Refinamento Sustentável e Explicável de Previsões Zero-Shot de Séries Temporais com Boosting

Diogo Tuler Chaves, Gisele Lobo Pappa e Wagner Meira Junior
Universidade Federal de Minas Gerais (UFMG)
Belo Horizonte, Brasil
{diogochaves, glpappa, meira}@dcc.ufmg.br

***Abstract*—A previsão de séries temporais recém-observadas é desafiadora quando há poucos dados locais disponíveis, mudanças recentes de regime ou necessidade de adaptação rápida. Modelos fundacionais de séries temporais, como o TimesFM, permitem previsão *zero-shot*, reduzindo a necessidade de treinamento específico para cada nova série. Entretanto, por serem otimizados para generalização ampla, esses modelos podem deixar erros sistemáticos quando a série alvo apresenta dinâmica local recente, deslocamentos de tendência, mudanças abruptas de nível ou padrões sazonais específicos. Este trabalho apresenta o Time BoostRer, uma estratégia híbrida que mantém o modelo fundacional congelado e utiliza XGBoost como uma camada leve de refinamento local. Em vez de ajustar os pesos do modelo original, a abordagem aprende correções a partir de atributos estatísticos e temporais de baixo custo, incluindo a previsão base, o índice do horizonte, resumos recentes da série e codificações cíclicas de calendário. Para reduzir o aprendizado a partir de padrões desatualizados, o refinador é treinado apenas no regime mais recente da série, identificado por detecção de pontos de mudança após remoção de tendência e sazonalidade. A avaliação combina métricas tradicionais de erro com uma métrica complementar baseada em similaridade de derivadas, voltada a medir fidelidade de tendência e curvatura. Os resultados em bases do Monash Time Series Forecasting Archive indicam que o refinamento melhora a qualidade das previsões em grande parte dos cenários, preservando eficiência computacional, interpretabilidade e baixo custo de adaptação. A proposta contribui, portanto, como uma alternativa prática ao *fine-tuning* completo de modelos fundacionais em séries temporais.**

***Index Terms*—Séries temporais, previsão zero-shot, modelos fundacionais, TimesFM, XGBoost, inteligência artificial explicável, IA sustentável.**

I. Introdução

A tomada de decisão em áreas como energia, saúde, finanças, mobilidade urbana e varejo depende frequentemente da capacidade de antecipar valores futuros a partir de observações históricas. Nesses contextos, a previsão de séries temporais apoia planejamento, alocação de recursos, dimensionamento de infraestrutura e identificação antecipada de riscos. Métodos clássicos, como ARIMA e suavização exponencial, seguem relevantes como baselines robustos, mas normalmente exigem modelagem e ajuste específicos para cada série, o que limita sua escalabilidade em cenários com grande volume de dados ou com necessidade de atualização frequente.

Nos últimos anos, a área passou de modelos locais para arquiteturas globais e, mais recentemente, para modelos fundacionais de séries temporais. Esses modelos são pré-treinados em coleções heterogêneas de séries e podem ser aplicados a novas tarefas em regime *zero-shot*, sem treinamento específico no domínio alvo. Essa característica torna sua utilização atraente quando se deseja gerar previsões rapidamente, com baixo esforço de engenharia e sem treinar um modelo do zero para cada nova série. A promessa central dessa abordagem é transferir regularidades temporais aprendidas em larga escala para domínios e frequências ainda não vistos durante a implantação.

Apesar dessas vantagens, a previsão *zero-shot* ainda pode falhar em padrões locais recentes. Um modelo fundacional congelado é treinado para bom desempenho médio em múltiplos domínios, mas essa generalização ampla pode não capturar mudanças abruptas de nível, deslocamentos de tendência, variações sazonais específicas ou desvios recentes provocados por eventos locais. Como consequência, uma previsão *zero-shot* fixa pode deixar erros sistemáticos que dependem da posição no horizonte de previsão, do último valor observado, do nível local da série e da posição de calendário do instante previsto.

Uma solução direta seria realizar *fine-tuning* do modelo fundacional para cada domínio ou conjunto de séries. Entretanto, esse procedimento exige otimização adicional, memória, tempo computacional, infraestrutura especializada e maior consumo de energia, especialmente quando muitas séries precisam ser adaptadas continuamente. Além disso, o ajuste fino altera a simplicidade operacional da inferência *zero-shot*, pois passa a exigir manutenção de versões adaptadas do modelo e novos ciclos de treinamento. Assim, o problema tratado neste trabalho é como adaptar previsões *zero-shot* a padrões recentes da série sem modificar o modelo fundacional.

Este trabalho propõe o Time BoostRer, uma alternativa leve e explicável para esse problema. O método utiliza um modelo fundacional congelado como previsor global e adiciona uma camada local de refinamento baseada em XGBoost. Em vez de modificar os pesos do modelo fundacional, a abordagem aprende uma função de correção a partir de atributos simples e semanticamente interpretáveis: previsão base, índice do horizonte, último valor observado, média móvel exponencial do contexto e codificações cíclicas de calendário. A hipótese central é que um modelo de *boosting* pode capturar erros estruturados deixados pela previsão *zero-shot* com custo computacional muito menor do que o ajuste completo do modelo original.

A metodologia segue uma estratégia em duas etapas.

Primeiro, o modelo fundacional produz previsões *zero-shot* para janelas de previsão. Em seguida, essas previsões são passadas para um refinador baseado em árvores, treinado para aprender correções locais a partir de sinais tabulares de baixo custo. Para tornar essa adaptação mais focada, o refinador é treinado apenas sobre o regime mais recente da série, identificado por detecção de pontos de mudança após remoção de tendência e sazonalidade. Dessa forma, busca-se evitar que padrões antigos, possivelmente incompatíveis com a dinâmica atual, influenciem a camada corretiva.

