

# Sobre a Aplicação de Regressão Kernel na Predição das Propriedades Mecânicas de Vergalhões de Aço: Novos Resultados

Renan Bessa

*Pós-Graduação em Eng. de Teleinformática (PPGETI)*  
*Universidade Federal do Ceará (UFC)*  
Campus do Pici, Centro de Tecnologia  
renanbessa@alu.ufc.br

Guilherme A. Barreto

*Pós-Graduação em Eng. de Teleinformática (PPGETI)*  
*Universidade Federal do Ceará (UFC)*  
Campus do Pici, Centro de Tecnologia  
gbarreto@ufc.br

Raphaella H. F. Murta

*Pós-Graduação em Eng. e Ciência de Materiais (PPGECM)*  
*Universidade Federal do Ceará (UFC)*  
Campus do Pici, Centro de Tecnologia  
raphaella\_hermonth@hotmail.com

Elineudo P. de Moura

*Pós-Graduação em Eng. e Ciência de Materiais (PPGECM)*  
*Universidade Federal do Ceará (UFC)*  
Campus do Pici, Centro de Tecnologia  
elineudo@ufc.br

**Resumo**—As propriedades mecânicas do aço dependem fortemente da composição química e dos parâmetros utilizados durante o processamento termomecânico. Tendo em vista garantir a qualidade e redução de custos na fabricação do aço, inúmeros estudos têm obtido sucesso na predição dessas propriedades por intermédio de técnicas de aprendizado de máquina, que são capazes de lidar com o grande número de variáveis envolvidas. Neste artigo, revisita-se a regressão kernel, bem sucedida em estudos recentes, em uma nova abordagem mais eficaz e eficiente.

**Index Terms**—Aço, propriedades mecânicas, regressão baseada em kernel, quantização vetorial, aprendizado de máquina.

## I. INTRODUÇÃO

A predição de propriedades mecânicas na fabricação do aço é um tema crucial na indústria de materiais e engenharia metalúrgica, sendo bastante dependente da escolha do material e dos processos empregados na produção. Dentre as principais propriedades de interesse para garantir a qualidade do aço, destacam-se o limite de escoamento (*yield strength*, YS), o limite de resistência à tração (*ultimate tensile strength*, UTS), a razão UTS/YS e o alongamento percentual (*percent elongation*, PE) [1, 2].

As grandezas YS e UTS são importantes indicadores de desempenho dos materiais em situações de carga, representando o ponto de início da deformação permanente e a carga máxima que o material pode suportar antes de falhar, respectivamente. Já a razão UTS/YS é uma medida da ductilidade do material, enquanto o PE é um indicador da capacidade do material em se deformar sem quebrar [3].

Agradecimento as seguintes instituições pelo suporte financeiro: CAPES (Código de Financiamento 001), CNPq (No.309379/2019-9) e FUNCAP (PS1-0186-00103.01.00/21).

As técnicas de aprendizado de máquina têm se apresentado bastante eficazes na estimação dessas propriedades, por exemplo, em chapas de aço laminadas a quente [4], em bobinas de aço galvanizado laminadas a frio [5] e em vergalhões de aço laminados a quente [1, 2], que é o foco deste trabalho. Em Murta et al. [1], são aplicados os métodos de regressão linear (*linear regression*, LR) e redes neurais artificiais (*artificial neural network*, ANN), especificamente uma rede perceptron com duas camadas ocultas, utilizando como entrada a composição química e o tratamento termomecânico dos vergalhões. Mais recentemente, em Murta et al. [2], avaliou-se três modelos de regressão baseados em kernel: a máquina de aprendizado mínimo (*minimal learning machine*, MLM), a regressão por vetor suporte (*support vector regression*, SVR) e a regressão por vetor suporte via mínimos quadrados (*least-squares support vector regression*, LSSVR), alcançando um aprimoramento nos resultados anteriores com o destaque para o modelo LSSVR.

Neste artigo, revisita-se o modelo LSSVR aplicado ao mesmo conjunto de dados de propriedades mecânicas de vergalhões de aço de Murta et al. [1, 2] com o objetivo torná-lo mais eficiente e melhorar seus resultados. Dessa maneira, a organização do restante deste trabalho é a seguinte. Na Seção II, é apresentada uma breve descrição do conjunto de dados. Na Seção III, encontra-se uma introdução sobre métodos de kernel via mínimos quadrados. Na Seção IV, apresentam-se formas de reduzir o custo computacional desses modelos por intermédio de algoritmos de quantização vetorial. Na Seção V, é descrita uma abordagem de busca para otimizar os hiperparâmetros dos métodos de kernel. Por fim, tem-se as seções de resultados, discussões e conclusões.

TABELA I: Parâmetros da laminação do tratamento térmico do vergalhão do aço.

