mathbf{x}_2)$  e  $\phi(\mathbf{x}_3)$ ; de modo análogo,  $\delta_2$  e  $\delta_3$  são calculados ao tentar expressar  $\phi(\mathbf{x}_2)$  e  $\phi(\mathbf{x}_3)$ , respectivamente, como combinações lineares dos outros vetores de suporte. O valor de  $\delta_t$  quantifica o grau de dependência linear do vetor  $\mathbf{x}_j$  em relação aos demais: quanto menor  $\delta_j$ , maior a redundância desse vetor no dicionário.

**Passo 2.** O algoritmo identifica o índice  $t_{min} = \arg \min_{t \in \{1,2,3\}} \delta_t$ , isto é, o VS cuja remoção gera o menor erro de aproximação.

**Passo 3.** O novo dicionário reduzido é então dado por  $\mathcal{D}' := \mathcal{D} \setminus \{\mathbf{x}_{t_{min}}\}$ , contendo apenas dois VSs.

Figura 12 – Diagrama de resumo da otimização dos hiperparâmetros do modelo LSSVR.



Fonte: Autoria própria.

#### 4.5 Sobre a Esparsificação do Modelo LSSVR

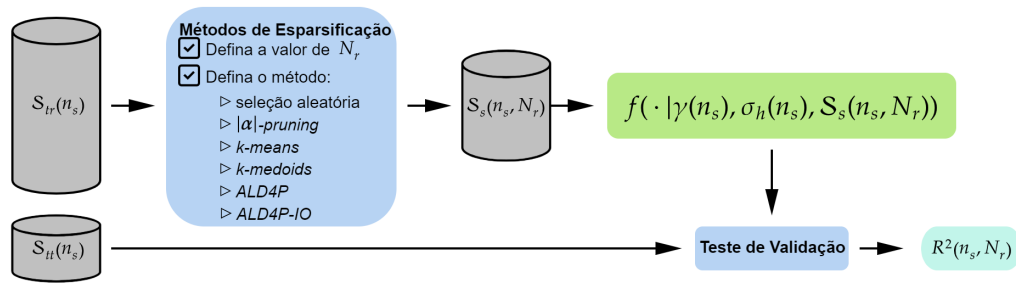
Para os métodos de redução de VSs, o conjunto de dados é embaralhado 100 vezes, cada vez com uma semente aleatória específica ( $n_s$ ). Em seguida, ele é dividido em dois grupos: os primeiros 80% dos elementos para treinamento e os restantes 20% para teste. Dessa forma, utilizam-se as mesmas 100 divisões de dados em todos os experimentos, garantindo comparações justas e consistentes entre os diferentes métodos.

Com base nos resultados apresentados no Capítulo 3, opta-se por utilizar o kernel de Cauchy neste estudo de esparsificação do modelo LSSVR. Assim, os hiperparâmetros desse modelo, a saber, a largura  $\sigma_h(n_s)$  da função de Cauchy e a constante de regularização  $\gamma(n_s)$ , são ajustados usando a abordagem conjunta dos algoritmos PSO e *Simplex*, aplicada ao conjunto de dados não esparsos. Os hiperparâmetros ótimos são aplicados a todos os outros métodos avaliados. O esquema dessa otimização é ilustrado na Figura 12. É importante destacar que as quatro saídas de propriedades mecânicas de vergalhões de aço são analisadas individualmente, o que implica na otimização separada dos hiperparâmetros do modelo e do conjunto esparsos de VSs.

Para a análise dos métodos de esparsificação, um número arbitrário  $N_r$  é selecionado para determinar o tamanho do conjunto esparsos resultante, e seu respectivo coeficiente  $R^2$  é calculado. Considera-se a retirada com a discretização de 10 VSs até que se atinja 500 VSs retirados. A linha de base adotada pelo método de seleção aleatória consiste em remover elementos sequencialmente do conjunto de treinamento embaralhado, dada a semente aleatória  $n_s$ . O método  $|\alpha|$ -*pruning* utiliza a estimação do modelo LSSVR para sua operação. Ou seja, os VSs são excluídos ao passo que o modelo com a função de Cauchy e hiperparâmetros  $\gamma_h(n_s)$   $\sigma_h(n_s)$  são mantidos para cada saída.

Quanto aos métodos de QV, são utilizadas as funções do MATLAB® @kmeans e

Figura 13 – Diagrama de resumo de metodologia de esparsificação.



Fonte: Autoria própria.

@kmedoids, configurando o número de protótipos (centróides ou medoides) resultantes igual a  $N_r$  e considerando o conjunto de dados de treinamento completo. O método ALD4P é implementado completamente do zero, seguindo as informações detalhadas na Seção 4.4. Este método utiliza o mesmo kernel do modelo LSSVR, com os seguintes hiperparâmetros: a constante de regularização  $\gamma^{-1} = 0,0001$  e a largura  $\sigma_h(n_s)$  da função de Cauchy. Os VSs são removidos sequencialmente até alcançar a quantidade desejada. O diagrama esquemático da metodologia dos experimentos é apresentado na Figura 13.

Durante a análise dos métodos de esparsificação para o modelo LSSVR, faz-se uma escolha deliberada para reduzir o número de VSs enquanto se mantêm a acurácia equivalente a do conjunto completo de VSs. Para auxiliar neste processo, uma linha contínua e duas pontilhadas em preto são traçadas para representar os valores alcançados pelo modelo LSSVR não esparsificado em termos de média e desvio padrão de  $R^2$ , respectivamente, nas figuras de comparação de desempenho desses métodos para as quatro saídas previstas (Figuras 14–17). Adicionalmente, as barras de erro na cor laranja são geradas para indicar o desempenho da remoção aleatória de VSs do modelo nessas representações, ou seja, uma remoção de VSs sem critério para contrastar com as demais abordagens avaliadas.

Em uma análise geral das quatro propriedades estudadas, observa-se que a seleção aleatória degrada rapidamente a acurácia do modelo à medida que o número de VSs é reduzido, ressaltando a importância de desenvolver técnicas eficazes de redução de quantidade de VSs. O método  $|\alpha|$ -pruning, na cor roxa, obtém um desempenho inferior às demais técnicas (com exceção da seleção aleatória), com destaque negativo para os resultados das saídas LE e LRT.

Os métodos *k-means*, *k-medoids*, ALD4P e ALD4P-IO apresentam desempenhos semelhantes para as quatro propriedades. Assim, a escolha do método vencedor se baseia na

Figura 14 – Barra de erro dos resultados dos métodos de esparsificação para saída LE.

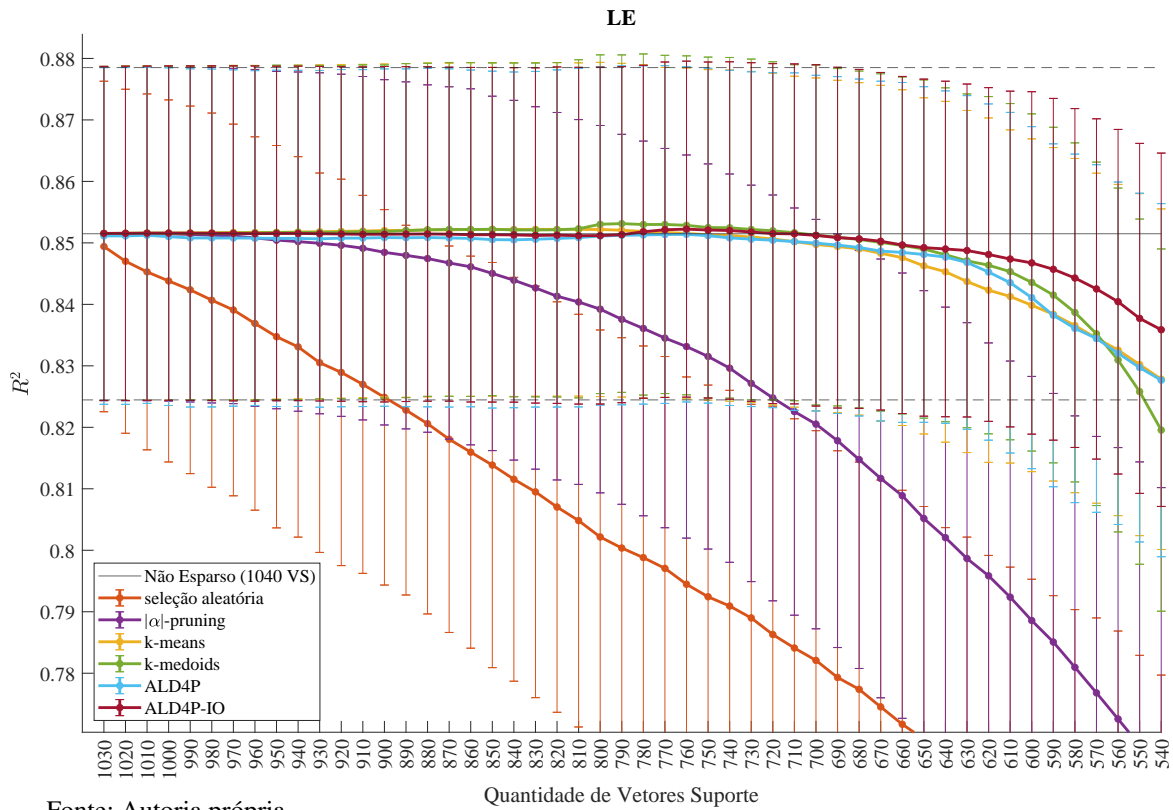


Figura 15 – Barra de erro dos resultados dos métodos de esparsificação para saída LRT.

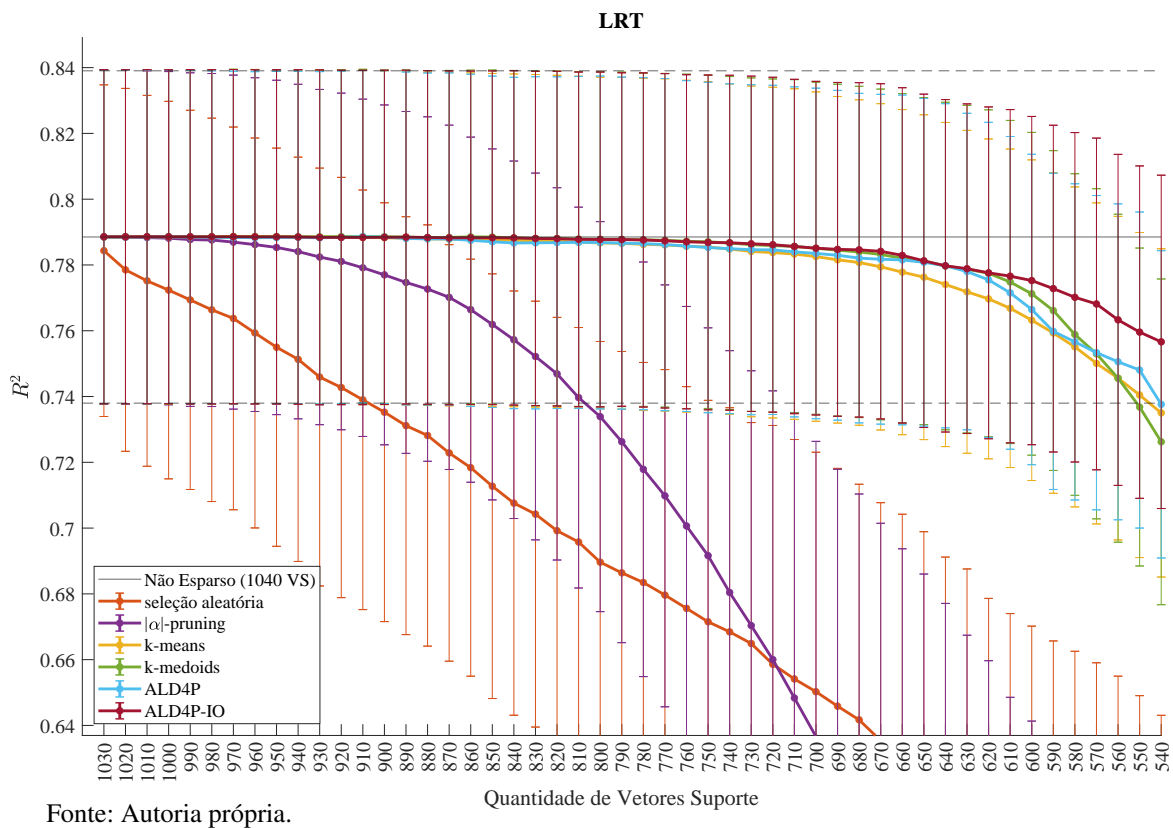


Figura 16 – Barra de erro dos resultados dos métodos de esparsificação para saída LRT/LE.

