Regressão por Métodos Kernel na Previsão de Longo Prazo de Precipitação Pluviométrica Mensal

Allan Kelvin M. Sales
Departamento de Telemática
Instituto Federal do Ceará (IFCE)
Campus Fortaleza
kelvin@ifce.edu.br

Guilherme A. Barreto
Pós-Grad. Eng. de Teleinformática (PPGETI)
Universidade Federal do Ceará (UFC)
Campus do PICI, Centro de Tecnologia
gbarreto@ufc.br

Francisco A. Souza Filho
Pós-Grad. em Eng. Civil (PPGEC)
Universidade Federal do Ceará (UFC)
Campus do PICI, Centro de Tecnologia
assis@ufc.br

***Resumo*—This article presents a systematic comparative analysis of sparse kernel adaptive filters (KAFs), specifically, KRLS-ALD, KRLS-T, SKRLS, QKLMS, and KNLMS, against traditional batch machine learning models (LSSVR, SVR-ν) for multi-step ahead monthly rainfall forecasting, using a univariate time series from Fortaleza, Brazil. As an original methodological contribution, we propose the IGNOS (Nonlinear Generalization Index for Relative Bias-Variance Tradeoff) metric to guide model selection by diagnosing overfitting and underfitting by quantifying the relationship between training and validation errors. The results demonstrate that while SVR-ν achieves a competitive test RMSE, KAFs exhibit significantly better computational efficiency and temporal robustness. In particular, the KRLS-ALD and KRLS-T algorithms better capture the non-stationary dynamics and extreme events that batch models tend to smooth out, establishing KAFs as a more practical and robust solution for hydrometeorological forecasting in challenging climates.**

***Index Terms*—Long-term prediction, rainfall forecasting, Kernel methods, Sparse models, Machine learning**

I. Introdução

A previsão acurada de precipitação de chuvas em longo prazo é fundamental para o planejamento hídrico e agrícola. Entretanto, séries temporais pluviométricas frequentemente exibem não estacionariedade e variabilidade climática que exigem modelos capazes de lidar com padrões complexos e não lineares [1]. Métodos como modelos numéricos de previsão temporal (NWP) e técnicas estatísticas clássicas (ARIMA, regressão linear, etc) são amplamente utilizados, mas sua eficácia é limitada em capturar eventos extremos e transições sazonais abruptas [2].

Técnicas de Aprendizado de Máquinas (AM), como *Support Vector Regression* (SVR), *Least Squares SVR* (LSSVR) e SVR-ν, avançam nessa direção, mas enfrentam obstáculos práticos, já que não escalam bem para grandes volumes de dados e demandam custos computacionais elevados para treinamento e ajuste de hiperparâmetros [3].

Diante dessas limitações, *Kernel Adaptive Filters* (KAFs) emergem como alternativas tecnicamente robustas. Algoritmos como o *Kernel Recursive Least Squares* (KRLS) oferecem uma abordagem *online* ao processar dados sequencialmente e atualizar o modelo a cada nova observação.

Pesquisas recentes destacam vantagens dos KAFs a métodos clássicos de AM em tarefas de identificação de sistemas e previsão temporal [4]. No entanto, apesar do potencial dos KAFs e do uso de métodos SVR na área [3], a literatura especializada carece de estudos sistemáticos que comparem KAFs e as técnicas de AM em previsões hidrometeorológicas de longo prazo [5].

Isto posto, este trabalho preenche essa lacuna ao propor três contribuições centrais. Primeiramente, provavelmente conduz a primeira avaliação empírica abrangente de Algoritmos KAFs esparsos: as variantes do KRLS: KRLS-ALD, KRLS-T e SKRLS, os algoritmos *quantized kernel least mean square* (QKLMS) e *kernel normalized least mean square* (KNLMS), comparando-os com LSSVR e SVR-ν por meio de séries univariadas.

Estes algoritmos integram adaptação *online*, mecanismos de esparsificação e complexidade computacional reduzida, características essenciais para séries não estacionárias e ambientes ruidosos [6], como é o caso das séries históricas de precipitação de chuvas.

Como inovação metodológica, o presente trabalho introduz o índice IGNOS (Índice de Generalização Não Linear de Otimização Relativa de Sobreajuste), métrica projetada para facilitar a seleção de modelos e quantificar a relação entre erros de treinamento e validação, por meio do diagnóstico de *overfitting* ou *underfitting*.

Por fim, realiza um estudo de caso com dados históricos consolidados pela Fundação Cearense de Meteorologia e Recursos Hídricos (FUNCEME) da região metropolitana de Fortaleza, escolhida por representar climas semiáridos com alta variabilidade pluviométrica.

O restante do artigo se estrutura da seguinte forma: a Seção II descreve os fundamentos teóricos dos KAFs; a Seção III detalha a metodologia empregada; a Seção IV apresenta e discute os resultados; e, finalmente, a Seção V conclui o trabalho e aponta direções futuras.

II. Modelos Esparsos Adaptativos para Regressão Kernel

Modelos adaptativos esparsos para regressão kernel, também chamados de *Kernel Adaptive Filters* (KAFs), representam uma categoria de algoritmos de processamento de sinais com aprendizado *online*, que combinam a capacidade de modelagem não linear dos métodos baseados em funções kernel com a adaptação da estrutura e dos parâmetros de filtros adaptativos [7].

Os KAFs operam em espaços de Hilbert de kernel reprodutor (RKHS), em que um mapeamento não linear, $\phi : \mathcal{X} \rightarrow \mathcal{H}$, leva os dados de entrada a um espaço de características, $\mathcal{H}$, de dimensionalidade alta ou infinita. Isso torna o cálculo explícito de $\phi(\mathbf{x})$ computacionalmente intratável. Produtos escalares, contudo, nesse mesmo espaço podem ser computados implicitamente via uma função kernel positiva definida. Essa equivalência, conhecida como *truque de kernel*, é a base dos métodos de kernel [8].

Basicamente, dada uma sequência de dados $\{(\mathbf{x}_n, y_n)\}_{n=1}^{N}$, onde $\mathbf{x}_n \in \mathbb{R}^d$ e $y_n \in \mathbb{R}$, um KAF busca estimar uma função $f(\cdot)$ não linear na forma paramétrica:

$$f(\mathbf{x}) = \langle \phi(\mathbf{x}), \mathbf{w} \rangle = \mathbf{w}^\top \cdot \phi(\mathbf{x}) \tag{1}$$

em que $\phi(\cdot)$ é uma projeção para o RKHS, $\mathbf{w}$ é o vetor de parâmetros, e $\langle \cdot, \cdot \rangle$ denota o produto interno no espaço de características.