Variáveis de Entrada	Mín.	Máx.	Média	D.P.
Área da seção transversal [mm <sup>2</sup> ]:	50	199	93,2	46,5
Temp. final de laminação [°C]:	742,7	1081,5	1031,9	21,9
Temp. inicial do trat. térmico [°C]:	654,7	961,8	920,7	21,1
Temp. final do trat. térmico [°C]:	337,6	584,1	436,8	47,5
Pressão de água do trat. tér. [Bar]:	0	15,5	5,8	5,0
Fluxo da água do trat. tér. [m <sup>3</sup> /h]:	0	68,2	29,2	23,7

TABELA II: Variáveis de entrada relacionadas à composição química [%]

Variáveis de Entrada	Mín.	Máx.	Média	D.P.
Carbono	0,156	0,474	0,3216	0,039
Nióbio	0,000	0,008	0,0026	0,001
Fosforo	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

## II. CONJUNTO DE DADOS

O conjunto de dados, descrito em detalhes em [1, 2], é formado por dois conjuntos diferentes de variáveis de entradas para os modelos de regressão, a saber, seis parâmetros do processo e doze relacionados à composição química do aço. No primeiro grupo, as entradas são a área da seção transversal da barra de aço após laminação, a temperatura de laminação final, a temperatura inicial e final de tratamento térmico, a pressão e o fluxo da água de tratamento térmico. A Tabela I fornece informações sobre intervalo dos parâmetros de laminação a quente e tratamento térmico aplicados. A área da seção transversal da barra de aço está relacionada à força necessária para reduzir suas dimensões no processo de laminação e ao gradiente de temperatura do tratamento térmico. A laminação final, a temperatura inicial e final do tratamento térmico são relacionadas à microestrutura do aço. A pressão da água de tratamento térmico afeta o gradiente de temperatura com o núcleo e a espessura da camada externa de martensita produzida.

As demais variáveis de entrada estão relacionadas à concentração de carbono, nióbio, fósforo, silício, molibdênio, cobre, manganês, vanádio, enxofre, cromo, níquel e estanho, que são elementos que podem influenciar as propriedades mecânicas do produto. A Tabela II mostra os valores mínimo, máximo, média e desvio padrão da composição química do conjunto de dados do aço utilizado nesta pesquisa.

Essas 18 variáveis de entrada são utilizadas pelos modelos de regressão kernel avaliados neste artigo para prever as seguintes variáveis de saída: YS, UTS, razão UTS/YS e PE, apresentadas na Tabela III.

TABELA III: Propriedades mecânicas do vergalhão previstas pelos modelos.

Variáveis Preditas	Mín.	Máx.	Média	D.P.
UTS [MPa]:	421	661	521,7	41,6
YS [MPa]:	596	914	682,8	35,1
Razão UTS/YS:	1,12	1,55	1,314	0,072
PE [%]:	8	20	15,1	1,897

## III. MÉTODOS DE REGRESSÃO KERNEL

A construção de um modelo de regressão  $f(\cdot)$  a partir de dados pressupõe a disponibilidade de um conjunto de  $N$  amostras para o seu treinamento, sendo representado por pares de entradas e saídas  $\{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^N$ , em que  $\mathbf{x}_n \in \mathbb{R}^{N_x}$  são variáveis regressoras e  $\mathbf{y}_n \in \mathbb{R}^{N_y}$  as respectivas saídas observadas. Sem perda de generalidade, considera-se uma única saída,  $N_y = 1$ , para descrever uma abordagem de métodos baseados em kernel, tendo seu modelo expresso por

$$f(\mathbf{x}) = \mathbf{w}^\top \phi(\mathbf{x}) + b, \quad (1)$$

onde  $\phi(\cdot) : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^d$  é um mapeamento não-linear em uma alta dimensão  $d$ ,  $\mathbf{w} \in \mathbb{R}^d$  é o vetor de parâmetros correspondente e  $b \in \mathbb{R}$  é o intercepto.

Assim, no arcabouço do modelo LSSVR [6], seus parâmetros são estimados considerando a minimização da seguinte função de perda:

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

sujeita a  $\mathbf{y}_n = \mathbf{w}^\top \phi(\mathbf{x}_n) + b + e_n$ ,  $n = 1, \dots, N$ , tal que  $e_n = \mathbf{y}_n - f(\mathbf{x}_n)$  é o erro referente a  $n$ -ésima amostra e  $\gamma > 0$  é uma constante de regularização a ser avaliada sobre a ótica do erro de treinamento e suavidade da função de regressão.