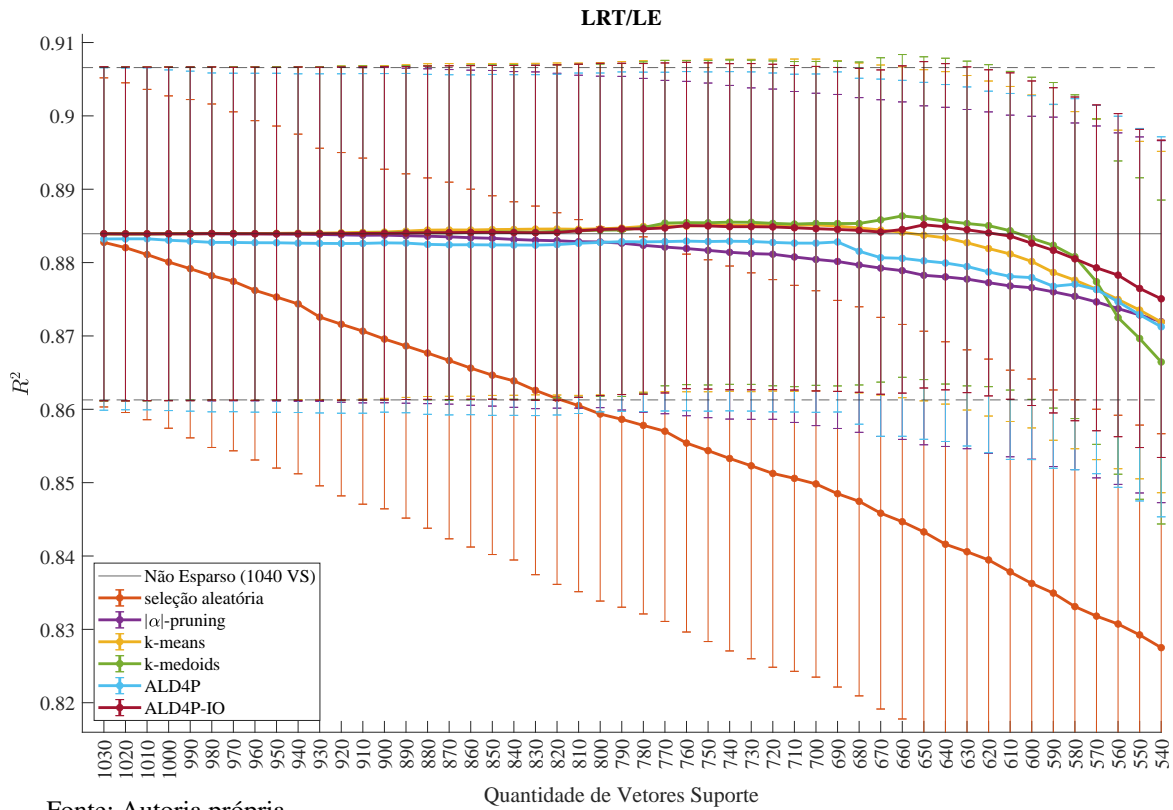


Figura 17 – Barra de erro dos resultados dos métodos de esparsificação para saída AP.

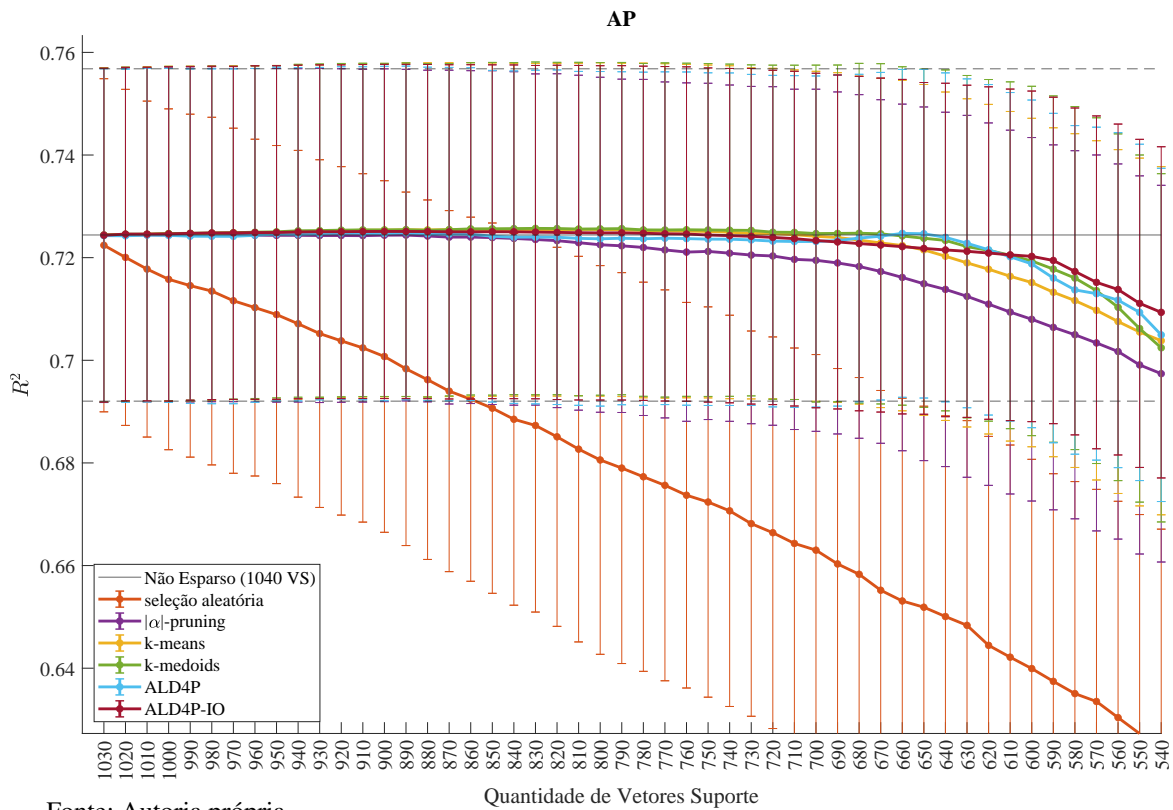


Tabela 6 – Comparação de desempenho entre diferentes com foco na esparsificação do modelos.

Modelos	LE			LRT			LRT/LE			AP		
	$R^2_{mean}$	$R^2_{std}$	SV%	$R^2_{mean}$	$R^2_{std}$	SV%	$R^2_{mean}$	$R^2_{std}$	SV%	$R^2_{mean}$	$R^2_{std}$	SV%
LSSVR <sup>1</sup>	0,7989	0,0342	86,4	0,7389	0,0630	87,1	0,8603	0,0240	85,5	0,6883	0,0315	86,1
LSSVR <sup>2</sup>	0,8515	0,0272	100	0,7885	0,0508	100	0,8839	0,0228	100	0,7244	0,0325	100
S-LSSVR <sup>3</sup>	0,8512	0,0276	67,3	0,7869	0,0506	72,1	0,8841	0,0221	59,6	0,7237	0,0324	67,3

<sup>1</sup> Murta *et al.* (2022); <sup>2</sup> Kernel de Cauchy; <sup>3</sup> Kernel de Cauchy e ALD-IO.

Fonte: Autoria própria.

avaliação de outros critérios, como memória para armazenamento do modelo, complexidade computacional e flexibilidade de aplicação. Em relação à memória, o algoritmo *k-means* apresenta a desvantagem de usar protótipos em vez de dados reais. Consequentemente, os quatro modelos criados não podem compartilhar seus VSs.

Considerando os métodos de esparsificação que selecionam os VSs, o algoritmo *k-medoids* se destaca pelo seu desempenho moderadamente melhor que os outros em termos de  $R^2$ . No entanto, não conta com um arcabouço teórico robusto como os métodos propostos baseados em ALD, que possuem métricas claras de seleção de amostras e não sofrem de aleatoriedade na inicialização. Além disso, o método ALD oferece uma metodologia de inserção e, a partir desta pesquisa com o ALD4P, de poda de VSs, proporcionando mais flexibilidade e a possibilidade de manutenção do modelo em cenários de aprendizado adaptativo.

Em Bessa *et al.* (2024), introduz-se o método ALD4P, que utiliza apenas a informação de entrada das amostras. Considerando a nova abordagem, o método ALD4P-IO alcança uma leve vantagem numérica para as saídas LE, LRT e LRT/LE. Como pode ser percebido nos gráficos, também apresenta um decaimento mais suave e estável. Por esses motivos apresentados, opta-se pelo ALD4P-IO como método de esparsificação mais adequado.

Ao realizar uma avaliação individual do método ALD4P-IO referentes a cada propriedade mecânica, observa-se que a barra de erro do método ALD4P-IO, destacada na cor vermelha na Figura 14, para a saída LE indica que, a partir de uma inspeção visual, aproximadamente 67,31% (700) dos VSs podem ser mantidos sem perda significativa de desempenho. Na Figura 15, a indicação para a saída LRT é de aproximadamente 72,12% (750) dos VSs. Em seguida, na Figura 16, a indicação aproximada para a saída LRT/LE é de 59,62% (620) dos VSs. Finalmente, a indicação para a saída PE é próxima de 67,31% (700) dos VSs, conforme demonstrado na Figura 17. As esparsificações propostas com seus respectivos valores de  $R^2$  são exibidas na Tabela 6 para o modelo para o modelo doravante chamado de *sparse* LSSVR (S-LSSVR) em comparação com o modelo não esparsificado e o proposto em Murta *et al.* (2022).