O objetivo do KAF é minimizar uma função de perda, $\boldsymbol{\mathcal{L}}$, à medida que novos dados são coletados, normalmente definida como:

$$\boldsymbol{\mathcal{L}}(\mathbf{w}) = \min_{\mathbf{w}} \sum_{n=1}^{N} \parallel \mathbf{y}_n - \mathbf{w}_n^\top \cdot \phi(\mathbf{x}_n) \parallel^2 \tag{2}$$

Se um kernel de Mercer for usado [9], o que permite a substituição: $k\left(\mathbf{x}_i, \mathbf{x}_j\right) = \langle \phi\left(\mathbf{x}_i\right), \phi\left(\mathbf{x}_j\right) \rangle$, o Teorema de Representação [10] garante que o vetor de pesos $\mathbf{w}$ pode ser expresso como uma combinação linear dos dados de treinamento mapeados:

$$\mathbf{w} = \sum_{n=1}^{N} \alpha_n \cdot \phi\left(\mathbf{x}_n\right) = \boldsymbol{\Phi}^\top \cdot \boldsymbol{\alpha} \tag{3}$$

Isso leva à forma de predição característica baseada em kernel ao substituir (3) em (1) [11]:

$$\begin{aligned} f(\mathbf{x}) &= \sum_{n=1}^{N} \alpha_n \cdot \langle \phi(\mathbf{x}) \cdot \phi\left(\mathbf{x_n}\right) \rangle \\ &= \sum_{n=1}^{N} \alpha_n k\left(\mathbf{x}, \mathbf{x}_n\right) = \mathbf{K} \cdot \boldsymbol{\alpha} \end{aligned} \tag{4}$$

em que $\mathbf{K}$ é a matriz de kernel; $k\left(\cdot, \cdot\right)$ é a função de kernel e α_n é o coeficiente associado a cada função de kernel. Neste artigo, a função de kernel gaussiana usada é dada pela equação:

$$k\left(\mathbf{x}, \mathbf{x}_n\right) = \exp\left(\gamma \cdot \left\|\mathbf{x} - \mathbf{x}_n\right\|^2\right) \tag{5}$$

em que $\gamma = -1/(2\sigma^2)$ e σ é um fator de escala chamado de largura de banda do kernel gaussiano.

A capacidade de atualizar o modelo a cada nova amostra $(\mathbf{x}_n, \mathbf{y}_n)$ coletada torna os KAFs inerentemente adequados para processamento *online* de séries temporais em ambientes não estacionários [12], um cenário comum na previsão de precipitação de chuvas.

Há dois algoritmos fundamentais: o KRLS e o *kernel least mean square* (KLMS), que estendem os filtros RLS e o LMS lineares ao RKHS, respectivamente. Contudo, existe uma limitação intrínseca a estas técnicas: a quantidade de vetores que forma a base de representação da função $f(\cdot)$ de kernel cresce linearmente com o número N de amostras processadas [12]. Isso leva rapidamente a uma complexidade computacional e de memória proibitivas para grande volume de dados [8].

Para viabilizar a aplicação dos KAFs em cenários reais e de longo prazo, foram desenvolvidas diversas estratégias de esparsidade para modelos preditivos baseados em kernel [12]–[18]. O objetivo comum é selecionar durante o processamento *online* um subconjunto reduzido de tamanho m ($m \ll N$) das amostras de entrada (instâncias ou centros), denominado dicionário.

O dicionário resultante deve ser capaz de representar a função de regressão com boa acurácia e precisão, o que é garantido mediante algum critério de seleção específico. Esta abordagem permite manter a complexidade computacional em níveis gerenciáveis, frequentemente $\mathrm{O}(m^2)$, mesmo com o crescimento de N.

Na sequência, cinco algoritmos KAF do estado da arte incluídos neste artigo são brevemente descritos.

1) ***KRLS-ALD****:* O algoritmo KRLS com dependência linear aproximada (KRLS-ALD) [13] controla a esparsidade do modelo por meio do critério ALD (*approximate linear dependency*). Este algoritmo avalia, para cada nova amostra $\mathbf{x}_n$, se sua imagem $\phi(\mathbf{x}_n)$ no RKHS pode ser adequadamente aproximada por uma combinação linear das imagens $\{\phi(\mathbf{x}_j)\}_{j=1}^{m}$ dos m vetores presentes no dicionário atual. A amostra $\mathbf{x}_n$ é adicionada ao dicionário se o erro quadrático mínimo dessa aproximação linear exceder o limiar ν. Matematicamente, a condição para adição é:

$$\delta_n = \min_{\alpha} \left\| \phi(\mathbf{x}_n) - \sum_{j=1}^{m} \alpha_j \phi(\mathbf{x}_j) \right\|^2 \geq \nu, \tag{6}$$

tal que $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_m]^T$ é o vetor de coeficientes da combinação linear e δ_n é o erro de aproximação linear no espaço de características. Se $\delta_n > \nu$, $\mathbf{x}_n$ é adicionado ao dicionário. Caso contrário, a amostra é descartada, preservando a esparsidade. O limiar ν controla o compromisso entre a acurácia da representação e o tamanho do dicionário.

2) ***KRLS-T*** *(KRLS - Tracker):* O algoritmo KRLS-T [12] estende a teoria do KRLS padrão sob uma perspectiva bayesiana ao integrar mecanismos de esquecimento para ambientes não estacionários. Além de gerar previsões,

Tabela I
CRITÉRIOS DE ESPARSIFICAÇÃO, ν, DOS ALGORITMOS ANALISADOS E SUAS EXPRESSÕES CORRESPONDENTES.