Esse problema de otimização possui sua formulação no espaço dual escrita como o seguinte lagrangiano:

$$\begin{aligned} J_d(\mathbf{w}, b, \mathbf{e}, \boldsymbol{\alpha}) &= J_p(\mathbf{w}, \mathbf{e}) \\ &\quad - \sum_{n=1}^N \alpha_n [\mathbf{w}^\top \phi(\mathbf{x}_n) + b + e_n - \mathbf{y}_n] \\ &= \frac{1}{2} \mathbf{w}^\top \mathbf{w} + \frac{\gamma}{2} \sum_{n=1}^N e_n^2 \\ &\quad - \sum_{n=1}^N \alpha_n [\mathbf{w}^\top \phi(\mathbf{x}_n) + b + e_n - \mathbf{y}_n], \end{aligned} \quad (3)$$

no qual  $\boldsymbol{\alpha} \in \mathbb{R}^N$  representa o vetor dos multiplicadores de Lagrange. A solução para (3) pode ser obtida pelas condições de otimalidade de Karush-Kuhn-Tucker, reescrevendo o problema em um sistema linear:

$$\begin{bmatrix} 0 & \mathbf{1}_N^\top \\ \mathbf{1}_N & \boldsymbol{\Omega} + \gamma^{-1} \mathbf{I}_N \end{bmatrix} \begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{y} \end{bmatrix}, \quad (4)$$

em que  $\mathbf{y} = [y_1, \dots, y_N]^\top$ ,  $\mathbf{1}_N \in \mathbb{R}^N$  é um vetor de 1's,  $\mathbf{I}_N \in \mathbb{R}^{N \times N}$  é uma matriz identidade e  $\boldsymbol{\Omega} \in \mathbb{R}^{N \times N}$  é a matriz de kernel com os elementos  $\Omega_{i,j} = \phi^\top(\mathbf{x}_i) \phi(\mathbf{x}_j) \forall i, j \in$

$\{1, \dots, N\}$ . Uma possibilidade para emular este produto interno de um mapeamento em alta dimensão dá-se pela aplicação do truque de kernel, em que se usa uma função que satisfaz as condições de Mercer, suficientes para garantir que seja simétrica e positiva definida, tal como a função gaussiana escolhida para os experimentos deste trabalho, dada por

$$\Omega_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j) = \exp \left\{ -\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma^2} \right\}, \quad (5)$$

onde  $k(\cdot, \cdot)$  denota a função de kernel,  $\|\cdot\|_2$  a norma euclidiana quadrática e  $\sigma \in \mathbb{R}$  seu hiperparâmetro. Portanto, o modelo de regressão é expresso agora como

$$f(\mathbf{x}) = \sum_{n=1}^N \alpha_n k(\mathbf{x}_n, \mathbf{x}) + b. \quad (6)$$

A técnica regressão de cumeeira kernel (*kernel ridge regression*, KRR) [7] pode ser entendida como uma versão simplificada do LSSVR, que consiste operacionalmente na desconsideração do viés  $b$  em (4) e da restrição  $\sum_{n=1}^N \alpha_n = 0$ . Dessa maneira, a solução do sistema linear é dada por

$$\boldsymbol{\alpha} = (\boldsymbol{\Omega} + \gamma \mathbf{I})^{-1} \mathbf{y}. \quad (7)$$

Diferente do modelo SVR [8], que produz uma solução esparsa no que se refere à construção do modelo usando alguns poucos vetores do conjunto de treinamento, os chamados *vetores suporte* (VS), os modelos LSSVR e KRR frequentemente utilizam todos os dados disponíveis. Assim, técnicas de redução do número de VS são empregadas com o intuito de simplificar a estrutura interna da LSSVR/KRR [9]. Na seção seguinte, são apresentadas algumas dessas técnicas para comparação de desempenho.

#### IV. MÉTODOS DE REDUÇÃO DE VETORES SUPORTE

São escolhidas quatro formas de esparsificação para os métodos de kernel testados, sendo a primeira uma simples *seleção aleatória* de vetores suporte dentre os dados de treinamento. Desse modo, pode-se aferir a qualidade das demais abordagens tendo esse desempenho de referência.

Em Murta et al. [2], aplica-se a *poda pela magnitude de  $|\alpha|$*  (valores absolutos dos coeficientes de Lagrange). Ou seja, encontra-se o vetor  $\boldsymbol{\alpha}$ , considerando todos os dados de treinamento, e excluem-se os VS referentes aos menores valores de  $|\alpha|$ . Em seguida, o modelo é recalculado com os VS restantes.

Na terceira abordagem, usa-se o algoritmo *k-means* [10] que é uma técnica quantização vetorial comumente usada para redução de volume de dados [9]. Nesse contexto, o objetivo é representar um conjunto de dados original por um conjunto reduzido formado pelos centroides de cada agrupamento encontrado sem que haja perda significativa de informação. Logo, a escolha dos VS se dá pela dissimilaridade, pois cada centroide substitui os dados similares a ele na construção do modelo esparsa.