Além da acurácia ponto a ponto, este trabalho considera que uma boa previsão deve preservar a dinâmica local da série. Métricas como MAE e RMSE são fundamentais, mas não indicam se a previsão acompanha a direção, a inclinação ou a curvatura do alvo. Por isso, também é utilizada uma avaliação baseada em similaridade de derivadas, que compara as primeiras e segundas diferenças da previsão e da série observada. Essa avaliação complementa métricas tradicionais ao medir explicitamente a fidelidade de tendência.

As contribuições deste trabalho são quatro. Primeiro, propõe-se um refinador baseado em *boosting* para previsões *zero-shot* produzidas por um modelo fundacional congelado. Segundo, define-se uma representação tabular que combina previsão base, informações de horizonte, estatísticas recentes e atributos de calendário. Terceiro, utiliza-se seleção de regime recente por detecção de pontos de mudança para evitar que o refinador aprenda padrões antigos. Quarto, avalia-se o método com métricas tradicionais de erro, similaridade de derivadas e análises de interpretabilidade por importância de atributos e SHAP. Os resultados indicam que a estratégia melhora a qualidade das previsões em grande parte dos cenários avaliados, mantendo baixo custo incremental em relação à inferência *zero-shot* pura.

II. Trabalhos Relacionados

Trabalhos recentes em previsão de séries temporais têm migrado de modelos locais e específicos para modelos globais treinados em múltiplas séries. Métodos estatísticos clássicos permanecem importantes pela robustez e interpretabilidade, mas a necessidade de ajuste individual limita sua aplicação em larga escala. Arquiteturas neurais globais reduziram parte dessa limitação ao compartilhar parâmetros entre diferentes séries. Nesse contexto, PatchTST [8] mostrou que arquiteturas baseadas em transformadores podem ser eficazes em previsão de longo horizonte quando treinadas como modelos globais, antecipando parte do movimento posterior em direção a modelos fundacionais.

Modelos mais recentes, como TimesFM [1], Chronos [5], Lag-Llama [6] e Moirai [7], ampliam esse paradigma ao realizar pré-treinamento em grandes coleções heterogêneas de séries e inferência *zero-shot* ou *few-shot* em novas tarefas. A intuição comum desses modelos é que regularidades temporais podem ser transferidas entre domínios e frequências, permitindo que um previsor pré-treinado seja aplicado a séries não vistas sem recomeçar o processo de treinamento. Essa propriedade é especialmente útil em cenários nos quais há muitas séries curtas, dados locais limitados ou necessidade de implantação rápida.

Apesar desse avanço, a adaptação continua sendo um desafio central. O *fine-tuning* de um modelo grande pode melhorar o desempenho em um domínio específico, mas também exige nova otimização, maior uso de memória, complexidade de engenharia e uma pipeline menos simples do que a inferência *zero-shot*. Para muitas aplicações, a questão prática não é apenas se o ajuste fino melhora o resultado, mas se o ganho compensa o custo adicional em cada novo ambiente de uso. Esse problema se torna ainda mais relevante quando a adaptação precisa ser repetida em vários conjuntos de séries, em diferentes frequências ou ao longo do tempo.

Uma etapa leve de correção é atraente porque preserva o conhecimento amplo do modelo pré-treinado e mantém a arquitetura pesada congelada. Essa ideia se aproxima de estratégias de calibração pós-hoc e de *stacking*, nas quais um preditor forte é refinado por um segundo modelo treinado sobre erros ou saídas intermediárias. Métodos baseados em *boosting*, em particular, são adequados para esse cenário por modelarem interações não lineares em espaços tabulares de baixa dimensão [2]. Em uma pipeline de previsão, uma camada de árvores pode explorar relações entre previsão base, horizonte, estatísticas recentes e covariáveis temporais sem exigir atualização dos parâmetros do modelo sequencial original.

O método proposto também se relaciona à literatura de não estacionariedade e quebras estruturais. Séries temporais reais frequentemente apresentam mudanças de distribuição ao longo do tempo, incluindo alterações de nível, tendência, variância ou sazonalidade. Quando uma série passa por pontos de mudança, janelas antigas podem ser pouco informativas ou mesmo prejudiciais para uma camada corretiva local. O algoritmo PELT oferece uma forma eficiente de detectar esses pontos sob hipóteses adequadas [3]. Neste trabalho, essa ideia é usada não para substituir o modelo de previsão, mas para selecionar quais janelas devem treinar o refinador.

Por fim, este trabalho dialoga com a literatura de avaliação de previsões. Métricas tradicionais de erro são indispensáveis, mas descrevem apenas parte da qualidade de uma previsão. Uma trajetória prevista pode ter erro médio razoável e ainda assim não acompanhar a direção ou a curvatura da série real. Ao introduzir uma métrica de similaridade baseada em derivadas, este trabalho complementa a avaliação ponto a ponto com uma noção explícita de fidelidade dinâmica, particularmente relevante quando o objetivo é preservar a forma da trajetória futura.

III. Fundamentos

A. Modelos Fundacionais de Séries Temporais

Modelos fundacionais de séries temporais são pré-treinados em grandes coleções de dados para aprender padrões temporais transferíveis entre tarefas. No cenário *zero-shot*, uma janela de contexto

$$\mathbf{y}_{t-C:t-1} = (y_{t-C}, \ldots, y_{t-1}) \tag{1}$$

é fornecida como entrada, e o modelo produz uma previsão para os próximos H passos:

$$\hat{\mathbf{y}}_{t:t+H-1} = f(\mathbf{y}_{t-C:t-1}), \tag{2}$$

em que f representa o previsor pré-treinado. A principal vantagem dessa formulação é permitir previsão imediata sem treinamento específico. A limitação é que, por ser otimizado para bom desempenho médio em múltiplos domínios, o modelo pode não capturar efeitos locais recentes, como mudança de regime, tendência curta ou desvios dependentes de calendário.