## 4.6 Conclusão do Capítulo

Este capítulo apresentou avanços significativos na redução da quantidade de VSs, mantendo a acurácia do modelo LSSVR. No próximo capítulo, é introduzida uma modelagem local para regressão baseada no modelo LSSVR, na qual o conjunto de treinamento é particionado. Essa abordagem difere da esparsificação do modelo LSSVR global e visa tornar o modelo mais eficiente. Além disso, é analisada a possibilidade de utilizar essa técnica como um complemento à esparsificação.

## 5 MODELAGEM LOCAL PARA REGRESSÃO KERNEL

Este capítulo explora a modelagem local no contexto do modelo LSSVR. Em vez de usar um modelo global, os dados são divididos em regiões menores utilizando agrupamentos rígidos através do algoritmo *k-means*. Essa abordagem visa reduzir os custos computacionais, especialmente durante a predição do modelo. Adicionalmente, avalia-se a viabilidade de um modelo local e esparso. Os resultados alcançados são apresentados e discutidos em detalhe.

### 5.1 Modelagem Local Baseada em Agrupamentos

O paradigma de aprendizado local permite a adaptação do modelo ao capturar relações complexas entre as variáveis de entrada e saída em diferentes regiões do seu espaço de características. Em vez de assumir uma única representação global, modelos locais são tipicamente mais simples e podem interagir de forma cooperativa ou competitiva, proporcionando maior interpretabilidade e flexibilidade operacional. Com frequência, a abordagem local permite manter ou até mesmo melhorar a precisão da previsão em comparação com o modelo global (Jacobs *et al.*, 1991).

No contexto do modelo LSSVR, a modelagem local baseada em agrupamentos (Alpaydin e Jordan, 1996) apresenta uma proposta interessante para reduzir os custos computacionais de predição. Para alcançar tal meta, o modelo local requer a partição do conjunto de dados de treinamento  $\mathcal{S}_{tr}$  (ou seu respectivo conjunto esparso  $\mathcal{S}_s$ ) com base em algum critério de localidade. Neste estudo, o algoritmo *k-means* é utilizado para agrupar o conjunto de dados de entrada,  $\mathcal{X}_{tr}$ , em  $K$  partições, guiado pela distância euclidiana quadrática. Para cada uma dessas partições  $\mathcal{P}_i \subset \mathcal{X}_{tr}$  um modelo LSSVR é construído com o seguinte subconjunto dos dados  $\mathcal{C}_i := \{\forall(\mathbf{x}_n, \mathbf{y}_n) \in \mathcal{S}_{tr} | \mathbf{x}_n \in \mathcal{P}_i\}$ .

Após o passo de treinamento, as saídas dos  $K$  modelos locais podem ser combinadas como

$$f(\mathbf{x}) = \sum_{i=1}^K h_i(\mathbf{x}) f_i(\mathbf{x}), \quad (5.1)$$

em que  $h_i(\cdot) : \mathbb{R}^{N_x} \rightarrow \mathbb{R}$  é uma função de ponderação para as saídas dos modelos LSSVR locais,  $f_i(\cdot)$ . Em uma abordagem do tipo “vencedor leva tudo” (*winner-takes-all*), apenas um modelo local é selecionado para realizar a predição da saída. Assim, o algoritmo calcula as distâncias do vetor de entrada  $\mathbf{x}$  da nova amostra para cada centróide correspondente às partições

encontradas pelo algoritmo *k-means*. Dessa forma, o modelo vencedor é aquele cujo respectivo centróide é o mais próximo do vetor de entrada atual, definindo a função de ponderação como

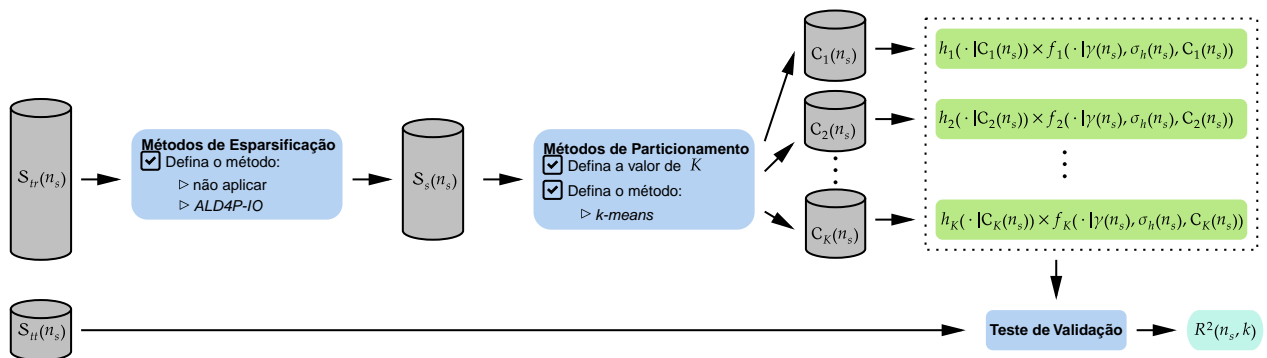
$$h_i(\mathbf{x}) = \begin{cases} 1, & \text{if } \|\mathbf{x} - \mathbf{c}_i\|_2^2 = \min_{1 \leq j \leq K} \|\mathbf{x} - \mathbf{c}_j\|_2^2, \\ 0, & \text{caso contrário.} \end{cases} \quad (5.2)$$

Em termos simples, a função de ponderação  $h_i(\cdot)$  na Equação 5.1 é determinada após uma busca pelo vetor protótipo mais próximo do vetor de entrada atual  $\mathbf{x}$ . Para o vetor protótipo mais próximo, define-se  $h_i(\mathbf{x}) = 1$ . Caso contrário,  $h_i(\mathbf{x}) = 0$ , como mostrado na Equação 5.2. Isso é válido apenas para técnicas de agrupamento rígido, implicando que apenas um subconjunto dos VSs disponíveis são utilizados em cada predição por meio de um modelo local e não se compartilha elementos entre agrupamentos. Para técnicas de agrupamento suave (Ferraro e Giordani, 2019), como algoritmos de *clustering fuzzy* (não utilizados na presente tese), o fator de ponderação  $h_i(\cdot)$  pode assumir valores entre 0 e 1, significando que os modelos locais podem usar amostras de dados de agrupamentos vizinhos ponderados por seus valores de pertinência. De maneira similar, também ocorre a ponderação de modelos com dados particionados em técnica de regressão por comitê (Gangquan *et al.*, 2015; Sun *et al.*, 2018). A principal vantagem do agrupamento rígido na modelagem local reside na significativa redução das operações com VSs, o que torna o modelo mais eficiente e, portanto, uma escolha atraente para aplicações que demandam alta eficiência computacional.

## 5.2 Sobre a Modelagem Local da para Regressão Usando LSSVR

Para a abordagem local para o modelo LSSVR, a função `@kmeans` do MATLAB<sup>®</sup> é utilizada para particionar os dados em  $K$  subconjuntos, com cada partição sendo utilizada para treinar um modelo local. O desempenho do LSSVR local é avaliado com o aumento da quantidade  $K$  de modelos locais. Os experimentos são realizados tanto no conjunto de dados de treinamento original quanto no conjunto esparsificado, especificado no Capítulo 4. Por simplicidade, os modelos locais herdam os hiperparâmetros do respectivo modelo global. A Figura 18 fornece uma descrição geral dos experimentos conduzidos neste capítulo. A metodologia aplicada para avaliação é semelhante à tratada na Seção 4.5.

Figura 18 – Diagrama de resumo de modelagem local.



Fonte: Autoria própria.

Os resultados do modelo LSSVR local são apresentados na Figura 19, a partir da qual avalia-se a partição dos VSs entre 2 a 10 agrupamentos do conjunto completo de VSs e para o conjunto esparso resultante do método ALD4P-IO. São traçadas barras de erro de três grupos de informações em termos de média e desvio padrão para 100 rodadas, sendo a primeira dos valores de  $R^2$ . A segunda considera a quantidade de operações médias com VSs dada a diferença do tamanho das partições dos modelos locais. E, a terceira, o tempo gasto na predição do conjunto de teste em relação ao aumento do número de modelos locais para cada saída analisada. Nota-se que as barras de erro de  $R^2$  entre usar o conjunto de dados completo e o esparso estão praticamente sobrepostas, permitindo a escolha da partição dos VSs selecionados do modelo S-LSSVR, resultando no modelo denominado doravante de modelo LSSVR local e esparso (*local sparse* LSSVR, LS-LSSVR) com  $K$  partições,  $K$ -LS-LSSVR. Por exemplo, a sigla 5-LS-LSSVR denota um modelo LS-LSSVR com 5 partições.

Os gráficos para as quatro propriedades mecânicas revelam que o aumento no número de modelos locais tem efeitos mínimos e não prejudica significativamente os valores de  $R^2$  para as quatro saídas analisadas. Entretanto, não se observa redução significativa no custo computacional pelos gráficos de quantidade de operações com VSs e tempo de execução a partir de 5 partições, ou seja, do modelo 5-LS-LSSVR. Isso provavelmente se deve à formação de agrupamentos pouco significativos, isto é, que reduzem os elementos dos demais agrupamentos e cujos modelos não são utilizados durante as predições do conjunto de teste.

Vale ressaltar que o modelo 2-LS-LSSVR não demonstra degradação no desempenho em nenhuma das quatro saídas previstas. Ademais, exibe uma redução média significativa no número de VSs utilizados durante a operação. Especificamente, utiliza aproximadamente 36,25% (377) dos SVs para estimativa de LE, 38,75% (403) para LRT, 31,83% (331) para LRT/LE e

Tabela 7 – Comparação de desempenho entre diferentes com foco na modelagem local.

Modelos	LE			LRT			LRT/LE			AP		
	$R^2_{mean}$	$R^2_{std}$	SV%	$R^2_{mean}$	$R^2_{std}$	SV%	$R^2_{mean}$	$R^2_{std}$	SV%	$R^2_{mean}$	$R^2_{std}$	SV%
LSSVR	0,8515	0,0272	100	0,7885	0,0508	100	0,8839	0,0228	100	0,7244	0,0325	100
S-LSSVR	0,8512	0,0276	67,3	0,7869	0,0506	72,1	0,8841	0,0221	59,6	0,7237	0,0324	67,3
2-LS-LSSVR	0,8508	0,0282	36,3	0,7867	0,0495	38,8	0,8837	0,0222	31,8	0,7224	0,0336	36,7
5-LS-LSSVR	0,8456	0,0297	16,6	0,7822	0,0507	17,7	0,8801	0,0224	14,7	0,7178	0,0354	16,3

Fonte: Autoria própria.

36,73% (382) para AP durante a operação. No caso do modelo 5-LS-LSSVR, utiliza apenas 16,63% (173) dos SVs para estimativa de LE, 17,69% (184) para LRT, 14,71% (153) para LRT/LE e 16,35% (170) para AP durante a operação. No entanto, já é perceptível uma ligeira redução no acurácia do modelo. A Tabela 7 reúne essas informações juntamente com os valores de  $R^2$  das abordagens global, esparsa e local-esparsa. Adicionalmente, apresenta-se o *boxplot* dos resultados desses experimentos na Figura 20.