Por essa perspectiva, o algoritmo *k-means* trabalha com o conjunto de dados com entradas e saídas concatenadas, ou seja,

$\{\mathbf{x}_n^* := [\mathbf{x}_n; \mathbf{y}_n]\}_{n=1}^N$ , separando-o em  $K$  partições e seguindo uma função objetivo especificada. Utiliza-se aqui a distância euclidiana quadrática, definindo o problema como:

$$\min \sum_{i=1}^K \sum_{\forall \mathbf{x}_n^* \in P_i} \|\mathbf{x}_n^* - \mathbf{c}_i\|_2^2, \quad (8)$$

em que  $\mathbf{c}_i \in \mathbb{R}^{N_x + N_y}$  é o vetor centroide da partição  $P_i$ . Para isso, o algoritmo segue os seguintes passos:

- 1) Inicie os centroides  $\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_K$  aleatoriamente a partir dos vetores do conjunto  $\{\mathbf{x}_n^*\}_{n=1}^N$ .
- 2) Atribua cada vetor  $\mathbf{x}_n^*$  ao centroide  $\mathbf{c}_i$  mais próximo, formando a partição  $P_i$ .
- 3) Atualize cada centroide  $\mathbf{c}_i$ , calculando sua nova posição a partir da média aritmética do conjunto de vetores de partição  $P_i$ .
- 4) Repita os passos 2 e 3 até que não ocorram alterações nas posições dos centroides.

Entretanto, o algoritmo *k-means* pode apresentar sensibilidade à presença de *outliers*. Dito isso, o segundo método de QV aplicado é o algoritmo *k-medoids* [11], que é uma abordagem robusta quando os centroides computados pela média dos pontos não é uma representação adequada dos *clusters*. Nessa técnica, o centroide  $\mathbf{c}_i$  é substituído pelo medoide  $\mathbf{m}_i$  em (8), tal que  $\mathbf{m}_i$  é obrigatoriamente um vetor do conjunto  $\{\mathbf{x}_n^*\}_{n=1}^N$ . Deste modo, o algoritmo *k-medoids* apresenta os seguintes passos:

- 1) Inicie os medoides  $\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_K$  aleatoriamente a partir dos vetores do conjunto  $\{\mathbf{x}_n^*\}_{n=1}^N$ .
- 2) Atribua cada vetor  $\mathbf{x}_n^*$  ao medoide  $\mathbf{m}_i$  mais próximo, formando a partição  $P_i$ .
- 3) Para cada vetor pertencente ao *cluster*  $P_i$ , calcule o somatório das distâncias entre ele e os demais elementos da partição.
- 4) Atualize cada medoide  $\mathbf{m}_i$  pelo vetor da respectiva partição  $P_i$  que apresenta o menor somatório das distâncias no passo 3.
- 5) Repita os passos 2 a 4 até que não ocorram alterações nos medoides.

#### V. OTIMIZAÇÃO DE HIPERPARÂMETROS

A sintonia dos parâmetros é uma etapa crítica na construção do modelo de regressão. Em Murta et al. [2], utiliza-se a *Toolbox LS-SVMLab* (versão 1.8) [12] para a implementação do modelo LSSVR. Na versão atual dessa *toolbox*, encontra-se uma otimização conjunta dos hiperparâmetros entre o método Recozimento Simulado Acoplado (*Coupled Simulated Annealing*, CSA) e o método *simplex*. Com isso, o algoritmo CSA é responsável pela exploração no espaço de busca e o método *simplex* pelo refinamento da solução, diminuindo o número e avaliações da função objetivo e melhorando a qualidade da solução quando comparados separadamente esses algoritmos. Nos experimentos realizados neste trabalho, opta-se por combinar a metaheurística otimização por exame de partícula (*particle swarm optimization*, PSO) [13] com o *simplex* de Nelder-Mead [14, 15].

A metaheurística PSO é um algoritmo de otimização metaheurística 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 a procura de soluções ótimas. Assim, a posição de cada partícula no enxame representa uma solução candidata para o problema de otimização tratado e são atribuídas 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.

$$\mathbf{p}_i(t+1) = \mathbf{p}_i(t) + \mathbf{v}_i(t+1). \quad (9)$$

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

O desempenho de 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

$$\mathbf{p}_i^*(t+1) = \begin{cases} \mathbf{p}_i^*(t) & \text{if } f(\mathbf{p}_i(t+1)) \geq f(\mathbf{p}_i^*(t)) \\ \mathbf{p}_i(t+1) & \text{if } f(\mathbf{p}_i(t+1)) < f(\mathbf{p}_i^*(t)). \end{cases} \quad (10)$$

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

$$f_o(\mathbf{g}(t)) = \min \left\{ f_o(\mathbf{p}_0^*(t)), \dots, f(\mathbf{p}_{N_p}^*(t)) \right\}, \quad (11)$$