No contexto deste trabalho, o modelo fundacional é tratado como um componente fixo. Isso significa que seus pesos não são atualizados, e sua saída é usada como uma previsão base. Essa decisão preserva a característica operacional da previsão *zero-shot*: o modelo continua sendo aplicado diretamente a novas séries, enquanto a adaptação local é delegada a uma etapa externa, menor e mais simples de treinar.

B. Boosting

Boosting é uma estrutura de aprendizado supervisionado que combina aprendizes fracos, geralmente árvores de decisão, para formar um preditor forte [12]. Em regressão, esse tipo de modelo é útil para capturar erros estruturados que permanecem após uma previsão inicial. Em gradient tree boosting, o preditor pode ser escrito como

$$g_M(x) = \sum_{m=1}^{M} \eta b_m(x), \tag{3}$$

em que b_m é uma árvore de decisão e η é a taxa de aprendizado. Cada nova árvore busca reduzir os erros ainda não explicados pelo conjunto anterior, produzindo uma correção progressiva da previsão.

O XGBoost [2] é uma implementação eficiente e regularizada desse princípio. Sua adequação ao problema vem do fato de que o refinamento proposto utiliza uma representação tabular pequena, composta por atributos heterogêneos e interpretáveis. Assim, o modelo pode capturar interações entre a previsão do TimesFM, a posição no horizonte, o nível local da série e variáveis de calendário, mantendo o custo de treinamento relativamente baixo e oferecendo mecanismos diretos de interpretação por importância de atributos.

IV. Metodologia

Esta seção apresenta o Time BoostRer. A metodologia segue quatro fases principais: uma série univariada é carregada e dividida em treino, validação e teste; o trecho de treino é processado para identificar o regime estrutural mais recente; janelas deslizantes são enviadas a um modelo fundacional congelado para gerar previsões *zero-shot*; e essas previsões são convertidas em uma matriz tabular usada para treinar um refinador XGBoost.

A escolha por uma camada externa de refinamento tem duas motivações. A primeira é estatística: se a previsão *zero-shot* deixa padrões residuais associados ao horizonte, ao nível recente ou ao calendário, um modelo tabular de *boosting* pode aprender essas regularidades de forma direta. A segunda é computacional: como o modelo fundacional permanece congelado, a adaptação local não exige retropropagação através da arquitetura principal, nem armazenamento de múltiplas versões ajustadas do modelo.

A. Boosting como Refinador

Seja $\{y_t\}_{t=1}^{T}$ uma série univariada, C o comprimento do contexto e H o horizonte de previsão. Para cada origem válida t, define-se o contexto

$$\mathbf{x}_t^{\text{ctx}} = (y_{t-C}, \ldots, y_{t-1}) \tag{4}$$

e o alvo futuro

$$\mathbf{y}_t^{\text{fut}} = (y_t, \ldots, y_{t+H-1}). \tag{5}$$

O modelo fundacional congelado produz a previsão base

$$\mathbf{p}_t = (p_{t,0}, \ldots, p_{t,H-1}) = f_0(\mathbf{x}_t^{\text{ctx}}). \tag{6}$$

O refinador transforma o problema sequencial em regressão tabular. Para cada janela e cada horizonte $h \in \{0, \ldots, H-1\}$, constrói-se um vetor de atributos $\phi_{t,h}$ e utiliza-se como alvo o valor escalar y_{t+h}. A previsão final é dada por

$$\hat{y}_{t+h} = g(\phi_{t,h}), \tag{7}$$

em que g é o modelo XGBoost. Embora a abordagem seja motivada por correção residual, a implementação prediz diretamente o valor final, e não o resíduo $y_{t+h} - p_{t,h}$. Assim, a previsão do modelo fundacional funciona como o principal sinal de entrada, mas o refinador pode aprender uma transformação não linear a partir de todos os atributos disponíveis.

O treinamento usa erro quadrático:

$$\mathcal{L}(g) = \frac{1}{N} \sum_{n=1}^{N} \left(y_n - g(\phi_n)\right)^2, \tag{8}$$

onde cada linha n corresponde a um passo futuro de uma janela. Esse desenho permite aprender correções dependentes do horizonte e interações entre nível recente e posição no calendário, mantendo o modelo fundacional inalterado.

B. Engenharia de Atributos e Seleção de Regime

A matriz de treinamento possui uma linha para cada par janela–horizonte. O vetor de atributos é definido como

$$\phi_{t,h} = [p_{t,h},\ h,\ \ell_t,\ e_t,\ c_{t+h}^{(1)}, \ldots, c_{t+h}^{(6)}], \tag{9}$$

em que $p_{t,h}$ é a previsão *zero-shot*, h é o índice do horizonte, $\ell_t = y_{t-1}$ é o último valor observado, e_t é a média móvel exponencial do contexto e $c_{t+h}^{(1)}, \ldots, c_{t+h}^{(6)}$ são codificações cíclicas de calendário. Para um índice periódico u com período P, os atributos cíclicos são dados por

$$\sin\left(2\pi \frac{u}{P}\right), \qquad \cos\left(2\pi \frac{u}{P}\right). \tag{10}$$

A média móvel exponencial resume o nível recente da série. Para valores de contexto $(z_1, \ldots, z_C)$ e *span* s, ela é calculada recursivamente por

$$\text{EMA}_1 = z_1, \qquad \text{EMA}_j = \alpha z_j + (1-\alpha)\text{EMA}_{j-1}, \tag{11}$$

com

$$\alpha = \frac{2}{s+1}. \tag{12}$$

O último valor da EMA é utilizado como uma estimativa suavizada do nível local. Em conjunto, o último valor observado e a EMA fornecem duas visões complementares do contexto: a primeira ancora a previsão no ponto mais recente, enquanto a segunda reduz a sensibilidade a ruídos pontuais. Essa combinação é importante porque erros da previsão base podem estar associados tanto a deslocamentos abruptos de nível quanto a tendências locais mais suaves.