Em uma análise final desta tese sobre a eficiência dos modelos, o *boxplot* ilustrado na Figura 21 apresenta a comparação entre o tempo de execução dos testes para quatro propriedades mecânicas previstas. Esse gráfico inclui o modelo LSSVR com kernel linear, permitindo uma comparação direta com os modelos locais e esparsos em termos de eficiência. Destaca-se que o modelo 2-LS-LSSVR já demonstra maior eficiência em relação ao modelo global linear. Além disso, observa-se que o modelo 2-LS-LSSVR reduz o tempo de processamento médio em aproximadamente 52% (de 25,4 ms para 12,2 ms) em comparação ao modelo global utilizando o mesmo kernel de Cauchy.

Em comparação com os resultados obtidos pelo modelo LSSVR em Murta *et al.* (2022), o presente estudo alcança resultados de  $R^2$  superiores em média. As melhorias obtidas para cada saída são aproximadamente 6,49% na média de  $R^2$  (de 0,7989 para 0,8508) para LE, 6,47% (de 0,7389 para 0,7867) para LRT, 2,72% (de 0,8603 para 0,8837) para LRT/LE e 4,95% (de 0,6883 para 0,7224) para AP. Com isso, observa-se pela Tabela 8 o modelo 2-LS-LSSVR se apresenta como a melhor técnica implementada de AM para a previsão de propriedades mecânicas de vergalhões de aço.

A Figura 22 apresenta gráficos de dispersão que comparam os valores preditos e observados em 100 rodadas para as quatro propriedades mecânicas, utilizando dois modelos distintos: 2-LS-LSSVR com kernel de Cauchy e LSSVR com kernel linear. Esses gráficos destacam o contraste entre um modelo não linear e um modelo linear, permitindo uma avaliação clara da eficácia de cada abordagem na predição das propriedades mecânicas.

Figura 19 – Desempenho dos modelos locais juntamente com as respectivas quantidades de VSs utilizados e o tempo gasto durante a previsão das saídas.

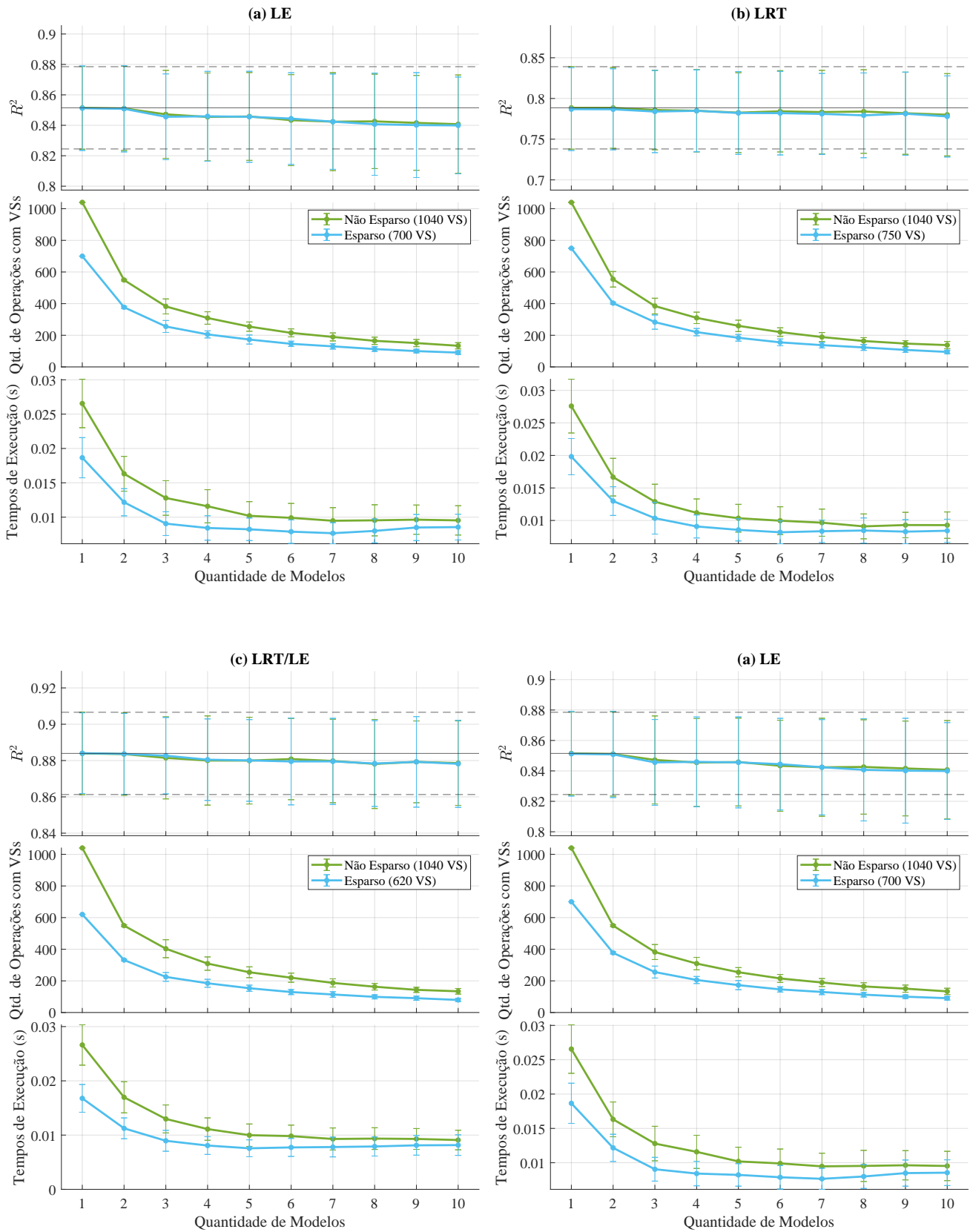
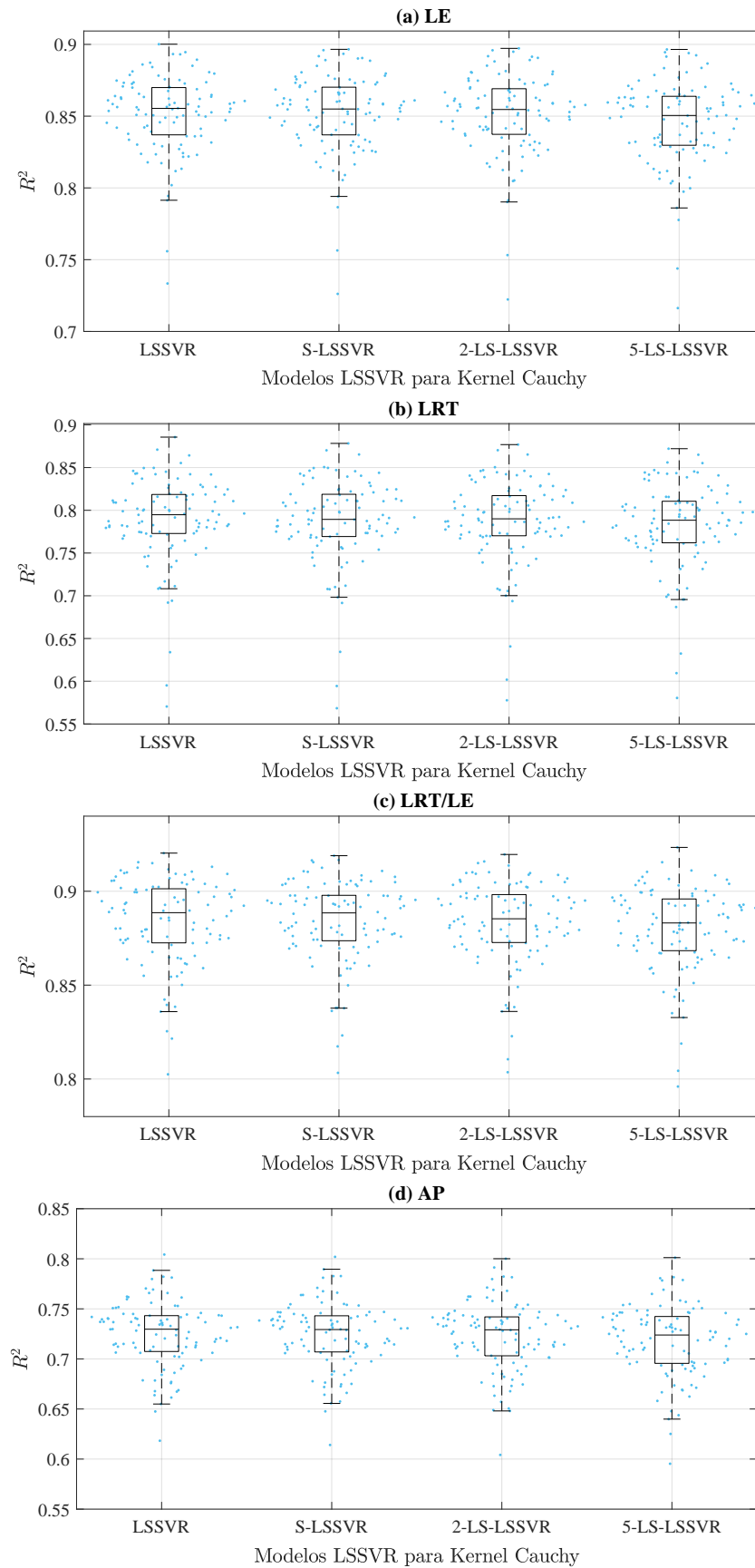
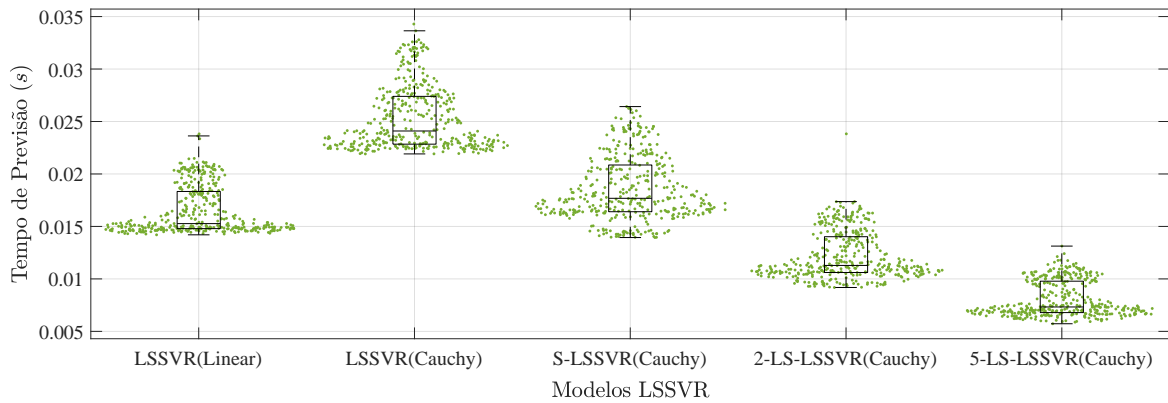


Figura 20 – *Boxplot* de desempenho para diferentes modelos LSSVR.



Fonte: Autoria própria.

