Otimização de Parâmetros de Buffer Pool com Aprendizado de Máquina em Ambientes Não Transacionais

Eduardo P. Mendizabal[1], **Geraldo P. Rocha Filho**[2], **Aleteia Araujo**[1]

[1]Instituto de Ciências Exatas – Departamento de Ciências da Computação
Universidade de Brasília (UNB) – Brasília – DF – Brasil

[2]Departamento de Ciencias Exatas e Tecnológicas
Universidade Estadual do Sudoeste da Bahia (UESB) – Vitória da Conquista – BA – Brasil.

220005362@aluno.unb.br, geraldo.rocha@uesb.edu.br, aleteia@unb.br

Abstract. *A parametrização eficiente de Sistemas Gerenciadores de Banco de Dados (SGBD) é condição essencial para assegurar níveis de serviço estáveis e mitigar picos de latência em ambientes corporativos críticos. Este trabalho apresenta uma solução automatizada de configuração do buffer pool, baseada em uma solução que emprega Aprendizado de Máquina (AM) e Otimização Bayesiana. Assim, aplicou-se Análise Fatorial Exploratória (AFE) associada a K-Means para reduzir a dimensionalidade e eleger métricas representativas. Em seguida, empregou-se regressão LASSO para filtrar variáveis determinantes. Por fim, utilizou-se Regressão com Processo Gaussiano (GPR) e políticas de aquisição bayesiana para gerar recomendações de configuração. Avaliada em um repositório de dados de grande instituição financeira, a solução proporcionou diminuição média de 18% na latência máxima de I/O síncrono. Os resultados evidenciam reprodutibilidade analítica, transparência nas decisões e adaptabilidade a cenários não transacionais, oferecendo uma solução extensível para governança proativa de SGBD em produção.*

1. Introdução

O desempenho de Sistemas Gerenciadores de Banco de Dados (SGBD) é fator crítico para garantir eficiência operacional, alta disponibilidade e baixa latência em ambientes corporativos exigentes. Contudo, a configuração manual de dezenas de parâmetros interdependentes, baseada em tentativa e erro, revela-se insuficiente diante da variabilidade das cargas de trabalho e da elevada dimensionalidade do espaço de configuração. Parâmetros subótimos podem resultar em aumentos expressivos de latência de I/O, queda de *throughput* e utilização ineficiente de recursos, gerando riscos operacionais e ônus financeiros [Aken et al. 2017].

Assim, embora soluções automatizadas para configuração de parâmetros já tenham sido amplamente validadas em cenários transacionais, sua aplicação em contextos não transacionais, como silos de desenvolvimento e *Data Warehouses* em grandes instituições financeiras, ainda é escassa. Nesse contexto, este artigo apresenta a adaptação e validação de uma solução originalmente projetada para cargas transacionais, comprovando sua eficácia em um ambiente real não transacional, representado por um silo de dados de uma grande instituição financeira.

A solução combina pré-processamento robusto de dados — incluindo limpeza de inconsistências, imputação de valores ausentes e remoção de *outliers* em um conjunto de 178.215 registros de 26 *buffer pools*, e uma análise exploratória temporal para capturar padrões diurnos e noturnos. Em seguida, emprega-se redução de dimensionalidade via Análise Fatorial Exploratória (AFE) associada à regressão *Least Absolute Shrinkage and Selection Operator* (LASSO) para selecionar as variáveis mais influentes, acompanhada de clusterização K-Means

para identificar métricas representativas. A etapa final utiliza Regressão com Processo Gaussiano (GPR) integrada à otimização bayesiana por meio de diferentes funções de aquisição, gerando recomendações de configuração que minimizam a latência síncrona de I/O.

A aplicação dessa solução ao ambiente não transacional do silo de desenvolvimento resultou em redução média de 18% na latência máxima de I/O síncrono, evidenciando a capacidade do método em capturar relações complexas entre configuração e desempenho. Esses resultados reforçam a reprodutibilidade científica, a transparência analítica e a adaptabilidade contextual da solução, oferecendo uma base extensível para gestão proativa e automatizada de SGBD em diversos cenários corporativos.