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

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

$$\mathbf{v}_i(t+1) = \mathbf{v}_i(t) + c_1\psi_1(t) \odot [\mathbf{p}_i^*(t) - \mathbf{p}_i(t)] + c_2\psi_2(t) \odot [\mathbf{g}(t) - \mathbf{p}_i(t)], \quad (12)$$

em que  $c_1, c_2 \in \mathbb{R}$  são constantes positivas de aceleração usadas para dimensionar a contribuição dos componentes cognitivo e social, respectivamente. Os vetores  $\psi_1(t), \psi_2(t) \in \mathbb{R}^{N_p}$  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 nesse 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 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. [15].

TABELA IV: Comparação de desempenho entre os modelos LSSVR e KRR em termos de  $R^2$  médio.

Modelo	UTS	YS	UTS/YS	PE
LSSVR	0,8387	0,7773	0,8772	0,7118
KRR	0,8390	0,7775	0,8766	0,7119

## VI. RESULTADOS E DISCUSSÕES

Na presente seção, reportam-se e discutem-se os resultados obtidos ao aplicar os modelos de regressão kernel e as técnicas de quantização vetorial previamente descritos para a predição de propriedades mecânicas de vergalhões de aço. Os experimentos são realizados no ambiente programação MATLAB<sup>®</sup> e utilizam-se algumas de suas funções nativas, que são indicadas no decorrer do texto.

O conjunto de dados possui 1300 instâncias, tendo seus pares de entradas e saídas embaralhados 100 vezes e divididos entre 80% (1040 amostras) para treinamento e 20% (260 amostras) para teste dos modelos. Dessa maneira, os algoritmos dispõem das mesmas 100 divisões em todos os experimentos. Além disso, a normalização *z-score* é aplicada com as estatísticas do conjunto de treinamento e as saídas preditas dos modelos são desnormalizadas para avaliação de desempenho.

Para cada experimento, os hiperparâmetros dos modelos de regressão kernel, isto é, a constante de regularização  $\gamma$  e a largura  $\sigma$  da função gaussiana, são otimizados por uma abordagem conjunta dos algoritmos PSO e *simplex*, em que se utiliza a validação cruzada com *k-fold* de 10 grupos para minimizar o erro médio quadrático das estimações dos dados de treinamento.

Para essas otimizações, elabora-se o algoritmo PSO global, como descrito na Seção V, com um enxame composto por cinco partículas de duas posições. O intervalo de busca para os dois hiperparâmetros está contido em  $[e^{-7.5}, e^{7.5}]$ . Assim, são calculadas cinco iterações para o enxame e o melhor resultado obtido é utilizado como ponto inicial do método *simplex*. Para isso, emprega-se a função `@fminsearch` do MATLAB<sup>®</sup> para o refinamento da solução encontrada pelo PSO, sendo limitada a 100 iterações em caso de não convergência da resposta.

Tendo em vista a comparação de desempenho com trabalhos de Murta et al. [1, 2], a figura de mérito escolhida é o coeficiente de determinação ( $R^2$ ), dado por

$$R^2 = 1 - \frac{\sum_{i=1}^{N_t} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{N_t} (y_i - \bar{y})^2}, \quad (13)$$

em que  $N_t$  é o número de amostras de treinamento,  $y_i$  é a  $i$ -ésima saída observada,  $\bar{y}$  é a média aritmética dos valores de todas as saídas observadas do conjunto de teste e  $\hat{y}_i = f(\mathbf{x}_i)$  é a  $i$ -ésima saída predita.

A primeira avaliação a ser feita é a comparação entre os modelos LSSVR e KRR. Na Tabela IV, pode-se perceber pelo  $R^2$  médio das 100 realizações que não há diferença significativa de desempenho. Assim sendo, opta-se pelo modelo mais simples que é o KRR para o estudo sobre a redução dos vetores suporte.