Para evitar que o refinador aprenda padrões antigos, a etapa de treinamento é restrita ao regime mais recente. A série é inicialmente tratada por interpolação linear e preenchimento de bordas. Em seguida, uma tendência linear é removida por mínimos quadrados:

$$y_t \approx at + b + r_t^{(1)}, \qquad r_t^{(1)} = y_t - (at+b). \tag{13}$$

Depois, remove-se uma componente sazonal simples de período m:

$$s_j = \frac{1}{|\mathcal{I}_j|} \sum_{t \in \mathcal{I}_j} r_t^{(1)}, \qquad \mathcal{I}_j = \{t : t \equiv j \pmod m\}, \tag{14}$$

$$r_t^{(2)} = r_t^{(1)} - s_{t \bmod m}. \tag{15}$$

O PELT é aplicado à sequência residual $\{r_t^{(2)}\}$, resolvendo genericamente

$$\min_{\tau_1,\dots,\tau_K} \sum_{k=0}^{K} \mathcal{C}\left(r^{(2)}_{\tau_k+1:\tau_{k+1}}\right) + \beta K, \tag{16}$$

com penalidade $\beta = \log(n)/2$. Quando pontos de mudança são identificados, seleciona-se o último ponto antes do fim da série e mantém-se o sufixo posterior a ele para gerar as janelas de treino. A remoção de tendência e sazonalidade é usada apenas para melhorar a detecção; o corte final é aplicado à série original. Essa distinção é relevante: o modelo de previsão e o refinador continuam observando valores na escala original da série, enquanto a decomposição auxiliar atua somente como etapa de identificação de regime.

C. Construção da Matriz de Treinamento

Após a seleção do regime recente, são geradas janelas deslizantes. Uma origem t é válida quando

$$t \geq C, \qquad t + H \leq T', \tag{17}$$

em que T' é o comprimento do sufixo mantido. A implementação usa uma heurística adaptativa de *stride* para controlar o número de janelas e reduzir sobreposição excessiva. Na prática, o espaçamento entre janelas é ajustado conforme o tamanho disponível da série, o horizonte configurado e a necessidade de manter um número mínimo de exemplos de treinamento. Isso evita tanto a geração de janelas quase idênticas quanto a escassez de amostras em séries curtas.

A matriz final de treinamento é

$$X = \begin{bmatrix} \phi_{t_1,0} \\ \phi_{t_1,1} \\ \vdots \\ \phi_{t_M,H_M-1} \end{bmatrix}, \qquad \mathbf{y} = \begin{bmatrix} y_{t_1} \\ y_{t_1+1} \\ \vdots \\ y_{t_M+H_M-1} \end{bmatrix}. \tag{18}$$

A mesma lógica é usada na inferência, reconstruindo as previsões achatadas para o formato janela–horizonte. Como cada linha da matriz representa um passo futuro de uma janela, o refinador pode aprender correções específicas para diferentes posições do horizonte sem deixar de compartilhar informação entre elas. No contexto da monografia, as atividades desenvolvidas incluíram a revisão de modelos fundacionais de séries temporais, a implementação da pipeline com TimesFM, a construção da camada XGBoost, a seleção de regime recente, a definição das métricas e a análise dos resultados.

V. Configuração Experimental

A. Bases de Dados

Os experimentos foram conduzidos em um subconjunto de 15 bases provenientes do Monash Time Series Forecasting Archive [4], abrangendo diferentes domínios e frequências. A implementação opera sobre arquivos no formato `.tsf` e seleciona séries individuais por índice. Quando disponível, o horizonte de previsão definido pela própria base é utilizado; caso contrário, emprega-se a configuração definida para o experimento.

A implementação é estritamente univariada. Quando uma série carregada possui múltiplas dimensões, apenas a primeira dimensão é considerada. Valores ausentes são tratados por interpolação linear seguida de preenchimento por valores adjacentes. Atributos de calendário são gerados de acordo com a frequência temporal da base e alinhados às janelas de contexto e previsão.

Embora o Monash Archive seja frequentemente usado em avaliação de origem fixa, este trabalho utiliza uma divisão cronológica interna para apoiar desenvolvimento e avaliação controlada. A porção inicial da série é usada para gerar janelas de treinamento, enquanto regiões posteriores são reservadas para validação e teste. Assim, a ordem temporal é preservada e alvos futuros não são usados no treinamento. As amostras de treino são construídas exclusivamente a partir da região de treinamento por meio de múltiplas janelas contexto–futuro. Em contraste, validação e teste são avaliados por janelas extithold-out, nas quais o contexto vem do trecho mais recente disponível antes do alvo e o horizonte previsto corresponde à região futura reservada.