Algoritmo	**Critério**	**Expressão**	**Adotado em**
KRLS-ALD	ALD	$\delta_n = \min_\alpha \left\|\phi(\mathbf{x}_n) - \sum_{j=1}^{m} \alpha_j \phi(\mathbf{x}_j)\right\|^2 \geq \nu$	[13]
KRLS-T	Relevância[1]	$\gamma_n^2 = k(\mathbf{x}_n, \mathbf{x}_n) - \mathbf{k}_n^\top \mathbf{Q}_{n-1} \mathbf{k}_n \geq \nu$	[12]
SKRLS	Significância	$\Delta\mathcal{L}_n = \frac{1}{2} \Delta\boldsymbol{\alpha}_n^\top \mathbf{H}_n \Delta\boldsymbol{\alpha}_n \geq \nu$	[15]
KNLMS	Coerência	$\mu = \max_{1 \leq j \leq m} \lvert k(\mathbf{x}_n, \mathbf{x}_j)\rvert < \nu$	[14]
QKLMS	Quantização	$d_{min} = \min_{1 \leq j \leq m} \lVert \mathbf{x}_n - \mathbf{c}_j \rVert > \nu$	[18]

o KRLS-T fornece intervalos de confiança usados para adaptar dinamicamente os parâmetros do modelo. O algoritmo emprega três mecanismos principais: (1) um fator de esquecimento, $\lambda \in (0, 1]$, que controla a taxa de adaptação ao atenuar a influência de dados antigos; (2) a variância do ruído, σ_n^2, que regula a suavidade do modelo ao quantificar a incerteza das observações; e (3) um limiar de relevância (não redundância), ν, que determina se uma nova amostra, $\mathbf{x}_n$, é adicionada ao dicionário. A decisão é baseada no erro (incerteza) de projeção normalizado, definido como:

$$\gamma_n^2 = k(\mathbf{x}_n, \mathbf{x}_n) - \mathbf{k}_n^\top \mathbf{Q}_{n-1} \mathbf{k}_n \geq \nu \tag{7}$$

em que: $\mathbf{Q}_{n-1} = \mathbf{K}_{n-1}^{-1}$. Para gerenciar a complexidade computacional e manter o tamanho do dicionário $m = M$ fixo, o KRLS-T remove a amostra menos importante quando M atinge seu limite máximo. Nesse momento, a relevância é calculada via baixo impacto na matriz de covariância posterior, (8), e a base que possuir o menor ϵ_s é descartada [12].

$$\epsilon_s = \left(\frac{[\mathbf{Q}_n \boldsymbol{\mu}_n]_i}{[\mathbf{Q}_n]_{i,i}} \right)^2. \tag{8}$$

aqui, $\boldsymbol{\mu}_n$ é o vetor média do sinal e i é o índice da linha ou coluna da base a ser descartada.

3) ***SKRLS*** *(Sparse Kernel Recursive Least Squares):* O algoritmo SKRLS [15] é uma variante esparsa do KRLS que utiliza a matriz hessiana da função de perda para controlar a complexidade do modelo. Para cada nova amostra $\mathbf{x}_n$, a significância é calculada usando a contribuição da amostra para a curvatura da função de perda:

$$\Delta\mathcal{L}_n = \frac{1}{2} \Delta\boldsymbol{\alpha}_n^\top \mathbf{H}_n \Delta\boldsymbol{\alpha}_n, \tag{9}$$

em que $\Delta\boldsymbol{\alpha}_n$ é o incremento nos coeficientes $\boldsymbol{\alpha}_n$ se $\mathbf{x}_n$ fosse adicionado ao dicionário, e $\mathbf{H}_n$ é a matriz hessiana da função de perda $\mathcal{L}$, $\mathbf{H}_n = \frac{\partial^2 \mathcal{L}_n}{\partial \boldsymbol{\alpha}_n^2}$, correspondente à amostra. Se $\Delta L_n \geq \nu$, em que ν é um limiar pré-definido, a amostra é adicionada ao dicionário; caso contrário, é descartada. A combinação entre o critério de significância baseado na hessiana e a atualização RLS permite que o SKRLS opere com complexidade reduzida, sendo adequada para aprendizado *online* [15].

4) ***KNLMS*** *(Kernel Normalized LMS):* O algoritmo KNLMS [14] é uma extensão do NLMS para espaços do tipo RKHS. Assim como o NLMS, ele normaliza sua taxa de aprendizado η com base na energia do sinal de entrada:

$$\eta_n = \frac{\eta}{\lVert \mathbf{x}_n \rVert^2 + \epsilon}, \tag{10}$$

em que ϵ é uma constante para evitar divisão por zero. Isso garante estabilidade e evita oscilações excessivas, problema comum no LMS tradicional. Além disso, o KNLMS emprega o denominado *critério de coerência* para controle de esparsidade do dicionário. Para cada nova amostra $\mathbf{x}_n$, o critério é definido por:

$$\max_{1 \leq j \leq m} \lvert k(\mathbf{x}_n, \mathbf{x}_j)\rvert < \nu, \tag{11}$$

em que: $k(\cdot, \cdot)$ é a função de kernel, ν é um limiar de coerência ($0 < \nu < 1$). Se a condição acima for violada, $\mathbf{x}_n$ é descartada; caso contrário, é adicionado ao dicionário.

5) ***QKLMS*** *(Quantized Kernel Least Mean Squares):* O algoritmo QKLMS [18] é uma variante adaptativa do KLMS que controla o crescimento das instâncias do dicionário por meio de uma técnica de quantização. Um limiar ν é usado para avaliar a inclusão de novas amostras com base na distância euclidiana aos centros existentes, reduzindo redundâncias e mantendo a esparsidade. Para cada nova amostra $\mathbf{x}_n$, calcula-se a distância euclidiana ao centro mais próximo no dicionário atual $\mathcal{D} = \{\mathbf{c}_1, \ldots, \mathbf{c}_M\}$:

$$d(\mathbf{x}_n, \mathcal{D}) = \min_{1 \leq j \leq M} \lVert \mathbf{x}_n - \mathbf{c}_j \rVert. \tag{12}$$

Se $d(\mathbf{x}_n, \mathcal{D}) > \nu$, $\mathbf{x}_n$ é adicionado a $\mathcal{D}$ como novo centro. Caso contrário, o dicionário $\mathcal{D}$ não é alterado. A quantização garante coerência entre os centros ao limitar a distância máxima entre amostras similares. O cálculo da distância tem custo $\mathcal{O}(M)$, linear em relação ao tamanho do dicionário, o que otimiza o algoritmo para aplicações em tempo real.

[1]No KRLS-T, o *critério de relevância* é definido neste trabalho pelos componentes: (i) não redundância (7), e (ii) baixo impacto na remoção(8).

A Tabela I apresenta de forma resumida os diversos critérios de esparsidade e o respectivo algoritmo KAF associado.

Na literatura especializada, embora os KAFs sejam uma abordagem promissora e competitiva comparada a outros modelos de AM, não foram encontrados estudos que apliquem essas técnicas especificamente para a predição de longo prazo da precipitação média mensal de chuvas.

Para efeito de análise de desempenho, nesse trabalho os algoritmos descritos são comparados com os algoritmos tradicionais de SVR-ν e o LSSVR.