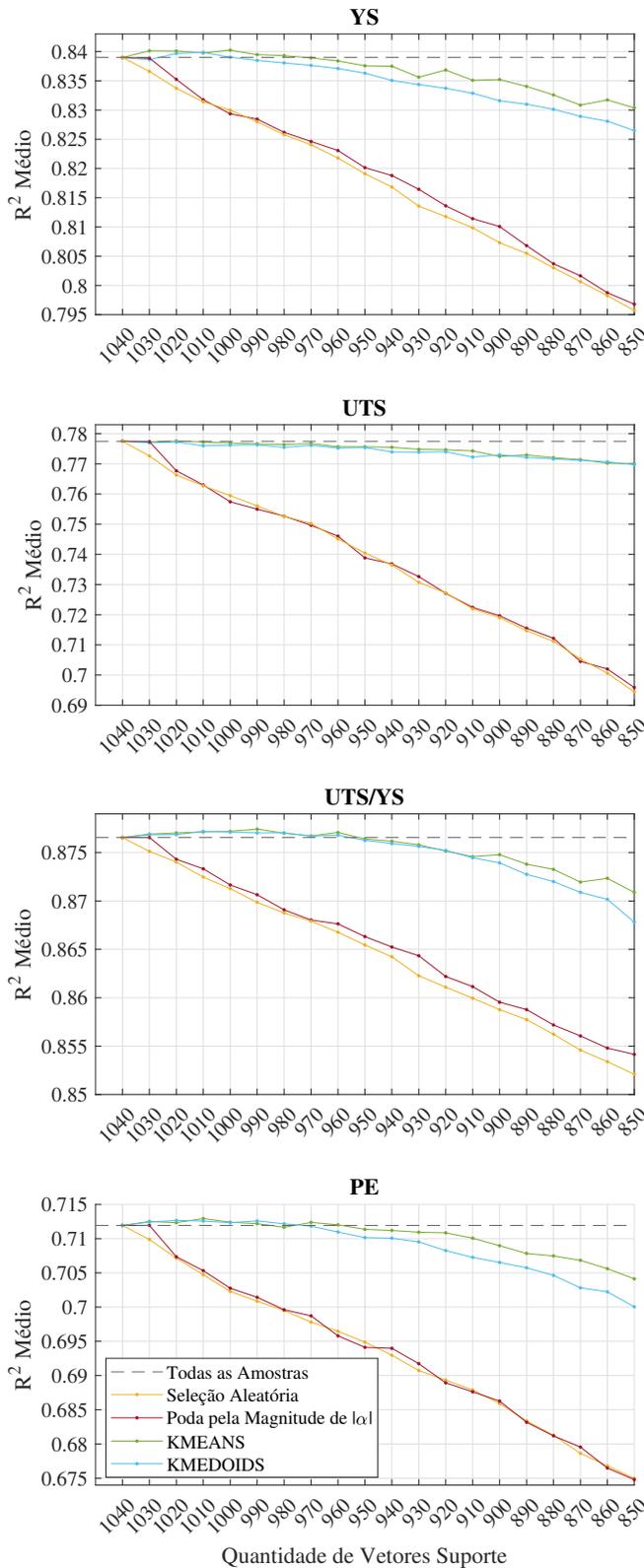


Fig. 1: Desempenho do modelo KRR com o emprego de técnicas de redução do conjunto de vetores suporte.

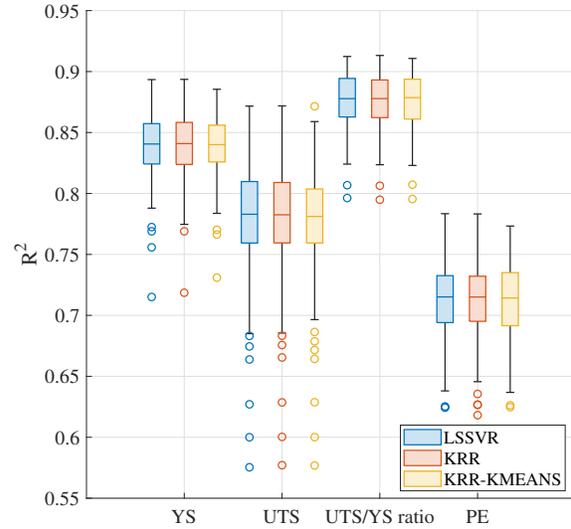


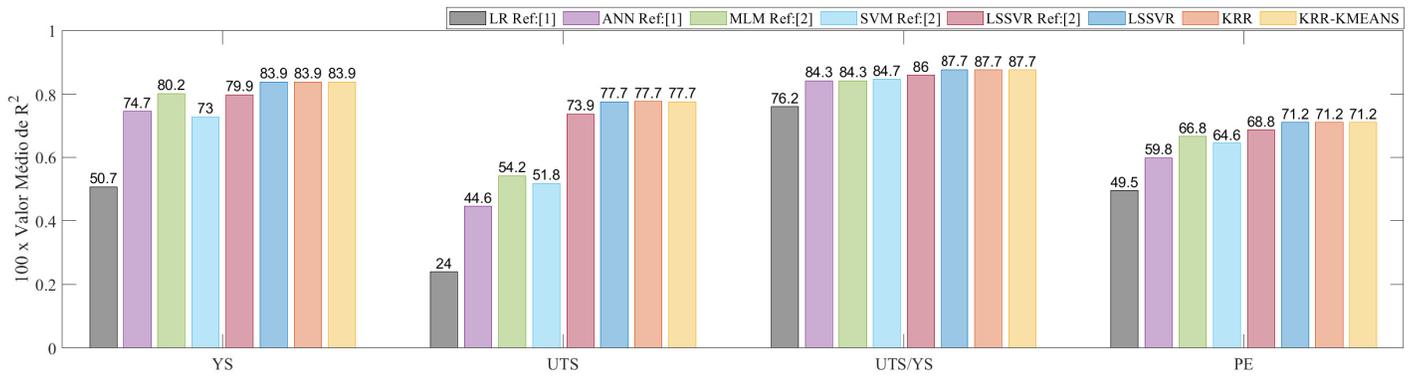
Fig. 2: Boxplot dos resultados das 100 realizações para os algoritmos LSSVR, KRR e KRR-KMEANS.