O restante deste artigo está organizado da seguinte forma. A Seção 2 apresenta os trabalhos relacionados, enquanto a Seção 3 descreve os fundamentos teóricos necessários para compreender a solução proposta. A Seção 4 apresenta a solução proposta e como foi implantado. A Seção 5 apresenta os resultados obtidos para validar a solução. Por fim, a Seção 6 apresenta as conclusões e os trabalhos futuros.

2. Trabalhos Relacionados

A otimização automática de parâmetros em SGBD tem sido amplamente discutida na literatura, com abordagens que vão desde métodos heurísticos empíricos [Storm and Surendra 2006, Vasyliev 2024, Ansel et al. 2014] até implementações sofisticadas de Aprendizado de Máquina (AM) e Inteligência Artificial [Aken et al. 2017, Cai et al. 2022, Trummer 2021]. A partir de 2006, trabalhos como STMM [Storm and Surendra 2006], PGTune [Vasyliev 2024] e OpenTuner [Ansel et al. 2014], o ajuste de parâmetros era conduzido com base na experiência do administrador ou em experimentos isolados de *workload*, estratégia que se mostra limitada diante da complexidade e dinâmica de ambientes corporativos modernos.

Para reduzir o espaço de métricas antes da otimização propriamente dita, estudos clássicos adotaram técnicas de seleção de AFE e regressão LASSO [Aken et al. 2017], enquanto outros empregaram PCA e clusterização K-Means para agrupar métricas correlacionadas [Cai et al. 2022]. Trummer et al. [Trummer 2021] estenderam essa abordagem ao utilizar TF-IDF para caracterizar padrões de operação em diferentes tipos de carga de trabalho, facilitando a escolha de variáveis mais representativas.

No núcleo das soluções de otimização, a otimização bayesiana consolidou-se como técnica de referência. Processos GPR associados à aquisição *Expected Improvement* foram aplicados com sucesso em OtterTune [Aken et al. 2017], Restune [Zhang et al. 2021] e CGPTuner [Cereda et al. 2021], demonstrando equilíbrio entre custo computacional e qualidade das recomendações. Em paralelo, abordagens baseadas em aprendizado profundo empregam Redes Neurais Profundas e CNNs para estimar desempenho sem executar exaustivamente todas as cargas de trabalho [Tan et al. 2019, Aken et al. 2021], enquanto métodos de aprendizado por reforço modelam o problema como uma sequência de decisões, embora demandem volume elevado de interações e supervisão especializada [Zhao et al. 2023].

Adicionalmente, algumas propostas exploram técnicas de transferência de conhecimento para reaproveitar históricos de otimização e reduzir o tempo de convergência em novos ambientes. Kanellis e Llamatune [Kanellis et al. 2022] demonstraram mapeamentos entre cargas de trabalho similares, e Zhang et al. [Zhang et al. 2022] empregaram modelos de transferência para acelerar ajustes iniciais em sistemas inéditos.

Durante a pesquisa bibliográfica, constatou-se que poucas pesquisas validam suas soluções em cenários não transacionais de larga escala, especialmente no contexto de *buf-*

fer pool em *mainframes* como o IBM DB2 for z/OS, ou quantificam a incerteza nas recomendações. A abordagem proposta neste trabalho preenche essas lacunas ao integrar, em um ambiente real de produção de grande instituição financeira, pré-processamento avançado, análise temporal, redução de dimensionalidade, modelagem probabilística e otimização bayesiana numa única solução reprodutível e adaptável. Além disso, a solução proposta foi aplicada em um ambiente real não transacional.

3. Fundamentação Teórica

Esta seção apresenta os fundamentos de aprendizado supervisionado, não supervisionado, redução de dimensionalidade, e otimização bayesiana que embasam nossa solução de configuração automática de parâmetros de SGBD. Compreender esses elementos é essencial para interpretar os resultados e assegurar a reprodutibilidade da abordagem em diferentes cenários de produção.