B. Métricas

As previsões são avaliadas com métricas ponto a ponto e métricas de fidelidade dinâmica. As métricas ponto a ponto quantificam proximidade numérica entre previsão e alvo, enquanto as métricas de derivadas avaliam se a previsão preserva direção, inclinação e curvatura local. Essa distinção é importante porque uma previsão pode reduzir erro médio e ainda assim representar mal a trajetória futura. As métricas

tradicionais utilizadas são MAE, RMSE, SMAPE modificada e MAE escalado. Para um horizonte de comprimento H:

$$\text{MAE} = \frac{1}{H}\sum_{i=1}^{H}|y_i - \hat{y}_i|, \qquad \text{RMSE} = \sqrt{\frac{1}{H}\sum_{i=1}^{H}(y_i - \hat{y}_i)^2}. \tag{19}$$

A SMAPE modificada é calculada como

$$\text{mSMAPE} = \frac{1}{H}\sum_{i=1}^{H}\frac{2|y_i - \hat{y}_i|}{\max(|y_i| + |\hat{y}_i| + \varepsilon, \delta + \varepsilon)}, \tag{20}$$

com $\varepsilon = 0{,}1$ e $\delta = 0{,}5$. Também se calcula o MAE escalado em relação a uma previsão ingênua que repete o último valor observado:

$$\text{MAE escalado} = \frac{\text{MAE}(y, \hat{y})}{\text{MAE}(y, \hat{y}^{\text{naive}})}. \tag{21}$$

Além das métricas ponto a ponto, utiliza-se similaridade baseada em derivadas para medir fidelidade de tendência. Se ∇y e $\nabla \hat{y}$ denotam as primeiras diferenças da série alvo e da previsão, a similaridade de primeira derivada é

$$S_1 = \frac{\langle \nabla y, \nabla \hat{y} \rangle}{\|\nabla y\| \, \|\nabla \hat{y}\| + \epsilon}. \tag{22}$$

A mesma formulação é aplicada às segundas diferenças:

$$S_2 = \frac{\langle \nabla^2 y, \nabla^2 \hat{y} \rangle}{\|\nabla^2 y\| \, \|\nabla^2 \hat{y}\| + \epsilon}. \tag{23}$$

Na implementação, ambos os escores são limitados ao intervalo $[0, 1]$. O escore S_1 enfatiza concordância de direção e inclinação local, enquanto S_2 enfatiza concordância de curvatura e mudanças de aceleração. Juntos, esses escores permitem identificar casos nos quais uma previsão está numericamente próxima, mas dinamicamente desalinhada, ou casos nos quais a magnitude ainda apresenta erro, mas a forma da trajetória é mais bem preservada.

C. Protocolo de Avaliação

As janelas de treinamento são geradas apenas da região de treino após seleção do regime recente. O modelo fundacional permanece congelado em todos os experimentos. O refinador é treinado sobre atributos extraídos das janelas de treino e de suas previsões *zero-shot*. A mesma lógica de construção de atributos é aplicada nas regiões de validação e teste.

A explicabilidade é obtida por importância de atributos do XGBoost e visualizações baseadas em SHAP [11]. Como a representação tabular é pequena e semântica, as correções podem ser interpretadas em termos de previsão base, horizonte, nível recente e calendário. Essa análise é descritiva, e não causal, mas permite verificar se o refinador atua como uma recalibração coerente da previsão fundacional ou se passa a substituir de maneira instável o modelo base. Os experimentos usam semente fixa para o XGBoost, resultando em execuções determinísticas sob a configuração adotada. Não foi realizada média sobre múltiplas sementes, portanto os resultados devem ser interpretados como execuções de configuração única para cada série selecionada. A implementação também inclui checagens simples de vazamento entre treino, validação e teste, embora a principal garantia venha da divisão cronológica dos dados.

VI. Resultados

Esta seção apresenta os resultados empíricos do Time BoostRer usando TimesFM como modelo fundacional. A análise considera métricas de similaridade de derivadas, métricas tradicionais de erro, interpretabilidade e custo computacional. O objetivo não é apenas verificar se o refinador reduz erro médio, mas também avaliar se a camada adicional produz correções coerentes com a dinâmica local da série e se esse ganho é obtido com custo compatível com a proposta de adaptação leve.

A. Similaridade de Derivadas

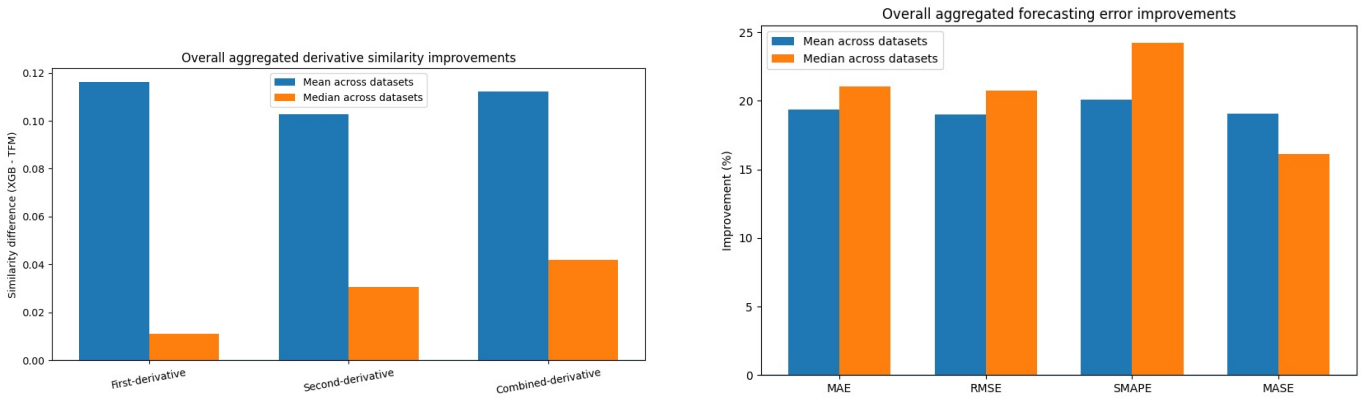
A Tabela I mostra que o Time BoostRer frequentemente melhora o alinhamento das derivadas, sugerindo que o refinamento baseado em XGBoost preserva melhor a tendência e a curvatura local do horizonte previsto em várias bases. Esse resultado é importante porque indica que o ganho do refinador não se limita à redução de nível ou deslocamento vertical da série: em diversos casos, a previsão refinada também passa a acompanhar melhor a forma da trajetória futura.