Por conseguinte, pode-se observar na Figura 1 a linha tracejada na cor preta que representa o valor de  $R^2$  médio obtido na estimação com todos os dados disponíveis do treinamento para o modelo KRR. Na cor amarela, tem-se a linha de referência do desempenho da *seleção aleatória*, que apresenta uma piora significativa quanto menos VS são utilizados no modelo. Já a técnica empregada por Murta et al. [2], a *poda pela magnitude de  $|\alpha|$* , só se apresenta útil na exclusão dos primeiros 10 VS, tendo a trajetória similar à *seleção aleatória* no restante do gráfico, sendo desenhada pela linha vermelha.

Para os métodos de quantização vetorial, são empregadas as funções do MATLAB<sup>®</sup>: `@kmeans` e `@kmedoids`, variando o número de centroides (ou medoides). Ainda considerando os gráficos da Figura 1, os métodos de QV se apresentam adequados à tarefa de esparsificação do modelo até aproximadamente a quantidade de 970 VS para as quatro saídas avaliadas. Destaca-se o método *k-means*, denotado pela linha verde, que apresenta melhor desempenho nas saídas YS, UTS/YS e PE e, além disso, possui um custo computacional menor que o algoritmo *k-medoids*, apresentado pela linha azul.

Ilustra-se na Figura 2 os *boxplots* dos resultados das 100 predições das quatro saídas estudadas para LVSSR e KRR com todas as amostras de treinamento disponíveis e para a metodologia proposta KRR-KMEANS com 970 VS. Afere-se, então, a mesma eficácia desses modelos. Entretanto, o KRR-KMEANS é mais eficiente no que diz respeito à quantidade de VS a ser armazenada no modelo.

Por fim, é apresentado na Figura 3 um comparativo de desempenho dos algoritmos testados neste artigo e em Murta et al. [1, 2]. É importante ressaltar que a implementação do modelo LSSVR realizada neste estudo demonstra um desempenho significativamente superior em relação à versão

Fig. 3: Comparação de desempenho dos modelos de regressão testados em termos de  $R^2$  médio.TABELA V: Comparação de desempenho dos modelos de regressão testados em termos de média e de desvio padrão de  $R^2$ .

Modelos	UTS		YS		UTS/YS		PE	
	Média	D.P.	Média	D.P.	Média	D.P.	Média	D.P.
LR [1]	0,5072	0,0515	0,2398	0,0440	0,7618	0,0289	0,4950	0,0424
ANN [1]	0,7472	0,0540	0,4455	0,0971	0,8428	0,0289	0,5978	0,0636
MLM [2]	0,8022	0,0368	0,5416	0,2165	0,8426	0,0317	0,6679	0,0595
SVM [2]	0,7296	0,0338	0,5180	0,0588	0,8469	0,0271	0,6456	0,0383
KRR-KMEANS	0,8384	0,0254	0,7756	0,0502	0,8771	0,0228	0,7120	0,0325

proposta por Murta et al. [2]. Essa melhoria se dá pelo aumento de 20% para 100% dos dados de treinamento utilizados durante a otimização dos hiperparâmetros. Ademais, a partir das informações de média e de desvio padrão de  $R^2$  presentes na Tabela V, pode-se destacar que o modelo KRR-KMEANS, esparsos e mais simples que o modelo LSSVR, se torna o estado da arte nesta tarefa, apresentando avanços promissores em relação às predições de propriedades mecânicas no conjunto de dados em análise e enfatizam a relevância do presente estudo.

## VII. CONCLUSÕES

Neste artigo, relatam-se experimentos para avaliar possíveis estratégias para aperfeiçoar os resultados dos modelos de regressão kernel relatados trabalhos anteriores, dada a importância da aplicação para a metalurgia/siderurgia. Constatase a importância da otimização dos hiperparâmetros durante o treinamento ao melhorar o desempenho dos modelos. Além disso, apresenta com êxito uma técnica de esparsificação via quantização vetorial com o algoritmo *k-means* em conjunto com uma simplificação do modelo LSSVR, resultando no modelo KRR-KMEANS.