No aprendizado supervisionado, modelos são treinados com dados rotulados para mapear entradas em saídas conhecidas [Vyawahare 2022, Shaveta 2023]. Essa técnica permite capturar relações complexas entre múltiplas variáveis de configuração e métricas de desempenho de I/O em SGBD. Para predição de métricas contínuas, a Regressão Linear via Mínimos Quadrados Ordinários (OLS) é frequentemente empregada pela sua simplicidade e fácil interpretação dos coeficientes [Joshi et al. 2020, Li et al. 2020]. Porém, em ambientes com dezenas ou centenas de parâmetros interdependentes, o OLS apresenta alta variância e dificuldade de selecionar variáveis relevantes [Casella and Berger 2002]. O LASSO supera essas limitações ao incluir um termo de penalização L1 na função de custo, induzindo esparsidade e automaticamente removendo coeficientes próximos de zero [Tibshirani 1996]. Essa regularização reduz o risco de *overfitting* e destaca apenas as variáveis mais influentes para o desempenho do *buffer pool*, como indicado na Equação (1):

$$\hat{\beta}_{\text{lasso}} = \arg\min_{\beta} \left\{ \frac{1}{2n} \|y - X\beta\|_2^2 + \lambda \|\beta\|_1 \right\}. \tag{1}$$

No aprendizado não supervisionado, o *K*-means agrupa instâncias em k *clusters* com base na similaridade dos vetores de métricas de desempenho, minimizando a soma dos quadrados *intra-cluster* [Lee 2019, Duarte and Ståhl 2019]. Essa clusterização facilita a identificação de perfis de carga de trabalho semelhantes, e reduz o espaço de busca ao selecionar centroides representativos, como mostrado na Equação (2):

$$J = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2. \tag{2}$$

Técnicas como o método do cotovelo e análise da silhueta orientam a escolha de k, garantindo equilíbrio entre detalhamento e generalização dos agrupamentos [Sullivan 2012]. A AFE visa revelar estruturas latentes que explicam correlações entre variáveis de configuração e métricas de desempenho [Yong and Pearce 2013]. Inicialmente, aplicam-se os testes de esfericidade de Bartlett e KMO para verificar a adequação dos dados à fatorabilidade [Trendafilov and Hirose 2023]. Em seguida, retêm-se apenas fatores com autovalores superiores a 1 (critério de Kaiser) e realiza-se rotação Varimax para maximizar a interpretabilidade das cargas fatoriais, permitindo que cada fator represente um conjunto coeso de métricas.

Na otimização bayesiana, modelou-se a métrica-alvo como um Processo Gaussiano (*GP*), que fornece estimativas pontuais e de variância preditiva, essenciais para mensurar

confiança nas recomendações [Rasmussen and Williams 2005]. A cada iteração, funções de aquisição como *Expected Improvement*, *Probability of Improvement* e *Upper Confidence Bound* balanceiam exploração de regiões desconhecidas do espaço de configuração e exploração de áreas já promissoras [Shahriari et al. 2016]. Esse procedimento é particularmente eficaz em cenários de produção, onde cada teste de configuração implica custo de tempo e recursos computacionais.

Em conjunto, essas técnicas formam a base teórica de nossa solução, habilitando a redução do espaço de parâmetros, a seleção dos ajustes mais impactantes e a geração de recomendações robustas para o *buffer pool* de SGBD em ambientes críticos. Essa integração garante não apenas desempenho aprimorado, mas também transparência analítica e reprodutibilidade dos experimentos.

4. Metodologia Proposta

A metodologia proposta para a otimização automática de parâmetros em SGBD via AM é estruturada em uma solução sequencial e modular, projetado para garantir a reprodutibilidade e eficácia na identificação de configurações ideais. Este fluxo de trabalho abrange desde a preparação inicial dos dados até a recomendação de parâmetros otimizados, conforme ilustrado pelas etapas subsequentes.

O processo tem início com a etapa de Pré-processamento de Dados, voltada à limpeza, padronização e integração de métricas de desempenho e parâmetros de configuração, de modo a formar uma amostra coesa para análise. Em ambientes não transacionais, essa etapa se mostra significativamente mais rigorosa, exigindo uma remoção mais criteriosa e agressiva de *outliers*, dada a variabilidade e a natureza menos padronizada das cargas de trabalho.