Os ganhos mais claros aparecem em bases como `nn5_daily_no_missing`, `tourism_monthly` e `tourism_quarterly`, nas quais a similaridade de primeira e segunda derivadas aumenta substancialmente. Isso sugere que a camada de refinamento conseguiu capturar padrões locais que o TimesFM, em regime *zero-shot*, não representou de forma suficiente. Quando ocorrem perdas, elas tendem a ser pequenas, frequentemente da ordem de poucos centésimos, sugerindo degradação limitada no alinhamento dinâmico. A principal exceção é `solar_weekly`, em que a queda é mais pronunciada. Uma explicação plausível é que essa base usa horizonte de apenas cinco pontos, tornando as similaridades de primeira e segunda derivadas sensíveis a pequenas flutuações locais. Nesse regime, os vetores de diferenças são muito curtos, e a diferenciação numérica pode amplificar ruídos ou pequenas variações de fase.

B. Métricas de Erro

A Tabela II mostra que o refinamento geralmente melhora a acurácia em relação ao TimesFM base. Em bases como `cif_2016`, `covid_deaths`, `tourism_monthly`, `tourism_quarterly` e `tourism_yearly`, as reduções de erro são expressivas.

Algumas bases apresentam comportamento misto. Em `australian_electricity_demand` e `nn5_weekly`, o refinamento piora levemente as métricas de erro, embora apresente ganhos em alinhamento de derivadas, sugerindo um possível compromisso entre erro ponto a ponto e fidelidade dinâmica. Em `pedestrian_counts`, há melhora em MAE e RMSE, mas não nas métricas livres de escala, indicando que o benefício não é uniforme quando o erro é normalizado por uma referência ingênua.



(a) Melhorias em similaridade de derivadas.

(b) Melhorias em métricas de erro.

Figure 1. Melhorias agregadas do Time BoostRer em relação ao TimesFM. À esquerda, melhorias em similaridade de derivadas. À direita, melhorias percentuais em métricas tradicionais de erro.

Table I
RESULTADOS DE SIMILARIDADE DE DERIVADAS POR BASE PARA TIMESFM (TFM) E TIME BOOSTRER. VALORES MAIORES INDICAM MELHOR ALINHAMENTO DE TENDÊNCIA E CURVATURA; O MELHOR VALOR DE CADA PAR ESTÁ EM NEGRITO.

	Similaridade da 1ª derivada		Similaridade da 2ª derivada	
Base	TFM	Time BoostRer	TFM	Time BoostRer
australian_electricity_demand	**0.0000**	**0.0000**	0.0054	**0.0631**
bitcoin_no_missing	0.0806	**0.2624**	0.0789	**0.2929**
cif_2016	0.4528	**0.4639**	**0.4900**	0.4261
covid_deaths	0.1830	**0.2295**	0.0548	**0.0854**
fred_md	**0.2295**	0.1925	**0.1439**	0.1337
hospital	**0.1181**	0.1077	0.1353	**0.1438**
nn5_daily_no_missing	0.0000	**0.7978**	0.0000	**0.7454**
nn5_weekly	0.1310	**0.2423**	0.1344	**0.2646**
pedestrian_counts	**0.1567**	0.1197	**0.1071**	0.0929
saugeenday	**0.0000**	**0.0000**	0.0000	**0.2153**
solar_weekly	**0.5850**	0.1340	**0.6322**	0.1501
tourism_monthly	0.0000	**0.7785**	0.0000	**0.5598**
tourism_quarterly	0.3163	**0.5089**	0.3155	**0.5271**
tourism_yearly	0.1996	**0.3575**	**0.3181**	0.2871
traffic_hourly	**0.0078**	0.0069	**0.0803**	0.0509

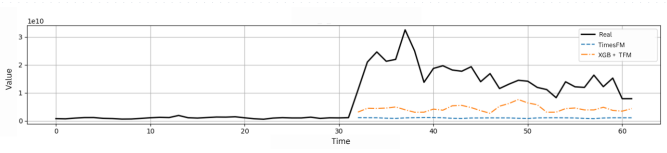


Figure 2. Exemplo em que o Time BoostRer acompanha melhor a trajetória alvo do que o TimesFM em direção e forma local.

O caso de bitcoin_no_missing exige cuidado adicional. As médias brutas são dominadas por uma série extrema com magnitude explosiva, o que comprime a diferença aparente entre TimesFM e Time BoostRer. Em uma análise complementar excluindo essa série, o refinamento reduz o MAE médio de $1{,}0107 \times 10^{12}$ para $6{,}9813 \times 10^{11}$, o RMSE médio de $1{,}0244 \times 10^{12}$ para $7{,}2007 \times 10^{11}$ e o SMAPE médio de $0{,}9871$ para $0{,}8907$. Esse resultado reforça que a leitura das médias agregadas precisa considerar sensibilidade a outliers de escala extrema.