A melhoria da predição dos modelos encontrada reforça que a composição química e os parâmetros de tratamento térmico podem ser utilizados para prever as quatro principais propriedades mecânicas finais do aço: YS, UTS, relação UTS/YS e PE, motivando assim a aplicação de novas técnicas nesse conjunto de dados para superar os resultados apresentados.

Os esforços atuais em nossa pesquisa estão em buscar novas estratégias de esparsificação dos modelos de regressão kernel. Além de estudos de outras técnicas que utilizam projeções aleatórias [16] e modelos locais [17].

## REFERÊNCIAS

- [1] R. H. F. Murta, F. D. Braga, P. P. N. Maia, O. B. F. Diogenes, and E. P. de Moura, "Mathematical modelling for predicting mechanical properties in rebar manufacturing," *Ironmaking & Steelmaking*, vol. 48, no. 2, pp. 161–169, Apr. 2020. [Online]. Available: <https://doi.org/10.1080/03019233.2020.1749357>
- [2] R. H. F. Murta, E. P. de Moura, and G. A. Barreto, "Mechanical properties prediction in rebar using kernel-based regression models," *Ironmaking & Steelmaking*, vol. 49, no. 10, pp. 1011–1020, Jun. 2022. [Online]. Available: <https://doi.org/10.1080/03019233.2022.2075691>
- [3] J. T. Black and R. A. Kohser, *DeGarmo's materials and processes in manufacturing*, 2019.
- [4] Q. Xie, M. Suvarna, J. Li, X. Zhu, J. Cai, and X. Wang, "Online prediction of mechanical properties of hot rolled steel plate using machine learning," *Materials & Design*, vol. 197, p. 109201, Jan. 2021. [Online]. Available: <https://doi.org/10.1016/j.matdes.2020.109201>
- [5] S. Lalam, P. K. Tiwari, S. Sahoo, and A. K. Dalal, "Online prediction and monitoring of mechanical properties of industrial galvanised steel coils using neural networks," *Ironmaking & Steelmaking*, vol. 46, no. 1, pp. 89–96, Jul. 2017. [Online]. Available: <https://doi.org/10.1080/03019233.2017.1342424>
- [6] J. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Processing Letters*, vol. 9, no. 3, pp. 293–300, 1999. [Online]. Available: <https://doi.org/10.1023/a:1018628609742>
- [7] C. Saunders, A. Gammernan, and V. Vovk, "Ridge regression learning algorithm in dual variables," 1998.
- [8] H. Drucker, C. J. C. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," in *Advances in Neural Information Processing Systems*, M. Mozer, M. Jordan, and T. Petsche, Eds., vol. 9. MIT Press, 1996. [Online]. Available: <https://alex.smola.org/papers/1997/DruBurSmoKauetal97.pdf>
- [9] A. R. Neto and G. A. Barreto, "Opposite maps: Vector quantization algorithms for building reduced-set SVM and LSSVM classifiers," *Neural Processing Letters*, vol. 37, no. 1, pp. 3–19, Dec. 2012. [Online]. Available: <https://doi.org/10.1007/s11063-012-9265-6>
- [10] J. MacQueen, "Classification and analysis of multivariate observations," in *5th Berkeley Symp. Math. Statist. Probability*. University of California Los Angeles LA USA, 1967, pp. 281–297.
- [11] L. Kaufman and P. J. Rousseeuw, *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons, 2009.

- [12] K. De Brabanter, P. Karsmakers, F. Ojeda, C. Alzate, J. De Brabanter, K. Pelckmans, B. De Moor, J. Vandewalle, and J. A. Suykens, *LS-SVMLab toolbox user's guide: version 1.8*. Katholieke Universiteit Leuven, 2011.
- [13] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*. IEEE. [Online]. Available: <https://doi.org/10.1109/icnn.1995.488968>
- [14] J. A. Nelder and R. Mead, "A simplex method for function minimization," *The Computer Journal*, vol. 7, no. 4, pp. 308–313, Jan. 1965. [Online]. Available: <https://doi.org/10.1093/comjnl/7.4.308>
- [15] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence properties of the nelder–mead simplex method in low dimensions," *SIAM Journal on Optimization*, vol. 9, no. 1, pp. 112–147, Jan. 1998. [Online]. Available: <https://doi.org/10.1137/s1052623496303470>
- [16] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Proceedings of the 20th International Conference on Neural Information Processing Systems*, ser. NIPS'07. Red Hook, NY, USA: Curran Associates Inc., 2007, p. 1177–1184.
- [17] R. B. Drummond, R. F. Albuquerque, G. A. Barreto, and A. H. Souza, "Pattern classification based on regional models," *Applied Soft Computing*, vol. 129, p. 109592, Nov. 2022. [Online]. Available: <https://doi.org/10.1016/j.asoc.2022.109592>