Para identificar as métricas relevantes é utilizada duas técnicas combinadas. A AFE é utilizada para compreender o comportamento das métricas e reduzir sua dimensionalidade seguido da redução a partir do agrupamento de métricas via algoritmo KMeans sobre os *loadings* da AFE para consolidar as métricas em grupos representativos. Além disso, para identificar e ordenar os parâmetros de configuração mais influentes, a técnica de seleção via LASSO é aplicada, gerando uma lista otimizada de parâmetros com maior impacto na métrica-alvo.

Com as métricas e parâmetros já ajustados, inicia-se o modelo preditivo a partir da regressão com GPR e Otimização, no qual o modelo GPR é treinado para prever o desempenho, e as estratégias de aquisição são empregadas para recomendar configurações de parâmetros que visam a melhoria da métrica-alvo.

Cada uma dessas etapas é detalhada nas seções seguintes, com ênfase na sua implementação e nos resultados obtidos. A abordagem modular permite que cada componente seja avaliado e aprimorado independentemente, contribuindo para a robustez geral do sistema. A metodologia é concebida para ser totalmente reproduzível, com todas as etapas explicitamente definidas e baseadas nos dados fornecidos, em conformidade com os princípios da pesquisa científica.

4.1. Pré-processamento e Análise Exploratória dos Dados

A qualidade e a estrutura dos dados de entrada são determinantes para o sucesso de qualquer modelo de AM. Esta seção detalha as etapas cruciais de preparação e análise inicial dos dados, que estabelecem a base para as fases subsequentes de redução de dimensionalidade e modelagem preditiva.

A fase de pré-processamento engloba todas as operações necessárias para preparar os dados brutos de métricas de desempenho e parâmetros de configuração. Após o carregamento, é realizada a limpeza e conversões para a geração da amostra. Realiza-se a remoção de caracteres especiais e a conversão de valores numéricos e temporais. Além disso, é necessária a identificação de campos do tipo `object` que deverão ser categorizados numericamente ou não poderão ser utilizados na otimização e devem ser descartados.

Um aspecto crítico da limpeza de dados envolve a remoção de colunas que não contribuem para a análise ou que podem introduzir redundância. Quatro colunas com variância zero foram identificadas e removidas, uma vez que a ausência de variabilidade nessas colunas significa que elas não fornecem informações discriminatórias para o modelo. Adicionalmente, duas colunas são removidas devido à sua correlação absoluta ser superior a 99,5%. A remoção de atributos altamente correlacionados é uma prática recomendada para mitigar a multicolinearidade, que pode levar a modelos instáveis e com interpretação comprometida. A Tabela 1 apresenta as métricas removidas por não possuírem variabilidade considerável.

Tabela 1. Métricas removidas por baixíssima variabilidade.

Métrica	Motivo da Remoção
Pending Time: Device Busy Delay (ms)	Variância igual a zero
Changed Pages to be Written (GiB)	Variância igual a zero
Decrypted Read Throughput (MB/s)	Variância igual a zero
Encrypted Write Throughput (MB/s)	Variância igual a zero
Getpages Hits in Buffer Pool (Getpages/s)	Correlação maior que 99,5%
Random Read (I/Os/s)	Correlação maior que 99,5%

A amostra final é construída pela junção interna dos *DataFrames* de métricas e parâmetros, utilizando a coluna referente ao nome do *buffer pool* como chave. Após a junção, 446 registros contendo valores nulos (0,25% da amostra total) foram removidos para garantir a completude dos dados para análises subsequentes. A amostra geral resultante desse processo tem uma dimensão de 178.215 linhas e 29 colunas, e com a possibilidade de otimização de 26 *buffer pools*. A Tabela 2 apresenta um resumo da amostra geral após o pré-processamento.

Tabela 2. Resumo da amostra geral após pré-processamento.