Figura 21 – *Boxplot* dos tempos de execução dos diferentes modelos LSSVR durante a previsão das suas saídas.



Fonte: Autoria própria.

Tabela 8 – Comparação de desempenho entre diferentes modelos de AM.

Modelos	LE		LRT		LRT/LE		AP	
	$R^2_{mean}$	$R^2_{std}$	$R^2_{mean}$	$R^2_{std}$	$R^2_{mean}$	$R^2_{std}$	$R^2_{mean}$	$R^2_{std}$
LR <sup>1</sup>	0,5072	0,0515	0,2398	0,0440	0,7618	0,0289	0,4950	0,0424
MLP <sup>1</sup>	0,7472	0,0540	0,4455	0,0971	0,8428	0,0289	0,5978	0,0636
MLM <sup>2</sup>	0,8022	0,0368	0,5416	0,2165	0,8426	0,0317	0,6679	0,0595
SVR <sup>2</sup>	0,7296	0,0338	0,5180	0,0588	0,8469	0,0271	0,6456	0,0383
LSSVR <sup>2</sup>	0,7989	0,0342	0,7389	0,0630	0,8603	0,0240	0,6883	0,0315
2-LS-LSSVR	0,8508	0,0282	0,7867	0,0495	0,8837	0,0222	0,7224	0,0336

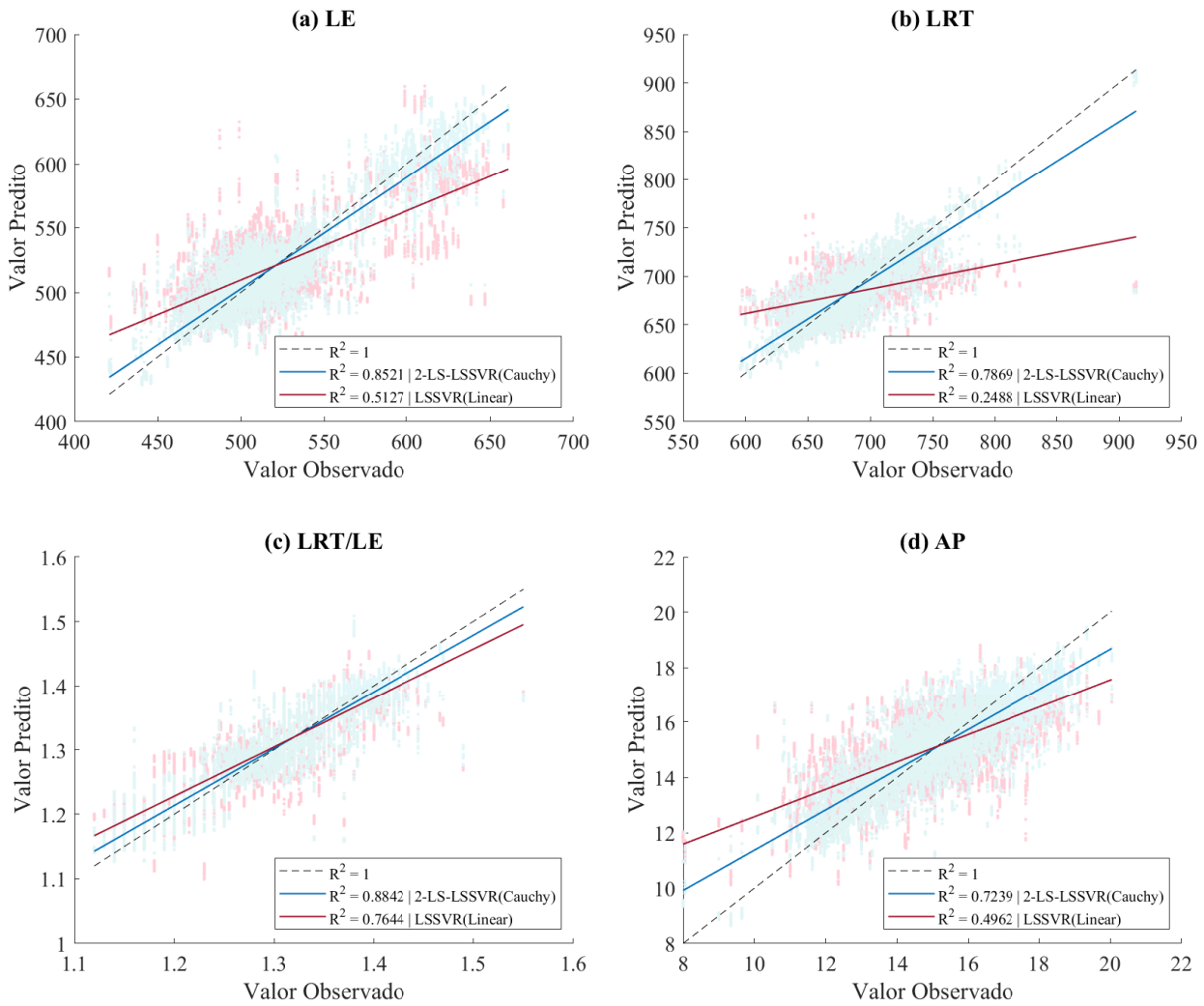
<sup>1</sup> Murta *et al.* (2021); <sup>2</sup> Murta *et al.* (2022).

Fonte: Autoria própria.

### 5.3 Conclusão do Capítulo

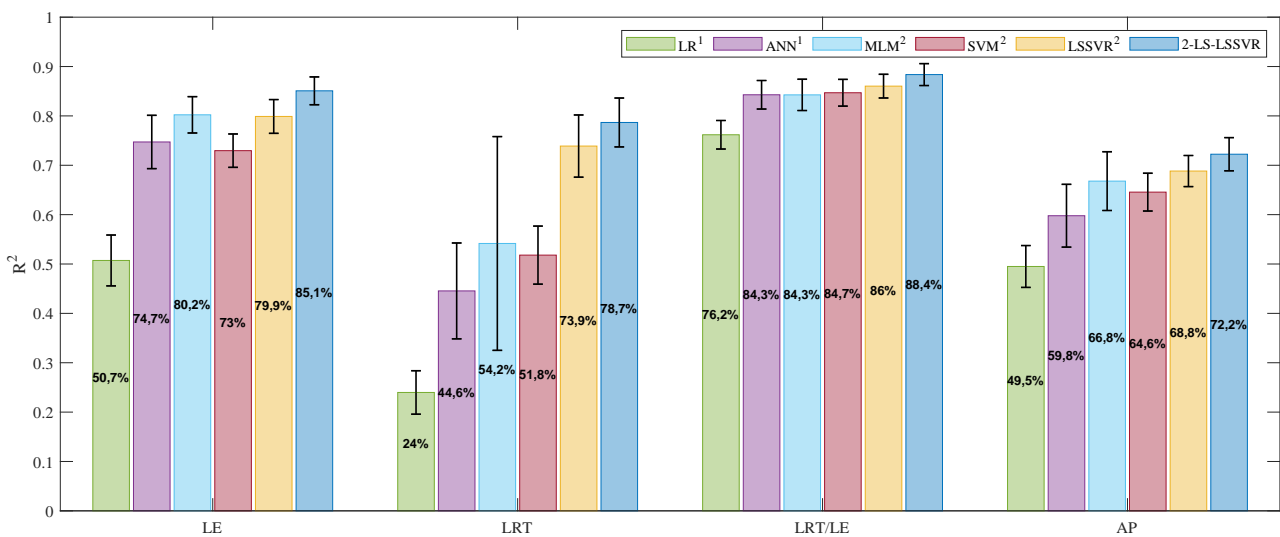
Os resultados deste capítulo confirmam que a combinação de métodos de esparsificação com modelagem local oferece vantagens significativas no custo computacional. A análise revela que, a partir de 5-LS-LSSVR, não há redução substancial nos custos computacionais, sugerindo a necessidade de explorar outros algoritmos de agrupamento. Além disso, abre caminho para estudos futuros focados na otimização de hiperparâmetros e na esparsificação específica por partição, com potencial para aprimorar ainda mais os resultados. Conclui-se que o modelo 2-LS-LSSVR se destaca como o novo estado da arte na previsão de propriedades mecânicas, conforme evidenciado pelos resultados alcançados, consolidando-se como o modelo mais eficiente entre os avaliados até o momento. A Figura 23 apresenta a comparação com a literatura, destacando o desempenho superior do modelo proposto.

Figura 22 – Gráficos de dispersão comparando os valores preditos e observados das 100 rodadas.



Fonte: Autoria própria.

Figura 23 – Comparação geral de desempenho dos modelos avaliados em valores de  $R^2$  médio e de desvio padrão para 100 experimentos com novos resultados.



<sup>1</sup> Murta *et al.* (2021); <sup>2</sup> Murta *et al.* (2022).

Fonte: Adaptado de Murta *et al.* (2022).

## 6 CONCLUSÃO E TRABALHOS FUTUROS

Neste capítulo, discutem-se as contribuições desta tese ao avanço do estado da arte na predição das propriedades mecânicas de vergalhões de aço, baseando-se em variáveis relacionadas à composição química e ao tratamento térmico aplicado. Ao final, são apresentadas sugestões para trabalhos futuros que podem expandir e aprimorar as descobertas desta pesquisa.

### 6.1 Contribuições da tese

Nesta tese, relatam-se os resultados de experimentos que avaliam estratégias para aprimorar o desempenho do modelo LSSVR, conforme discutido em estudos anteriores, devido à importância de sua aplicação na metalurgia e siderurgia. Enfatiza-se a importância da otimização dos hiperparâmetros durante o treinamento para maximizar o desempenho do modelo. Além disso, a investigação de diferentes funções de kernel resulta na escolha de um modelo mais simples, o modelo LSSVR com o kernel de Cauchy, que se mostra aproximadamente 24% mais eficiente no cálculo de previsão da saída ainda na abordagem global, além de apresentar uma melhoria na acurácia.

São demonstradas técnicas bem-sucedidas de esparsificação via QV e baseadas em ALD. Entre essas técnicas, destacam-se duas variantes do ALD propostas nesta tese, ALD4P e ALD4P-IO, que se mostram eficazes ao selecionar amostras informativas, reduzindo o conjunto de dados de treinamento original sem comprometer o desempenho do modelo. Em particular, o método ALD4P-IO permite uma abordagem híbrida de poda e inserção de VSs, o que é extremamente vantajoso em cenários práticos. Além disso, essa metodologia se adapta bem à manutenção contínua do modelo, facilitando a integração de novas amostras conforme elas são adquiridas. Isso é particularmente útil em ambientes não estacionários, onde os dados podem evoluir rapidamente e a atualização frequente do modelo é necessária para manter sua relevância e precisão.

Uma segunda contribuição desta tese, o desenvolvimento de modelos LSSVR locais, surge como uma forma eficiente de tornar o modelo de regressão mais parcimonioso na previsão, utilizando apenas os VSs representativos do *cluster* ao qual a amostra de teste pertence, reduzindo assim o número de operações necessárias para a estimativa. Observa-se uma redução média substancial nas operações de VSs, utilizando apenas 36,3% para a estimativa de LE, 38,8% para LRT, 31,8% para LRT/ LE e 36,7% para AP do total disponível de VSs. Além disso, essa técnica

pode ser aplicada em cenários com grande número de amostras para possibilitar a inversão da matriz de kernel e facilitar a otimização de hiperparâmetros.