III. Metodologia

A. Dados Experimentais e Pré-processamento

Para realizar os experimentos dessa pesquisa, utilizou-se o sistema operacional Ubuntu 24.10, com códigos e algoritmos implementados a partir do ambiente OCTAVE 9.2.0 [19]. Os códigos para implementar os KAFs são adaptados do toolbox KAFBOX [7].

Os dados analisados consistem de uma série histórica de precipitação pluviométrica mensal acumulada (em milímetros) da região metropolitana da cidade de Fortaleza, Ceará. As informações coletadas foram obtidas diretamente do site da FUNCEME (Fundação Cearense de Meteorologia e Recursos Hídricos) [20], e abrangem o período de Jan/1974 a Dez/2024 (612 observações).

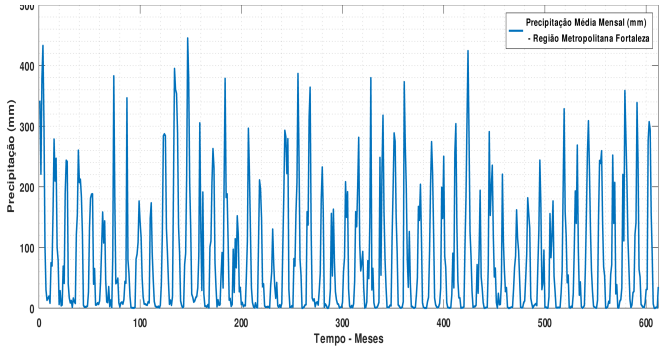


Figura 1. Precipitação média mensal de chuvas da região metropolitana de Fortaleza, Ceará - 1974/2024

Nesse artigo, optou-se por adotar uma modelagem estritamente univariada para preservar a série em sua forma bruta (sem dessazonalização ou remoção de tendência *a priori*). O objetivo é avaliar a capacidade intrínseca dos modelos em capturar diretamente a dinâmica complexa e potencialmente não estacionária da precipitação [2].

A Figura 1 ilustra a série temporal de chuva, caracterizada por alta variabilidade sazonal e interanual, típica de climas semiáridos.

B. Divisão dos Dados e Estratégia de Avaliação

A capacidade preditiva de modelos de aprendizagem de máquinas normalmente é avaliada por meio de técnicas estatísticas de validação baseada em geral nos erros de predição (e.g., RMSE ou MAE) ou na correlação entre os valores preditos e os observados da saída (coeficientes de correlação e de determinação). [21].

No presente estudo, para avaliar de maneira mais realista a capacidade preditiva dos modelos selecionados, adotou-se a técnica *holdout* com uma divisão estritamente cronológica. Para isso, utilizou-se os 396 dados mais recentes da série disponível (desde Jan/1994 até Dez/2024).

A sequência foi particionada da seguinte forma: os primeiros 360 meses (30 anos)[2] de Jan/1992 a Dez/2021, usados para o treinamento um passo à frente (*one-step-ahead*,OSA); os 24 meses subsequentes (Jan/2022 a Dez/2023) usados para otimizar os hiperparâmetros, conforme técnica discutida a seguir e, finalmente, os últimos 12 meses (Jan-Dez/2024) são então destinados a simular a previsão de longo prazo, mês a mês, e avaliar a generalização do modelo em dados não usados nas etapas anteriores.

A avaliação da capacidade de generalização de longo prazo com múltiplos passos à frente (*multi-step ahead*, MSA) foi realizada por meio de *simulação livre* em 36 meses consecutivos, iniciados imediatamente após o treinamento OSA. Neste modo iterativo, para as previsões do modelo para um passo à frente, os dados de treinamento foram aplicados para estimar o modelo não linear: $y_n = f(\mathbf{x}_n)$, onde $\mathbf{x}_n = (\mathbf{x}_{n-1}, \mathbf{x}_{n-2}, \ldots, \mathbf{x}_{n-k})$, técnica conhecida como teorema de Takens [22].

Para o presente trabalho, o problema de predição utilizou os dados dos 24 meses anteriores para prever a média de precipitação do mês atual, com a cobertura contínua dos períodos designados para validação (primeiros 24 meses da simulação livre) e teste (últimos 12 meses da simulação livre). Esta estratégia é desafiadora, pois testa explicitamente a estabilidade do modelo e sua robustez à acumulação e propagação de erros de previsão [23].

C. Índice IGNOS, Seleção de Hiperparâmetros e Modelo

Este trabalho introduz o índice IGNOS (Índice de Generalização Não linear de Ótimo-relativo ao Sobreajuste) para seleção de modelos em séries temporais não lineares, validado empiricamente neste artigo como indicador confiável de generalização de modelos preditivos.

A partir do conceito de viés-variância em [24], essa métrica permite facilitar a seleção de modelos com bom equilíbrio entre ajuste e generalização e diagnosticar riscos de sobreajuste ou subajuste. O índice é definido como:

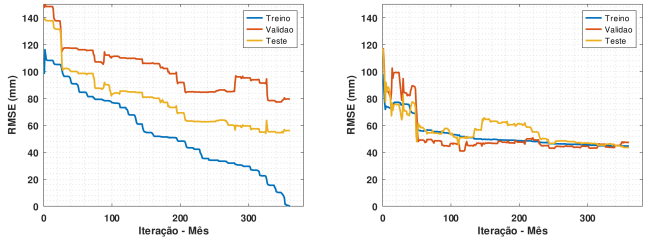
$$\text{IGNOS} = \frac{2r_v r_t}{1 + r_v + r_t} + r_v \frac{|r_v - r_t|}{1 + r_t} \tag{13}$$

em que r_t e r_v são, respectivamente, o RMSE calculado nos dados de treino e o calculado no período de validação.

O primeiro termo, $\frac{2r_v r_t}{1+r_v+r_t}$, penaliza modelos com erros altos em treino ou validação, identificando *underfitting* ou

[2]O tempo de 30 anos como conjunto de treinamento foi escolhido por ser esse o período utilizado pela FUNCEME como "normal climatológico", uma recomendação padrão da *World Metereological Organization*-WMO.

modelos ineficazes. O segundo, $r_v \frac{|r_v - r_t|}{1+r_t}$, amplifica casos em que $rv \gg rt$, priorizando a detecção de *overfitting*. Um valor IGNOS menor indica um melhor compromisso viés-variância para o modelo avaliado.