C. Interpretabilidade

Os resultados de interpretabilidade indicam que o refinamento é guiado por um conjunto pequeno e estável de sinais de correção. Um padrão recorrente é que a previsão original

Table II
ERRO MÉDIO DE PREVISÃO POR BASE PARA TIMESFM (TFM) E TIME BOOSTRER. VALORES MENORES INDICAM MELHOR DESEMPENHO; O MELHOR VALOR DE CADA PAR ESTÁ EM NEGRITO.

	MAE		RMSE		SMAPE		MAE escalado	
Base	TFM	Time BoostRer	TFM	Time BoostRer	TFM	Time BoostRer	TFM	Time BoostRer
australian_electricity_demand	**587.9367**	632.0213	**737.1278**	751.1904	**0.1828**	0.2076	**0.9831**	1.1738
bitcoin_no_missing*	**5.244e+18**	5.272e+18	**5.271e+18**	5.298e+18	1.0399	**0.9500**	**0.9808**	1.5379
cif_2016	1616651.9227	**803566.1420**	1743553.6936	**876496.7288**	0.3122	**0.2027**	1.0961	**0.7328**
covid_deaths	1142.4835	**814.7973**	1152.1434	**824.0724**	0.5940	**0.3529**	0.9893	**0.7292**
fred_md	29375.6278	**23037.7423**	29422.3196	**23107.2076**	0.5245	**0.3474**	1.0115	**0.7765**
hospital	28.3108	**24.9974**	33.7261	**30.1522**	0.2187	**0.2006**	**0.9585**	0.9727
nn5_daily_no_missing	13.9449	**7.3033**	17.1455	**9.9619**	0.3910	**0.2008**	1.3170	**0.6897**
nn5_weekly	**18.6406**	19.6295	**21.8564**	22.9289	**0.1392**	0.1470	**1.0220**	1.1428
pedestrian_counts	484.7154	**481.8089**	605.9316	**551.7216**	**1.2830**	1.3159	**1.3981**	1.7643
saugeenday	81.3355	**80.4900**	97.0513	**96.0975**	1.2374	**1.2152**	0.9941	**0.9838**
solar_weekly	3035.3803	**2197.1004**	3167.5428	**2396.9263**	0.5157	**0.4038**	1.0817	**0.8273**
tourism_monthly	1893.2634	**923.4309**	2453.6419	**1208.7397**	0.5163	**0.2712**	1.5253	**0.7440**
tourism_quarterly	45300.4341	**27288.4554**	47023.5795	**29457.1328**	0.5717	**0.3530**	1.0263	**0.7614**
tourism_yearly	74130.1269	**51922.1834**	75580.2971	**53885.7060**	0.5943	**0.4027**	1.1309	**0.8507**
traffic_hourly	0.0374	**0.0352**	0.0446	**0.0419**	0.1246	**0.1173**	0.9814	**0.9063**

* Os erros médios de bitcoin_no_missing são fortemente influenciados por uma série extrema, o que reduz a interpretabilidade das médias brutas. Em análise complementar sem essa série, o refinamento reduz o MAE médio de $1{,}0107 \times 10^{12}$ para $6{,}9813 \times 10^{11}$, o RMSE médio de $1{,}0244 \times 10^{12}$ para $7{,}2007 \times 10^{11}$ e o SMAPE médio de 0,9871 para 0,8907.

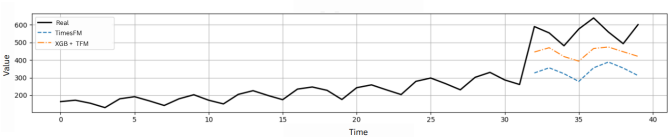


Figure 3. Exemplo de melhoria de erro: o Time BoostRer permanece mais próximo do alvo ao longo do horizonte, reduzindo diferença de nível e preservando a trajetória geral.

do TimesFM permanece entre os atributos mais importantes, indicando que a segunda etapa atua principalmente como recalibração, e não como substituição completa do modelo fundacional.

Em bases como traffic_hourly, fred_md e tourism_quarterly, a previsão base aparece entre os principais sinais, enquanto o índice do horizonte e as informações de contexto recente também contribuem significativamente. Isso sugere que o refinador captura vieses dependentes do horizonte e ajusta a previsão ao nível local atual. Por outro lado, em bases como hospital, nn5_daily_no_missing, solar_weekly e covid_deaths, os atributos de calendário ganham maior importância, indicando que a correção aprendida possui forte componente sazonal.

Esses resultados reforçam que o método aprende estratégias de correção adequadas a cada base, em vez de aplicar uma regra uniforme. Quando a previsão fundacional já captura boa parte da estrutura, o refinador tende a atuar como recalibrador local; quando efeitos sazonais são mais relevantes, ele passa a explorar atributos de calendário com maior intensidade. Assim, os ganhos observados não são apenas numéricos, mas também estruturalmente interpretáveis. Esse comportamento é desejável para uma monografia de cunho científico porque mostra que a camada adicional não funciona como uma caixa-preta arbitrária, mas como um mecanismo inspecionável de adaptação local.

D. Custo Computacional e Sustentabilidade

O método preserva a capacidade *zero-shot* do TimesFM e evita adaptação por gradiente no modelo fundacional. Na prática, o principal custo da pipeline é a geração repetida de previsões pelo TimesFM, necessária para construir a base de refinamento. Em agregação ponderada por série sobre 2.595 séries avaliadas, o tempo médio por série é de 16,24 segundos para geração de previsões pelo TimesFM, 5,97 segundos para a etapa de *boosting* e 22,21 segundos para a pipeline completa.