O aprimoramento na previsão do modelo reforça que a composição química e os parâmetros de tratamento térmico neste conjunto de dados podem ser usados para prever as quatro principais propriedades mecânicas do aço. As melhorias obtidas são de aproximadamente 6,49% em média para  $R^2$  de LE, 6,47% para LRT, 2,72% para LRT/LE e 4,95% para AP, motivando a aplicação de novas técnicas neste conjunto de dados para superar os resultados apresentados neste estudo em trabalhos futuros. Por isso, o modelo 2-LS-LSSVR com o kernel de Cauchy é introduzido como o novo estado da arte para o conjunto de dados de fabricação de vergalhões de aço, alcançando os melhores resultados até o momento. Isso sugere que técnicas de esparsificação e modelagem local podem ser utilizadas simultaneamente sem afetar o desempenho do modelo de regressão.

Em conclusão, avalia-se que o objetivo geral deste trabalho tenha sido alcançado. A seguir, apresentam-se as contribuições para cada objetivo trabalhado:

- Propõem-se e testam-se novas funções de kernel em experimentos, resultando em melhorias significativas no desempenho preditivo do modelo;
- Implementam-se e avaliam-se duas novas abordagens ao método de esparsificação ALD, ALD4P e ALD4P-IO, resultando em uma esparsificação mais eficiente sem perda de precisão;
- Implementa-se a abordagem baseada em modelos locais, demonstrando redução significativa nos custos computacionais e manutenção de alta precisão preditiva;
- As análises demonstram que os métodos propostos resultam em avanços consideráveis na eficácia e eficiência do modelo LSSVR, confirmando sua contribuição significativa para a área de previsão de propriedades mecânicas de vergalhões de aço.

Com esses objetivos alcançados, a pesquisa reforça a relevância das técnicas de esparsificação e modelagem local para a otimização do desempenho do modelo LSSVR.

## 6.2 Trabalhos Futuros

Com base nas contribuições alcançadas, possíveis trabalhos futuros podem explorar as seguintes direções de pesquisa: (i) realização de uma análise abrangente de sensibilidade nos modelos de regressão baseados em LSSVR para entender como cada característica de entrada afeta a previsão das propriedades mecânicas; (ii) implementação dos métodos desenvolvidos

nesta tese para funcionamento em tempo real na planta de produção de vergalhões de aço, visando uma integração direta com o processo produtivo e a otimização contínua da qualidade dos produtos; (iii) desenvolvimento de arquiteturas de aprendizado profundo para a previsão de propriedades mecânicas de vergalhões de aço; (iv) desenvolvimento de técnicas de modelagem local baseadas em agrupamento suave; (v) investigação do desempenho dos modelos de regressão propostos nesta tese em outros conjuntos de dados no domínio da metalurgia e fabricação de aço, que poderiam fornecer *insights* valiosos e ampliar a aplicabilidade desses métodos.

Como desdobramento imediato, estão em andamento análises envolvendo modelos locais baseados em  $\varepsilon$ -SVR e  $\nu$ -SVR (Schölkopf *et al.*, 2000), além da aplicação das técnicas de esparsificação investigadas nesta tese como etapa de pré-processamento para tais modelos. Também será avaliado o desempenho de modelos randomizados, como a ELM e as *random fourier features* (Avron *et al.*, 2017), buscando compreender suas vantagens em termos de generalização e eficiência computacional.

Complementarmente, estudos comparativos envolvendo diferentes modelos de regressão e técnicas de comitê poderão ser conduzidos para uma avaliação mais abrangente da eficácia e da robustez das abordagens propostas em distintos cenários. Além disso, a integração de conhecimento de domínio e de perspectivas fornecidas por especialistas ao processo de modelagem representa uma direção promissora, capaz de aprimorar a precisão das previsões e oferecer uma compreensão mais profunda dos fatores subjacentes que influenciam as propriedades mecânicas do aço.

Por fim, em consonância com o compromisso de transparência e reprodutibilidade científica, o código-fonte e os conjuntos de dados utilizados foram disponibilizados em um repositório aberto, servindo como material de apoio e referência para pesquisas futuras na área.

## REFERÊNCIAS

- ABONYI, J.; NEMETH, S.; VINCZE, C.; ARVA, P. Process analysis and product quality estimation by self-organizing maps with an application to polyethylene production. **Computers in Industry**, Elsevier BV, v. 52, n. 3, p. 221–234, 2003. ISSN 0166-3615. Disponível em: [http://dx.doi.org/10.1016/S0166-3615\(03\)00128-3](http://dx.doi.org/10.1016/S0166-3615(03)00128-3).
- ALPAYDIN, E.; JORDAN, M. Local linear perceptrons for classification. **IEEE Transactions on Neural Networks**, Institute of Electrical and Electronics Engineers (IEEE), v. 7, n. 3, p. 788–794, 1996. ISSN 1941-0093. Disponível em: <http://dx.doi.org/10.1109/72.501737>.
- ARUMUGAM, D.; NAIK, D. L.; SAJID, H. U.; KIRAN, R. Relationship between nano and macroscale properties of postfire ASTM A36 steels. **Journal of Materials in Civil Engineering**, American Society of Civil Engineers (ASCE), v. 34, n. 6, 2022. ISSN 1943-5533. Disponível em: [http://dx.doi.org/10.1061/\(asce\)mt.1943-5533.0004218](http://dx.doi.org/10.1061/(asce)mt.1943-5533.0004218).
- AVRON, H.; KAPRALOV, M.; MUSCO, C.; MUSCO, C.; VELINGKER, A.; ZANDIEH, A. Random Fourier features for kernel ridge regression: Approximation bounds and statistical guarantees. In: PRECUP, D.; TEH, Y. W. (Ed.). **Proceedings of the 34th International Conference on Machine Learning**. PMLR, 2017. (Proceedings of Machine Learning Research, v. 70), p. 253–262. Disponível em: <https://proceedings.mlr.press/v70/avron17a.html>.
- AYAT, N.; CHERIET, M.; REMAKI, L.; SUEN, C. KMOD - a new support vector machine kernel with moderate decreasing for pattern recognition. application to digit image recognition. In: **Proceedings of Sixth International Conference on Document Analysis and Recognition**. IEEE Comput. Soc. (ICDAR-01), p. 1215–1219. Disponível em: <http://dx.doi.org/10.1109/ICDAR.2001.953976>.
- AYAT, N.; CHERIET, M.; SUEN, C. KMOD - a two-parameter SVM kernel for pattern recognition. In: **Object recognition supported by user interaction for service robots**. IEEE Comput. Soc, 2002. (ICPR-02). Disponível em: <http://dx.doi.org/10.1109/ICPR.2002.1047860>.
- BAHLEDA, F.; BUJŇÁKOVÁ, P.; KOTEŠ, P.; HASAJOVÁ, L.; NOVÝ, F. Mechanical properties of cast-in anchor bolts manufactured of reinforcing tempcore steel. **Materials**, v. 12, n. 13, 2019. ISSN 1996-1944. Disponível em: <https://www.mdpi.com/1996-1944/12/13/2075>.
- BESSA, R.; BARRETO, G. A. Sobre a aplicação de regressão kernel na predição das propriedades mecânicas de vergalhões de aço: Novos resultados. In: **Anais do XVI Congresso Brasileiro de Inteligência Computacional**. SBIC, 2023. (CBIC 2023). Disponível em: <http://dx.doi.org/10.21528/CBIC2023-125>.
- BESSA, R.; BARRETO, G. A.; COELHO, D. N.; MOURA, E. P. de; MURTA, R. H. F. On least squares support vector regression for predicting mechanical properties of steel rebars. **Metals**, MDPI AG, v. 14, n. 6, p. 695, jun. 2024. ISSN 2075-4701. Disponível em: <http://dx.doi.org/10.3390/met14060695>.
- BLACK, J. T.; KOHSER, R. A. **DeGarmo's materials and processes in manufacturing**. 13. ed. Nashville, TN: John Wiley & Sons, 2021. ISBN 978-1-119-49282-5.
- BOTO, F.; MURUA, M.; GUTIERREZ, T.; CASADO, S.; CARRILLO, A.; ARTEAGA, A. Data driven performance prediction in steel making. **Metals**, MDPI AG, v. 12, n. 2, p. 172, 2022. ISSN 2075-4701. Disponível em: <http://dx.doi.org/10.3390/met12020172>.

BRABANTER, K. D.; KARSMARKERS, P.; OJEDA, F.; ALZATE, C.; BRABANTER, J. D.; PELCKMANS, K.; MOOR, B. D.; VANDEWALLE, J.; SUYKENS, J. A. **LS-SVMlab toolbox user's guide: version 1.8**. Leuven-Heverlee, Belgium: Katholieke Universiteit Leuven, 2011.

CALLISTER, W. D.; RETHWISCH, D. G. **Fundamentals of materials science and engineering**. 5. ed. Nashville, TN: John Wiley & Sons, 2015.

CHICCO, D.; WARRENS, M. J.; JURMAN, G. The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. **PeerJ Computer Science**, PeerJ, v. 7, p. e623, jul. 2021. ISSN 2376-5992. Disponível em: <http://dx.doi.org/10.7717/peerj-cs.623>.

CHOU, P.-Y.; TSAI, J.-T.; CHOU, J.-H. Modeling and optimizing tensile strength and yield point on a steel bar using an artificial neural network with taguchi particle swarm optimizer. **IEEE Access**, Institute of Electrical and Electronics Engineers (IEEE), v. 4, p. 585–593, 2016. ISSN 2169-3536. Disponível em: <http://dx.doi.org/10.1109/ACCESS.2016.2521162>.

COELHO, D. N.; BARRETO, G. A. Approximate linear dependence as a design method for kernel prototype-based classifiers. Springer International Publishing, p. 241–250, 2019. ISSN 2194-5365. Disponível em: [http://dx.doi.org/10.1007/978-3-030-19642-4\\_24](http://dx.doi.org/10.1007/978-3-030-19642-4_24).

COELHO, D. N.; BARRETO, G. A.; MEDEIROS, C. M. S. Detection of short circuit faults in 3-phase converter-fed induction motors using kernel SOMs. In: **2017 12th International Workshop on Self-Organizing Maps and Learning Vector Quantization, Clustering and Data Visualization (WSOM)**. IEEE, 2017. Disponível em: <http://dx.doi.org/10.1109/WSOM.2017.8020016>.

CRAWFORD, B.; SOURKI, R.; KHAYYAM, H.; MILANI, A. S. A machine learning framework with dataset-knowledgeability pre-assessment and a local decision-boundary crispness score: An industry 4.0-based case study on composite autoclave manufacturing. **Computers in Industry**, Elsevier BV, v. 132, p. 103510, 2021. ISSN 0166-3615. Disponível em: <http://dx.doi.org/10.1016/j.compind.2021.103510>.