(a) KRLS-ALD - RMSE (mm) (b) KRLS-ALD - IGNOS

Figura 2. Curva de aprendizado KRLS-ALD: a) melhor modelo via menor RMSE de treino; b) melhor modelo via menor IGNOS

Para ilustrar o efeito dessa técnica na seleção de modelos, inicialmente utilizou-se apenas o algoritmo KRLS-ALD e o RMSE de treino (R_{tr}) como mecanismo de seleção do melhor modelo de hiperparâmetros para predição, como é feito tradicionalmente.

Neste cenário, obteve-se um $R_{tr} = 3,0101e^{-13}mm$, um valor muito próximo de zero, enquanto que o valor de IGNOS foi 6334,4, muito alto, o que caracteriza um aprendizado com sobreajuste severo. Isso pode ser observado pela diferença entre as curvas de aprendizado de validação e treino no gráfico da Figura 2(a).

No segundo cenário, o índice IGNOS é usado como técnica de seleção do melhor modelo. Neste caso, obtém-se $R_{tr} = 44,655mm$ e $IGNOS = 48,43$, o que coincide com o comportamento mais confiável de generalização obtido no gráfico da Figura 2(b).

D. Seleção de Hiperparâmetros e Modelo

A otimização dos hiperparâmetros para cada algoritmo KAF, SVR-ν e LSSVR foi realizada por meio de busca em grade. O critério para selecionar a melhor combinação de hiperparâmetros foi o desempenho durante a fase de *validação* de simulação livre via comparação com o índice IGNOS. O modelo final para cada algoritmo, com os hiperparâmetros selecionados via IGNOS, foi então considerado para avaliação no teste.

A Tabela II mostra os hiperparâmetros selecionados para cada algoritmo analisado neste artigo. Neste contexto, por simplicidade de notação, os critérios de seleção utilizados como limiares para a inclusão de novos dados no dicionário são referidos como $\boldsymbol{\nu}$.

Os demais hiperparâmetros são: γ a largura de banda da função de kernel; η fator de aprendizagem; β o fator de esquecimento do SKRLS; para o KRLS-T, temos: M é o tamanho fixo do dicionário, sn_2 a regularização, enquanto λ é seu fator de esquecimento; e, finalmente, C a regularização para o SVR-ν ou LSSVR.

E. Métricas de Avaliação e Comparação Final

O desempenho final dos modelos otimizados foi avaliado no período de teste, os últimos 12 meses, em simulação livre. As métricas quantitativas primárias, Tabela III, incluem RMSE e NMSE de treino, validação e teste, R^2 (coeficiente de determinação), tamanho do modelo (número de Instâncias/SVs) e tempo de treino. O índice IGNOS também foi considerado como indicador de generalização.

Tabela II
Hiperparâmetros para cada modelo avaliado.

Algoritmo	**Hiperparâmetros**							
	$\boldsymbol{\nu}$	$\boldsymbol{\gamma}$	$\boldsymbol{\eta}$	$\boldsymbol{\beta}$	**M**	$\mathbf{sn_2}$	λ	**C**
KRLS-ALD	0,4	5e-6	—	—	—	—	—	—
KRLS-T	0,1	9e-6	—	—	50	0,5	1	—
SKRLS	1	3e-5	—	0,94	—	—	—	—
KNLMS	0,2	3e-6	0,4	—	—	—	—	—
QKLMS	8e4	1.6e-5	0,7	—	—	—	—	—
LS-SVR	-	6e-6	—	—	—	—	—	1
SVR-ν	0,1	9,7e-6	-	—	—	—	—	210

A comparação global entre os modelos priorizou uma análise integrada dessas métricas e das seguintes evidências visuais: curvas de erro (RMSE) de teste ao longo da simulação (Figura 3), curvas de predição e aderência temporal às observações de teste (Figura 4), e capacidade de captura de anomalias climáticas (Figura 5).

Vale ressaltar que, dada a amostra limitada (N=12 meses) no período de teste final e as conhecidas limitações de potência e confiabilidade de testes como Diebold-Mariano/HLN nestes cenários [25], optou-se por não incluir estes testes formais na presente pesquisa.

As conclusões sobre o desempenho relativo dos modelos se fundamentam no conjunto abrangente das evidências quantitativas, diagnósticas e visuais coletadas ao longo da rigorosa simulação livre de 36 meses.

IV. Resultados e Discussão

Para os experimentos a seguir, os hiperparâmetros dos modelos são selecionados por meio de busca em grade e validação *holdout* conforme descrito na seção III-B, e definidos conforme a Tabela II.

A. Métricas de Desempenho

Os resultados quantitativos apresentados na Tabela III demonstram que dois dos algoritmos KAFs, o KRLS-T e o KRLS-ALD, superaram os modelos tradicionais LSSVR e SVR-ν em precisão preditiva e eficiência computacional.

O KRLS-T alcançou o segundo menor RMSE de teste (44.341 mm) e R^2=0.810 com apenas 50 instâncias em seu dicionário. Além disso, o algoritmo manteve adaptabilidade às variações temporais, mesmo com o fator de esquecimento inativo (λ=1), o que pode ser observador em sua curva preditiva mensal na Fig. 3.

Essa capacidade contínua de generalização, decorre possivelmente de seu mecanismo eficaz de renovação do dicionário de tamanho fixo (M=50). A técnica seleciona continuamente as bases mais relevantes e remove as consideradas obsoletas, o que assegura um acompanhamento dinâmico das flutuações ao longo do tempo.

Tabela III
Resultados das Métricas de Desempenho, IGNOS, SVs e Tempo de Treino para cada Algoritmo Analisado.

Algoritmo	Treino			Validação			Teste			Seleção	#Dic.	Tempo
	RMSE	NMSE	R^2	RMSE	NMSE	R^2	RMSE	NMSE	R^2	IGNOS		
KRLS-ALD	44.655	0.227	0.773	47.140	0.171	0.822	47.140	**0.177**	**0.818**	48.430	111	**0.051**
KRLS-T	**42.433**	**0.205**	**0.795**	**46.010**	**0.163**	**0.830**	**44.341**	0.185	0.810	**47.445**	**50**	0.114
SKRLS	57.946	0.381	0.617	57.393	0.253	0.736	40.390	0.154	0.842	57.711	337	0.204
KNLMS	59.908	0.408	0.591	60.790	0.284	0.704	52.482	0.259	0.733	60.731	4	0.137
QKLMS	51.313	0.299	0.700	51.658	0.205	0.786	44.800	0.189	0.806	51.330	143	0.059
LSSVR	**42.139**	**0.202**	**0.798**	52.640	0.213	0.778	45.620	0.133	0.855	59.622	360	2.406
SVR-ν	49.597	0.202	0.720	**48.892**	**0.183**	**0.809**	**42.524**	**0.116**	**0.874**	**49.923**	**128**	**2.384**

O KRLS-ALD, por sua vez, destacou-se pela eficiência: com RMSE de teste (R_{te}) de 47,140 mm, e R^2=0,818, e um tempo de treino de 0,051s, ou seja, o mais rápido entre todos os algoritmos. Seu mecanismo de dependência linear aproximada (ALD) limitou o dicionário a 111 instâncias. Isso reduz a sua complexidade computacional sem comprometer a acurácia.