Esse resultado é central para a discussão de sustentabilidade. O refinamento adiciona custo incremental relativamente pequeno em comparação com a previsão *zero-shot* já necessária. Em vez de introduzir uma segunda etapa pesada de otimização, o método especializa as previsões por meio de uma camada tabular compacta. Como a maior parte do tempo já é consumida pela geração das previsões do TimesFM, a camada de *boosting* altera pouco o perfil computacional dominante da pipeline.

O contraste com *fine-tuning* torna essa vantagem mais clara. O TimesFM possui cerca de 200 milhões de parâmetros [1]; portanto, sua adaptação por *fine-tuning* exige otimização dentro da arquitetura do próprio modelo. Trabalhos recentes de adaptação financeira reportam pré-treinamento contínuo em dezenas de milhões de pontos e ajuste em GPU [9]. Mesmo alternativas parametrizadas eficientes, como LoRA [10], ainda exigem treinamento dentro do backbone do modelo. O Time BoostRer evita esse custo ao manter o modelo fundacional congelado e deslocar a adaptação para um modelo tabular pequeno, treinado sobre previsões já calculadas e atributos temporais compactos. Essa é a principal razão pela qual a abordagem pode ser descrita como uma estratégia de adaptação sustentável por projeto.

VII. Discussão e Conclusão

Este trabalho apresentou o Time BoostRer, uma estratégia leve, sustentável e explicável para refinar previsões *zero-shot* de séries temporais produzidas por um modelo fundacional congelado. A principal contribuição está em deslocar a adaptação do modelo fundacional para uma camada tabular externa, treinada com XGBoost a partir de atributos simples e interpretáveis. Em vez de substituir o modelo fundacional ou ajustar seus parâmetros, a proposta explora a previsão *zero-shot* como sinal inicial forte e aprende correções locais associadas ao horizonte, ao nível recente e ao calendário.

Os resultados mostram que o refinamento melhora métricas tradicionais de erro em grande parte das bases avaliadas e, em vários casos, também melhora a fidelidade de tendência medida por similaridade de derivadas. Esse resultado sugere que o método não apenas aproxima os valores previstos dos alvos, mas também pode melhorar a capacidade de seguir a forma local da trajetória. A análise de interpretabilidade indica que o refinador atua de forma coerente, combinando a previsão base com informações de horizonte, nível recente e calendário. Em alguns conjuntos, a previsão do TimesFM permanece como principal sinal; em outros, atributos temporais e sazonais ganham relevância, mostrando que a correção se adapta ao perfil da base.

A contribuição de sustentabilidade está na forma como a adaptação é implementada. O modelo fundacional permanece congelado, e a especialização local é realizada por uma camada compacta treinada apenas sobre janelas do regime mais recente. Isso reduz o custo de adaptação em relação a alternativas baseadas em *fine-tuning*, preserva a simplicidade da inferência *zero-shot* e mantém o processo de correção inspecionável. A abordagem, portanto, oferece uma ponte prática entre modelos fundacionais de grande escala e necessidades locais de previsão.

A discussão dos resultados também revela limitações importantes. Os experimentos foram executados com configuração única e sem média sobre múltiplas sementes. Além disso, a avaliação atual utiliza janelas de validação e teste específicas, e não um protocolo completo de *rolling-origin backtesting*. A métrica de similaridade de derivadas também pode ser instável em horizontes muito curtos, como observado em alguns resultados, pois a diferenciação numérica passa a depender de vetores pequenos e sensíveis a ruído. Outra limitação é que a análise de explicabilidade é descritiva, não causal; ela indica quais atributos foram usados pelo refinador, mas não permite concluir relações causais sobre a geração dos erros.

Como trabalhos futuros, pretende-se expandir a avaliação para protocolos de origem móvel, testar outros modelos fundacionais além do TimesFM, comparar refinamento direto do valor com refinamento explícito de resíduos, incluir estimativas de incerteza, medir consumo energético de forma direta e explorar mecanismos de seleção automática de atributos e hiperparâmetros. Também seria relevante avaliar o comportamento do método em conjuntos com múltiplas frequências, em séries multivariadas e em cenários de atualização contínua, nos quais a identificação de regimes recentes pode ter impacto ainda maior.

References

[1] A. Das, W. Kong, R. Sen, and Y. Zhou, "A decoder-only foundation model for time-series forecasting," in *Proceedings of the 41st International Conference on Machine Learning*, 2024.

[2] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

[3] R. Killick, P. Fearnhead, and I. A. Eckley, "Optimal detection of changepoints with a linear computational cost," *Journal of the American Statistical Association*, vol. 107, no. 500, pp. 1590–1598, 2012.

[4] R. Godahewa, C. Bergmeir, G. I. Webb, R. J. Hyndman, and P. Montero-Manso, "Monash time series forecasting archive," in *Advances in Neural Information Processing Systems: Datasets and Benchmarks*, 2021.

[5] A. F. Ansari *et al.*, "Chronos: Learning the language of time series," *Transactions on Machine Learning Research*, 2025.

[6] K. Rasul *et al.*, "Lag-Llama: Towards foundation models for probabilistic time series forecasting," in *Proceedings of the 27th International Conference on Artificial Intelligence and Statistics*, 2024.

[7] G. Woo, C. Liu, A. Kumar, C. Xiong, S. Savarese, and D. Sahoo, "Unified training of universal time series forecasting transformers," arXiv preprint arXiv:2402.02592, 2024.

[8] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," in *International Conference on Learning Representations*, 2023.

[9] X. Fu, M. Hirano, and K. Imajo, "Financial fine-tuning a large time series model," arXiv preprint arXiv:2412.09880, 2024.

[10] E. J. Hu *et al.*, "LoRA: Low-rank adaptation of large language models," in *International Conference on Learning Representations*, 2022.

[11] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems*, 2017.

[12] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.