DABIRI, H.; FARHANGI, V.; MORADI, M. J.; ZADEHMOHAMAD, M.; KARAKOUZIAN, M. Applications of decision tree and random forest as tree-based machine learning techniques for analyzing the ultimate strain of spliced and non-spliced reinforcement bars. **Applied Sciences**, MDPI AG, v. 12, n. 10, p. 4851, 2022. ISSN 2076-3417. Disponível em: <http://dx.doi.org/10.3390/app12104851>.

DABIRI, H.; KHEYRODDIN, A.; FARAMARZI, A. Predicting tensile strength of spliced and non-spliced steel bars using machine learning- and regression-based methods. **Construction and Building Materials**, Elsevier BV, v. 325, p. 126835, 2022. ISSN 0950-0618. Disponível em: <http://dx.doi.org/10.1016/j.conbuildmat.2022.126835>.

DU, J. L.; FENG, Y. L.; ZHANG, M. Construction of a machine-learning-based prediction model for mechanical properties of ultra-fine-grained Fe–C alloy. **Journal of Materials Research and Technology**, Elsevier BV, v. 15, p. 4914–4930, 2021. ISSN 2238-7854. Disponível em: <http://dx.doi.org/10.1016/j.jmrt.2021.10.111>.

ENGEL, Y.; MANNOR, S.; MEIR, R. The kernel recursive least-squares algorithm. **IEEE Transactions on Signal Processing**, Institute of Electrical and Electronics Engineers (IEEE), v. 52, n. 8, p. 2275–2285, ago. 2004. ISSN 1053-587X. Disponível em: <http://dx.doi.org/10.1109/TSP.2004.830985>.

FERRARO, M. B.; GIORDANI, P. Soft clustering. **WIREs Computational Statistics**, Wiley, v. 12, n. 1, jun. 2019. ISSN 1939-0068. Disponível em: <http://dx.doi.org/10.1002/wics.1480>.

GANGQUAN, S.; ZHANG, G.; JIANQUAN, S.; HONG, G. Partitioned LSSVR modeling method for self adaptation under multi operating conditions. In: **2015 IEEE 7th International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)**. IEEE, 2015. Disponível em: <http://dx.doi.org/10.1109/iccis.2015.7274633>.

HONG, D.; KWON, S.; YIM, C. Exploration of machine learning to predict hot ductility of cast steel from chemical composition and thermal conditions. **Metals and Materials International**, Springer Science and Business Media LLC, v. 27, n. 2, p. 298–305, 2021. ISSN 2005-4149. Disponível em: <http://dx.doi.org/10.1007/s12540-020-00713-w>.

ISMAIL, S.; SHABRI, A.; SAMSUDIN, R. A hybrid model of self-organizing maps (SOM) and least square support vector machine (LSSVM) for time-series forecasting. **Expert Systems with Applications**, Elsevier BV, v. 38, n. 8, p. 10574–10578, 2011. ISSN 0957-4174. Disponível em: <http://dx.doi.org/10.1016/j.eswa.2011.02.107>.

JACOBS, R. A.; JORDAN, M. I.; NOWLAN, S. J.; HINTON, G. E. Adaptive mixtures of local experts. **Neural Computation**, MIT Press - Journals, v. 3, n. 1, p. 79–87, 1991. ISSN 1530-888X. Disponível em: <http://dx.doi.org/10.1162/neco.1991.3.1.79>.

KAUFMAN, L.; ROUSSEEUW, P. J. **Finding Groups in Data: An Introduction to Cluster Analysis**. Wiley, 1990. ISSN 1940-6347. ISBN 9780470316801. Disponível em: <http://dx.doi.org/10.1002/9780470316801>.

KENNEDY, J.; EBERHART, R. Particle swarm optimization. In: **Proceedings of ICNN'95 - International Conference on Neural Networks**. IEEE. Disponível em: <https://doi.org/10.1109/icnn.1995.488968>.

KHOZANI, Z. S.; HOSSEINJANZADEH, H.; MOHTAR, W. H. M. W. Shear force estimation in rough boundaries using SVR method. **Applied Water Science**, Springer Science and Business Media LLC, v. 9, n. 8, out. 2019. ISSN 2190-5495. Disponível em: <http://dx.doi.org/10.1007/s13201-019-1056-z>.

LAGARIAS, J. C.; REEDS, J. A.; WRIGHT, M. H.; WRIGHT, P. E. Convergence properties of the nelder–mead simplex method in low dimensions. **SIAM Journal on Optimization**, Society for Industrial & Applied Mathematics (SIAM), v. 9, n. 1, p. 112–147, 1998. Disponível em: <https://doi.org/10.1137/s1052623496303470>.

LALAM, S.; TIWARI, P. K.; SAHOO, S.; DALAL, A. K. Online prediction and monitoring of mechanical properties of industrial galvanised steel coils using neural networks. **Ironmaking & Steelmaking**, Informa UK Limited, v. 46, n. 1, p. 89–96, 2019. Disponível em: <https://doi.org/10.1080/03019233.2017.1342424>.

LI, Y.; LU, Y.; CHEN, J. A deep learning approach for real-time rebar counting on the construction site based on YOLOv3 detector. **Automation in Construction**, Elsevier BV, v. 124, p. 103602, 2021. ISSN 0926-5805. Disponível em: <http://dx.doi.org/10.1016/j.autcon.2021.103602>.

LIANG, D.; XUE, F. Integrating automated machine learning and interpretability analysis in architecture, engineering and construction industry: A case of identifying failure modes of reinforced concrete shear walls. **Computers in Industry**, Elsevier BV, v. 147, p. 103883, 2023. ISSN 0166-3615. Disponível em: <http://dx.doi.org/10.1016/j.compind.2023.103883>.

MACQUEEN, J. B. Classification and analysis of multivariate observations. In: CAM, L. M. L.; NEYMAN, J. (Ed.). **Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability**. Berkeley, CA: University of California Press, 1967. v. 1, p. 281–297.

MALANG, J.; YEO, W. S.; CHUA, Z. Y.; NANDONG, J.; SAPTORO, A. A comparison study between different kernel functions in the least square support vector regression model for penicillin fermentation process. **MATEC Web of Conferences**, EDP Sciences, v. 377, p. 01025, 2023. ISSN 2261-236X. Disponível em: <http://dx.doi.org/10.1051/mateconf/202337701025>.

MURPHY, K. P. **Machine Learning**. London, England: MIT Press, 2012. (Adaptive Computation and Machine Learning series).

MURTA, R. H. F. **Previsão das propriedades mecânicas de vergalhões de aço utilizados na construção civil**. Tese (Doutorado) – Universidade Federal do Ceará, Fortaleza, Ceará, 2022. Pós-Graduação em Engenharia e Ciência de Materiais.

MURTA, R. H. F.; BRAGA, F. D.; MAIA, P. P. N.; DIOGENES, O. B. F.; MOURA, E. P. de. Mathematical modelling for predicting mechanical properties in rebar manufacturing. **Ironmaking & Steelmaking**, Informa UK Limited, v. 48, n. 2, p. 161–169, 2021. Disponível em: <https://doi.org/10.1080/03019233.2020.1749357>.

MURTA, R. H. F.; MOURA, E. P. de; BARRETO, G. A. Mechanical properties prediction in rebar using kernel-based regression models. **Ironmaking & Steelmaking**, Informa UK Limited, v. 49, n. 10, p. 1011–1020, 2022. Disponível em: <https://doi.org/10.1080/03019233.2022.2075691>.

NAVON, R.; RUBINOVITZ, Y. A.; COFFLER, M. Reinforcement-bar manufacture: From design to optimized production. **International Journal of Computer Integrated Manufacturing**, Informa UK Limited, v. 11, n. 4, p. 326–333, jan. 1998. ISSN 1362-3052. Disponível em: <http://dx.doi.org/10.1080/095119298130660>.

NELDER, J. A.; MEAD, R. A simplex method for function minimization. **The Computer Journal**, Oxford University Press (OUP), v. 7, n. 4, p. 308–313, 1965. Disponível em: <https://doi.org/10.1093/comjnl/7.4.308>.

NGU, J. C. Y.; YEO, W. S.; THIEN, T. F.; NANDONG, J. A comprehensive overview of the applications of kernel functions and data-driven models in regression and classification tasks in the context of software sensors. **Applied Soft Computing**, Elsevier BV, v. 164, p. 111975, out. 2024. ISSN 1568-4946. Disponível em: <http://dx.doi.org/10.1016/j.asoc.2024.111975>.

NIEVES, L. **Environmental Product Declaration**. 2016. Gerdau Rebar, North American Steel Mill. Disponível em: [https://www2.gerdau.com/sites/gln\\_gerdau/files/downloadable\\_files/EPD%20Rebar%20Gerdau%20St.%20Paul%20-%20Expires%202021.pdf](https://www2.gerdau.com/sites/gln_gerdau/files/downloadable_files/EPD%20Rebar%20Gerdau%20St.%20Paul%20-%20Expires%202021.pdf). Acesso em: 22/06/2024.

OLIVEIRA, S. A.; GOMES, J. P.; NETO, A. R. R. Sparse least-squares support vector machines via accelerated segmented test: A dual approach. **Neurocomputing**, Elsevier BV, v. 321, p. 308–320, 2018. ISSN 0925-2312. Disponível em: <http://dx.doi.org/10.1016/j.neucom.2018.08.081>.

Rocha Neto, A. R.; BARRETO, G. A. Opposite maps: Vector quantization algorithms for building reduced-set SVM and LSSVM classifiers. **Neural Processing Letters**, Springer Science and Business Media LLC, v. 37, n. 1, p. 3–19, 2012. Disponível em: <https://doi.org/10.1007/s11063-012-9265-6>.

RUSTAM, Z.; KINTANDANI, P. Application of support vector regression in Indonesian stock price prediction with feature selection using particle swarm optimisation. **Modelling and Simulation in Engineering**, Hindawi Limited, v. 2019, p. 1–5, 2019. ISSN 1687-5605. Disponível em: <http://dx.doi.org/10.1155/2019/8962717>.

SAJID, H. U.; NAIK, D. L.; KIRAN, R. Microstructure–mechanical property relationships for post-fire structural steels. **Journal of Materials in Civil Engineering**, American Society of Civil Engineers (ASCE), v. 32, n. 6, 2020. ISSN 1943-5533. Disponível em: [http://dx.doi.org/10.1061/\(ASCE\)MT.1943-5533.0003190](http://dx.doi.org/10.1061/(ASCE)MT.1943-5533.0003190).

SAUNDERS, C.; GAMMERMAN, A.; VOVK, V. Ridge regression learning algorithm in dual variables. In: **Proceedings of the Fifteenth International Conference on Machine Learning**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998. (ICML '98), p. 515–521. ISBN 1558605568.

SAYED, A. M.; DIAB, H. M. Modeling of the axial load capacity of RC columns strengthened with steel jacketing under preloading based on FE simulation. **Modelling and Simulation in Engineering**, Hindawi Limited, v. 2019, p. 1–8, 2019. ISSN 1687-5605. Disponível em: <http://dx.doi.org/10.1155/2019/8653247>.

SCHÖLKOPF, B.; SMOLA, A. J.; WILLIAMSON, R. C.; BARTLETT, P. L. New Support Vector Algorithms. **Neural Computation**, v. 12, n. 5, p. 1207–1245, 05 2000. ISSN 0899-7667. Disponível em: <https://doi.org/10.1162/089976600300015565>.