Os demais KAFs: SKRLS, KNLMS e QKLMS tiveram desempenho inferior, o que evidencia possíveis limitações estruturais. O SKRLS, por exemplo, obteve o menor RMSE de teste (R_{te}=40,390mm; R^2=0,842), mas exigiu 337 instâncias no dicionário e registrou $IGNOS = 57,711$, o que representa *underfitting* severo. Isso sugere uma falha no balanço precisão-esparsidade, possivelmente devido à dependência excessiva da matriz hessiana em selecionar amostras sem significância.

Já o KNLMS (R_{te}=52,482mm e R^2=0,733), com apenas 4 vetores-protótipos no dicionário, mostrou-se excessivamente simplista, e resultou em *underfitting* (IGNOS=60,731). Isso o torna possivelmente incapaz de capturar a não estacionariedade da série temporal em outras condições. Enquanto que o QKLMS (R_{te}=44,80mm e R^2=0,806) apresenta um desempenho intermediário.

Os modelos tradicionais de vetores-suporte, embora competitivos em teste: (SVR-ν: R_{te}=42,524mm, R^2=0,874; LSSVR: R_{te}=45,620mm, R^2=0,855) exibiram treino lento ($\approx 2,4$s, 47 vezes mais que o KRLS-ALD), uma fragilidade operacional importante. Além disso, o SVR-ν demonstrou instabilidade de generalização (RMSE validação=48,892mm vs. treino), o que sinaliza subajuste, apesar do dicionário pequeno com 128 instâncias.

Finalmente, o LSSVR por ser um modelo em lote, requer todas as amostras de treinamento (360 instâncias) para compor sua estrutura, o que reflete sua alta complexidade computacional, O(N^3). Ainda assim, mesmo com tamanha carga de instâncias, seu R^2=0,855 foi bem próximo ao do KRLS-T (R^2=0,810) com um dicionário sete vezes menor.

B. Curvas de Predição - Teste

As curvas de predição destacam a superioridade dos KAFs em capturar a dinâmica da média de precipitação mensal em longo prazo. A Figura 3 mostra o resultado do melhor deles, KRLS-T, em comparação com o SVR-ν.

Percebe-se a superioridade adaptativa do KRLS-T, mesmo com um R_{te}=44,341mm maior que o SVR-ν. Sua estratégia preditiva permite maior fidelidade às mudanças rápidas da série, sem comprometer sua generalização, o que lhe confere o melhor IGNOS do estudo (47,445). Em contrapartida, o SVR-ν alcança um RMSE competitivo por meio de uma estratégia de modelagem que lhe rende um bom IGNOS ao custo da fidelidade aos extremos.

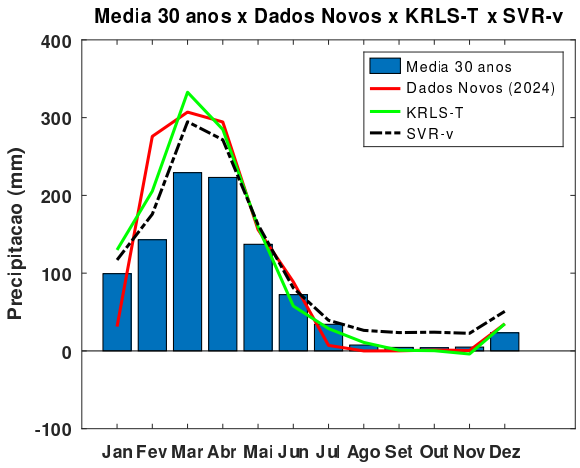


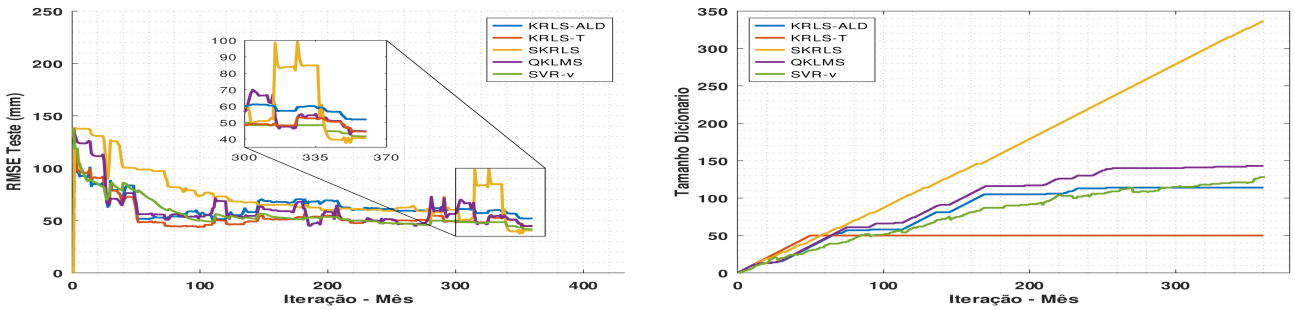
Figura 3. Curva preditiva mensal: KRLS-T, SVR-ν e média 30 anos

C. Curvas de Aprendizado - Teste

As curvas de aprendizado (Figura 4) revelaram padrões distintos de convergência e generalização entre os modelos. Como exemplo, os algoritmos KAFs: KRLS-ALD, KRLS-T, e QKLMS exibiram convergência rápida (< 60 iterações), enquanto o SKRLS teve desempenho semelhante ao SVR-ν, que só convergiu após 100 iterações.

Os modelos preditivos do KRLS-T, KRLS-ALD e SVR-ν mostraram comportamentos muito próximos de convergência, com bastante estabilidade, o que reflete um comportamento consistente com o IGNOS de cada um: 47,445, 48,430 e 49,923, respectivamente. Isso indica um equilíbrio considerável entre viés e variância.