SGROTT, D. M.; CERQUEIRA, F. M.; MIRANDA, F. J. F.; FILHO, J. F. S.; PARPINELLI, R. S. Modelling of steels using artificial neural networks and automated machine learning. In: \_\_\_\_\_. **Advances in Intelligent Systems and Computing**. Springer International Publishing, 2021. p. 659–668. ISBN 9783030730505. Disponível em: [http://dx.doi.org/10.1007/978-3-030-73050-5\\_64](http://dx.doi.org/10.1007/978-3-030-73050-5_64).

SILVA, K.; SERPA, P.; SGROTT, D.; CERQUEIRA, F.; MIRANDA, F.; FILHO, J. F. S.; PARPINELLI, R. Ensemble of artificial neural networks and automl for predicting steel properties. In: **Anais do XVI Congresso Brasileiro de Inteligência Computacional**. SBIC, 2023. (CBIC 2023). Disponível em: <http://dx.doi.org/10.21528/CBIC2023-149>.

SOUZA, A. H. d.; CORONA, F.; BARRETO, G. A.; MICHE, Y.; LENDASSE, A. Minimal learning machine: A novel supervised distance-based approach for regression and classification. **Neurocomputing**, Elsevier BV, v. 164, p. 34–44, 2015. ISSN 0925-2312. Disponível em: <http://dx.doi.org/10.1016/j.neucom.2014.11.073>.

SOUZA, S. Xavier-de; SUYKENS, J.; VANDEWALLE, J.; BOLLE, D. Coupled simulated annealing. **IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)**, Institute of Electrical and Electronics Engineers (IEEE), v. 40, n. 2, p. 320–335, abr. 2010. ISSN 1083-4419. Disponível em: <http://dx.doi.org/10.1109/TSMCB.2009.2020435>.

SUN, S.; WANG, S.; ZHANG, G.; ZHENG, J. A decomposition-clustering-ensemble learning approach for solar radiation forecasting. **Solar Energy**, Elsevier BV, v. 163, p. 189–199, mar. 2018. ISSN 0038-092X. Disponível em: <http://dx.doi.org/10.1016/j.solener.2018.02.006>.

- SUYKENS, J.; BRABANTER, J. D.; LUKAS, L.; VANDEWALLE, J. Weighted least squares support vector machines: robustness and sparse approximation. **Neurocomputing**, Elsevier BV, v. 48, n. 1–4, p. 85–105, out. 2002. ISSN 0925-2312. Disponível em: [http://dx.doi.org/10.1016/s0925-2312\(01\)00644-0](http://dx.doi.org/10.1016/s0925-2312(01)00644-0).
- SUYKENS, J.; VANDEWALLE, J. Least squares support vector machine classifiers. **Neural Processing Letters**, Springer Science and Business Media LLC, v. 9, n. 3, p. 293–300, 1999. Disponível em: <https://doi.org/10.1023/a:1018628609742>.
- TERVEN, J.; CÓRDOVA-ESPARZA, D.-M.; ROMERO-GONZÁLEZ, J.-A. A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS. **Machine Learning and Knowledge Extraction**, v. 5, n. 4, p. 1680–1716, 2023. ISSN 2504-4990. Disponível em: <https://www.mdpi.com/2504-4990/5/4/83>.
- VAPNIK, V. An overview of statistical learning theory. **IEEE Transactions on Neural Networks**, Institute of Electrical and Electronics Engineers (IEEE), v. 10, n. 5, p. 988–999, 1999. ISSN 1045-9227. Disponível em: <http://dx.doi.org/10.1109/72.788640>.
- WANG, L.; MU, Z.; GUO, H. Application of support vector machine in the prediction of mechanical property of steel materials. **Journal of University of Science and Technology Beijing, Mineral, Metallurgy, Material**, Elsevier BV, v. 13, n. 6, p. 512–515, 2006. ISSN 1005-8850. Disponível em: [http://dx.doi.org/10.1016/S1005-8850\(06\)60104-7](http://dx.doi.org/10.1016/S1005-8850(06)60104-7).
- WANG, X.; LI, H.; PAN, T.; SU, H.; MENG, H. Material quality filter model: Machine learning integrated with expert experience for process optimization. **Metals**, MDPI AG, v. 13, n. 5, p. 898, 2023. ISSN 2075-4701. Disponível em: <http://dx.doi.org/10.3390/met13050898>.
- XIE, Q.; SUVARNA, M.; LI, J.; ZHU, X.; CAI, J.; WANG, X. Online prediction of mechanical properties of hot rolled steel plate using machine learning. **Materials & Design**, Elsevier BV, v. 197, p. 109201, 2021. Disponível em: <https://doi.org/10.1016/j.matdes.2020.109201>.
- XIONG, Z.; LI, J.; ZHAO, P.; LI, Y. Prediction of mechanical properties of aluminium alloy strip using the extreme learning machine model optimized by the gray wolf algorithm. **Advances in Materials Science and Engineering**, Hindawi Limited, v. 2023, p. 1–16, 2023. ISSN 1687-8434. Disponível em: <http://dx.doi.org/10.1155/2023/5952072>.
- XU, Z.-W.; LIU, X.-M.; ZHANG, K. Mechanical properties prediction for hot rolled alloy steel using convolutional neural network. **IEEE Access**, Institute of Electrical and Electronics Engineers (IEEE), v. 7, p. 47068–47078, 2019. ISSN 2169-3536. Disponível em: <http://dx.doi.org/10.1109/ACCESS.2019.2909586>.
- ZHANG, G.; LI, Y.; CUI, D.; MAO, S.; HUANG, G.-B. R-elmnet: Regularized extreme learning machine network. **Neural Networks**, v. 130, p. 49–59, 2020. ISSN 0893-6080. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0893608020302240>.
- ZHENG, W.; WANG, C.; LIU, D. Combustion process modeling based on deep sparse least squares support vector regression. **Engineering Applications of Artificial Intelligence**, Elsevier BV, v. 132, p. 107869, 2024. ISSN 0952-1976. Disponível em: <http://dx.doi.org/10.1016/j.engappai.2024.107869>.
- ZHOU, T.; FENG, G.; ZHANG, S. Models LSSVR and PLSSVR with heteroscedastic gaussian noise characteristics and its application for short-term wind-speed forecasting. **IEEE Access**,

Institute of Electrical and Electronics Engineers (IEEE), v. 11, p. 86470–86479, 2023. ISSN 2169-3536. Disponível em: <http://dx.doi.org/10.1109/ACCESS.2023.3290020>.

ZHU, Z.; LIANG, Y.; ZOU, J. Modeling and composition design of low-alloy steel's mechanical properties based on neural networks and genetic algorithms. **Materials**, MDPI AG, v. 13, n. 23, p. 5316, 2020. ISSN 1996-1944. Disponível em: <http://dx.doi.org/10.3390/ma13235316>.

## APÊNDICE A – SINTONIA DE HIPERPÂMETROS COM *PSO+SIMPLEX*

A metaheurística PSO é um algoritmo de otimização baseado em população formada por um enxame de partículas, ou seja, um conjunto de possíveis soluções, que se movem pelo espaço de busca de maneira cooperativa à procura de soluções ótimas. Assim, a posição de cada partícula no enxame representa uma solução candidata para o problema de otimização tratado e é atribuída inicialmente de forma aleatória dentro do espaço de busca.

Na iteração  $t$ ,  $\mathbf{p}_i(t) \in \mathbb{R}^{N_d}$  denota a  $i$ -ésima partícula de dimensão  $N_d$ . A posição dessa partícula é modificada pela adição da parcela referente à velocidade,  $\mathbf{v}_i(t+1) \in \mathbb{R}^{N_d}$ , à sua posição atual, i.e.

$$\mathbf{p}_i(t+1) = \mathbf{p}_i(t) + \mathbf{v}_i(t+1). \quad (\text{A.1})$$

O vetor velocidade norteia o processo de otimização, incorporando o conhecimento experiencial da partícula e as informações compartilhadas socialmente com seus vizinhos.

O desempenho individual da partícula é avaliado usando uma função objetivo pré-definida,  $f_o(\cdot) : \mathbb{R}^{N_d} \rightarrow \mathbb{R}$ . Considerando um problema de minimização, a melhor posição da  $i$ -ésima partícula desde a primeira iteração,  $\mathbf{p}_i^*$ , é expressa por

$$\mathbf{p}_i^*(t+1) = \begin{cases} \mathbf{p}_i^*(t) & \text{if } f(\mathbf{p}_i(t+1)) \geq f(\mathbf{p}_i^*(t)) \\ \mathbf{p}_i(t+1) & \text{if } f(\mathbf{p}_i(t+1)) < f(\mathbf{p}_i^*(t)). \end{cases} \quad (\text{A.2})$$

Para o PSO global, cada partícula tem o conhecimento da situação global pela componente da melhor posição já encontrada pelo enxame, que é denotada por  $\mathbf{g}$ . Isso é,  $\mathbf{g}(t) \in \{\mathbf{p}_0^*(t), \dots, \mathbf{p}_{N_p}^*(t)\}$ , tal que

$$f_o(\mathbf{g}(t)) = \min \left\{ f_o(\mathbf{p}_0^*(t)), \dots, f(\mathbf{p}_{N_p}^*(t)) \right\}, \quad (\text{A.3})$$

onde  $N_p$  é a quantidade de partículas do enxame.

Com isso, a velocidade da  $i$ -ésima partícula pode ser calculada como

$$\begin{aligned} \mathbf{v}_i(t+1) = & \mathbf{v}_i(t) + c_1 \boldsymbol{\Psi}_1(t) \odot [\mathbf{p}_i^*(t) - \mathbf{p}_i(t)] \\ & + c_2 \boldsymbol{\Psi}_2(t) \odot [\mathbf{g}(t) - \mathbf{p}_i(t)], \end{aligned} \quad (\text{A.4})$$

em que  $c_1, c_2 \in \mathbb{R}$  são constantes positivas de aceleração usadas para dimensionar a contribuição dos componentes cognitivos e sociais, respectivamente. Os vetores  $\boldsymbol{\Psi}_1(t), \boldsymbol{\Psi}_2(t) \in \mathbb{R}^{N_p}$  têm seus elementos amostrados de uma distribuição  $\sim U(0, 1)$  a cada iteração  $t$  e o operador  $\odot$

representa uma multiplicação elemento a elemento. Essas operações são repetidas até que certos critérios de parada sejam satisfeitos, como um número máximo de iterações ou uma convergência satisfatória.

Seguindo para método *Simplex* de Nelder-Mead, utiliza-se a mesma função objetivo considerada para o algoritmo PSO neste paradigma de otimização. Um *Simplex* é um polígono convexo que contém  $N_d + 1$  vértices, onde  $N_d$  é o número de variáveis da função objetivo. Cada vértice representa uma solução candidata, em que um dos vértices é atribuído à melhor solução encontrada pelo PSO e os demais são atribuídos de forma aleatória na presente abordagem. O método realiza uma série de operações de reflexão, expansão, contração e redução para atualizar os vértices do simplex e aproximar-se do mínimo da função objetivo, como bem descrito por Lagarias *et al.* (1998).