Em contrapartida, embora os algoritmos SKRLS e QKLMS convirjam para valores baixos de R_{te}, conforme observa-se na Figura 4(a), apresentaram oscilações erráticas ao longo da etapa de treinamento e teste, e também R_{tr} relativamente mais altos (Tabela III). Isso



(a) Curva de aprendizagem - RMSE teste

(b) Crescimento do dicionario ao longo do treinamento

Figura 4. a) Curva de aprendizado - RMSE de teste em milímetros (mm) e b) Tamanho do dicionário para diversos KAFs e SVR-ν

demonstra que estes algoritmos possuem uma incapacidade em estabilizar, tornando-os inviáveis para aplicações práticas.

Finalmente, a Figura 4(b) mostra a evolução do dicionário à medida que o algoritmo aprende a generalizar o padrão da curva de precipitação pluviométrica. Observa-se a superioridade do KRLS-T e a equivalência entre o QKLMS, KRLS-ALD e SVR-ν.

D. Capacidade de Captura de Anomalias em Relação à Média Histórica

A Figura 5 apresenta o desvio percentual acumulado em relação à média climatológica de 30 anos (1994–2023) para todos os modelos, bem como para os dados reais observados em 2024, segmentados por períodos estratégicos: 1º trimestre, quadrimestre chuvoso e 1º semestre.

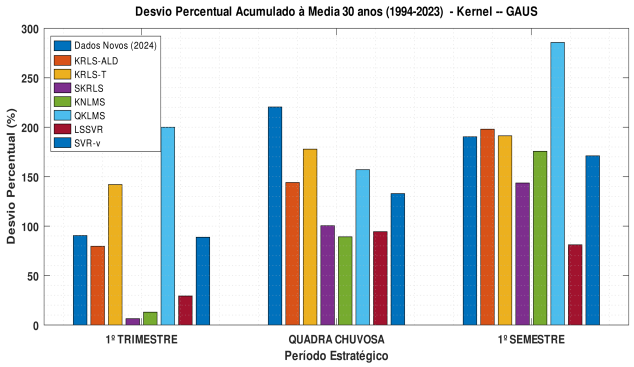


Figura 5. Desvio em relação à média dos dados novos e de todos os algoritmos analisados

A análise de desvios percentuais em relação à média histórica (5), permite avaliar a robustez dos modelos em regimes anômalos. A performance mais consistente foi observada nos modelos KRLS-T e KRLS-ALD.

No período mais abrangente (1º semestre), ambos replicaram com alta acurácia o desvio acumulado de +190% dos dados reais, enquanto modelos em lote (SVR-ν e o LSSVR) subestimaram essa anomalia.

A análise dos outros períodos revela um cenário mais complexo. Na "quadra chuvosa", todos os algoritmos subestimaram os dados reais. No entanto, KRLS-T, seguido pelo KRLS-ALD, QKLMS e SVR-ν alcançaram a maior precisão, respectivamente, enquanto os demais tenderam a subestimar bastante o volume de chuva nesse período.

Em contraste, os algoritmos SKRLS, KNLMS e LSSVR exibiram desvios consideráveis em todos os períodos estratégicos (1º trimestre, quadra chuvosa e 1º semestre), o que demonstra a incapacidade desses modelos em seguir as variações climáticas com acurácia.

Finalmente, a Figura 6 exibe o desempenho em toda a série temporal para os melhores modelos (KRLS-T e SVR-ν). A comparação das Figuras 6(a) (KRLS-T) e 6(b) (SVR-ν) evidencia o desempenho superior do KRLS-T em toda a série temporal, que demonstra notável aderência e adaptabilidade, pois captura a amplitude das variações e mantêm essa fidelidade nas fases de treinamento, validação e teste em simulação livre.

Já o SVR-ν apresenta uma predição notavelmente suavizada, uma característica que, embora resulte em um RMSE de teste competitivo, mascara uma limitação importante na modelagem de eventos extremos. Essa suavização pode ser entendida através da otimização que o modelo realiza para alcançar um IGNOS (49.923) razoável.

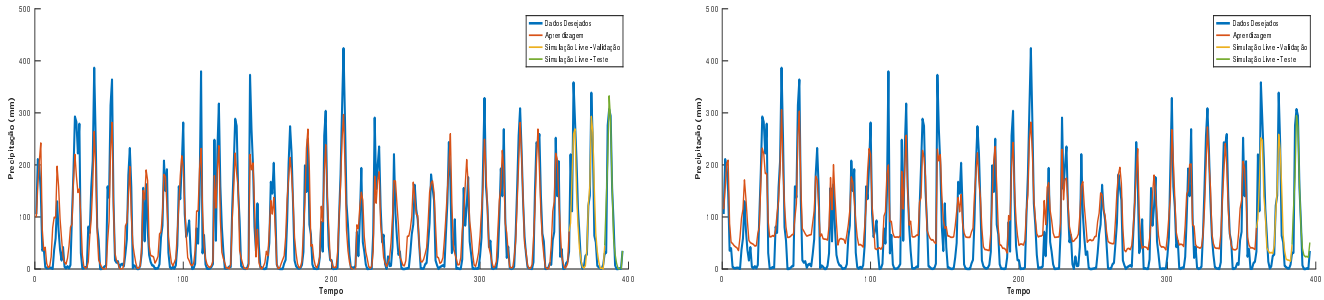
V. Conclusão

Os experimentos realizados e a avaliação por meio de um conjunto abrangente de métricas de desempenho revelaram a superioridade dos KAFs sobre os modelos de regressão kernel tradicionais (SVR-ν, LSSVR).

A métrica IGNOS foi indispensável para este diagnóstico. Um exemplo interessante é o SKRLS: apesar de apresentar o menor RMSE de teste, seu alto índice IGNOS revelou uma generalização deficiente, classificando-o como uma solução instável e pouco prática.

Da mesma forma, KNLMS e LSSVR apresentaram IGNOS elevados, indicando subajuste severo, enquanto o QKLMS mostrou instabilidade em seu aprendizado. No entanto, esse índice deve ser testado em outros cenários, para averiguar sua robustez e funcionalidade.

Os resultados obtidos consolidaram o KRLS-T como a solução de melhor equilíbrio, pois combina alta acurácia em treino e validação, menor índice IGNOS, dicionário compacto e aprendizado estável. Por sua vez, o KRLS-ALD destacou-se como a opção mais rápida, viável para cenários com restrições computacionais.



(a) Predição Mensal - KRLS-T

(b) Predição Mensal - SVR-ν

Figura 6. Curvas de Predição Mensal de Chuva: a) KRLS-T e b) SVR-ν

No geral, os modelos em lote (LSSVR, SVR-ν), apesar de desempenho competitivo em testes pontuais, foram superados pela natureza online, eficiência e robustez temporal dos KAFs, que se mostraram mais eficazes em capturar a dinâmica não-estacionária da série temporal de precipitação média mensal.

Como trabalhos futuros, propõe-se a extensão para abordagens multivariadas com variáveis exógenas (e.g., temperatura, umidade, El Niño) e a comparação com arquiteturas de aprendizado profundo, como LSTM, para avaliar a escalabilidade dos KAFs em cenários de maior complexidade.

Acknowledgment

Os autores agradecem à CAPES (finance code 001) e ao CNPq (processo 313251/2023-1) pelo apoio financeiro.

Referências

[1] S.-Q. Dotse, I. Larbi, A. M. Limantol, and L. C. De Silva, "A review of the application of hybrid machine learning models to improve rainfall prediction," *Model. Earth Syst. Environ.*, vol. 10, no. 1, pp. 19–44, Feb. 2024.

[2] K. L. Chong, S. H. Lai, Y. Yao, A. N. Ahmed, W. Z. W. Jaafar, and A. El-Shafie, "Performance enhancement model for rainfall forecasting utilizing integrated wavelet-convolutional neural network," *Water Resour. Manage.*, vol. 34, no. 8, pp. 2371–2387, Jun. 2020.

[3] P. K. Das, R. L. Sahu, and P. C. Swain, "Comparative analysis of machine learning models for rainfall prediction," *J. Atmos. Sol. Terr. Phys.*, vol. 264, no. 106340, p. 106340, Nov. 2024.

[4] J. D. A. Santos and G. A. Barreto, "Novel sparse LSSVR models in primal weight space for robust system identification with outliers," *J. Process Control*, vol. 67, pp. 129–140, Jul. 2018.

[5] H. Wang, H. Han, S. Zhang, and J. Ku, "An efficient kernel adaptive filtering algorithm with adaptive alternating filtering mechanism," *Digit. Signal Process.*, vol. 159, no. 104997, p. 104997, Apr. 2025.

[6] H. Zhao and B. Chen, "Kernel adaptive filters," in *Efficient Nonlinear Adaptive Filters*. Springer International Publishing, 2023, pp. 209–257.

[7] S. Van Vaerenbergh and I. Santamaría, "A comparative study of kernel adaptive filtering algorithms," in *2013 IEEE Digital Signal Processing and Signal Processing Education Meeting (DSP/SPE)*, 2013, pp. 181–186, software available at https://github.com/steven2358/kafbox/.

[8] J. Principe, W. Liu, and S. Haykin, *Kernel Adaptive Filtering: A Comprehensive Introduction*, ser. Adaptive and Cognitive Dynamic Systems: Signal Processing, Learning, Communications and Control. Wiley, 2011. [Online]. Available: https://books.google.com.br/books?id=eWUwB_P5pW0C

[9] J. Mercer, "XVI. functions of positive and negative type, and their connection the theory of integral equations," *Philos. Trans. R. Soc. Lond.*, vol. 209, no. 441-458, pp. 415–446, Jan. 1909.

[10] B. Schölkopf, R. Herbrich, and A. J. Smola, "A generalized representer theorem," in *Lecture Notes in Computer Science*, ser. Lecture notes in computer science. Springer Berlin Heidelberg, 2001, pp. 416–426.

[11] J. L. Rojo-Álvarez, M. Martínez-Ramón, J. Muñoz-Mariz, and G. Camps-Valls, *Adaptive Kernel Learning for Signal Processing*. John Wiley & Sons, Ltd, 2018, pp. 387–431.

[12] S. Van Vaerenbergh, M. Lazaro-Gredilla, and I. Santamaria, "Kernel recursive least-squares tracker for time-varying regression," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 8, pp. 1313–1326, 2012.

[13] Y. Engel, S. Mannor, and R. Meir, "The kernel recursive least-squares algorithm," *IEEE Transactions on Signal Processing*, vol. 52, no. 8, pp. 2275–2285, 2004.

[14] C. Richard, J. C. M. Bermudez, and P. Honeine, "Online prediction of time series data with kernels," *IEEE Transactions on Signal Processing*, vol. 57, no. 3, pp. 1058–1067, 2009.

[15] H. Fan and Q. Song, "A sparse kernel algorithm for online time series data prediction," *Expert Systems with Applications*, vol. 40, p. 2174–2181, 05 2013.

[16] S. Van Vaerenbergh, J. Via, and I. Santamaria, "A sliding-window kernel rls algorithm and its application to nonlinear channel identification," in *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, vol. 5, 2006, pp. V–V.

[17] Q. Song, X. Zhao, H. Fan, and D. Wang, "Robust recurrent kernel online learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 5, pp. 1068–1081, 2017.

[18] B. Chen, S. Zhao, P. Zhu, and J. C. Principe, "Quantized kernel least mean square algorithm," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 1, pp. 22–32, 2012.

[19] T. O. P. Developers, "Gnu octave," GNU Project. [Online]. Available: https://octave.sourceforge.io/

[20] Fundação Cearense de Meteorologia e Recursos Hídricos (FUNCEME), http://www.hidro.ce.gov.br/regiao_chuva_thiessen/mensal, 2025, acessado em 27 de abril de 2025.

[21] V. Cerqueira, L. Torgo, and C. Soares, "Model selection for time series forecasting an empirical analysis of multiple estimators," *Neural Process. Lett.*, vol. 55, no. 7, pp. 10 073–10 091, Dec. 2023.

[22] F. Takens, "Detecting strange attractors in turbulence," in *Lecture Notes in Mathematics*, ser. Lecture notes in mathematics. Springer Berlin Heidelberg, 1981, pp. 366–381.

[23] M. S. Duarte and G. A. Barreto, "Online sparse correntropy kernel learning for outlier-robust system identification," *IFAC-PapersOnLine*, vol. 52, no. 1, pp. 430–435, 2019.

[24] G. James, D. Witten, T. Hastie, R. Tibshirani, and J. Taylor, *An introduction to statistical learning*, ser. Springer texts in statistics. Springer International Publishing, 2023.

[25] D. Harvey, S. Leybourne, and P. Newbold, "Testing the equality of prediction mean squared errors," *Int. J. Forecast.*, vol. 13, no. 2, pp. 281–291, Jun. 1997.