Característica da Amostra	Valor
Registros Finais	178.215
Registros Removidos	446 (0,25%)
Parametros e Métrica Finais	29
Métricas Removidas no Pré-processamento	6 (17,14%)
Quantidade de Buffer Pools Otimizavéis	26

Para caracterizar o comportamento da métrica-alvo *Maximum Synchronous I/O Delay (ms)* (`MAX_SYNC_IO_DLY_MS`), conduziu-se uma análise exploratória centrada no 99º percentil por hora e por *buffer pool*, visando identificar picos de latência e padrões sazonais. Em seguida, comparou-se o desempenho entre os turnos diurno (07:00–19:00) e noturno (19:00–07:00), avaliando o impacto das variações de carga de trabalho sobre a latência do SGBD. Os resultados

revelaram diferença expressiva no 99º percentil de I/O síncrono entre os dois turnos em diversos *buffer pools*, refletindo o uso concentrado no período diurno típico de ambientes não transacionais. Observou-se redução superior a 99 % na volumetria e na métrica durante a noite em determinados pools, indicando que a segmentação de carga por turno, fundamental em cenários transacionais, não foi relevante para o ajuste final da configuração neste contexto.

4.2. Redução de Dimensionalidade e Agrupamento de Métricas

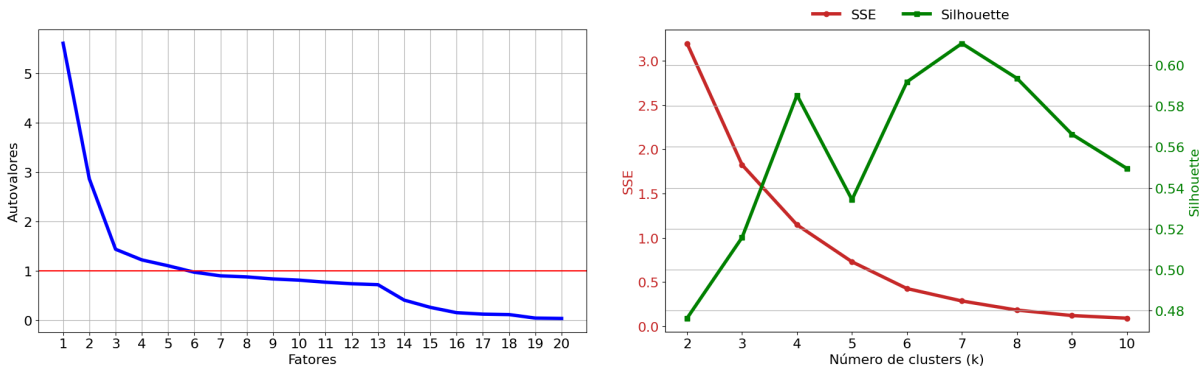
A grande quantidade de métricas de desempenho em SGBDs pode dificultar a análise e o desenvolvimento de modelos preditivos. Esta seção aborda a complexidade das métricas, aplicando técnicas para reduzir sua dimensionalidade e identificar agrupamentos significativos, tornando o conjunto de dados mais gerenciável e interpretável.

A AFE é empregada para reduzir a dimensionalidade do conjunto de métricas numéricas, identificando fatores latentes que explicam a covariância entre as variáveis observadas.O processo inicia-se com a extração das 20 métricas numéricas válidas da amostra geral.

A adequação dos dados para a AFE é avaliada por meio de dois testes estatísticos cruciais: o Teste de Esfericidade de Bartlett e o índice Kaiser-Meyer-Olkin (KMO).O Teste de Esfericidade de Bartlett verifica se a matriz de correlação observada é significativamente diferente de uma matriz identidade, o que indicaria que as variáveis são correlacionadas e, portanto, adequadas para a análise fatorial. Os resultados do teste ($\chi^2 = 2297474,62, p = 0,0000$) indicam um p-valor extremamente baixo, confirmando que as métricas possuem correlações significativas e são apropriadas para a AFE.

O índice KMO mede a adequação da amostra, indicando a proporção da variância nas variáveis que pode ser explicada por fatores subjacentes. Um KMO do modelo de 0,83 é considerado excelente (valores acima de 0,6 são geralmente aceitáveis), reforçando a adequação dos dados para a análise fatorial.

A determinação do número ótimo de componentes (fatores) é realizada utilizando o critério de Kaiser, que sugere reter fatores com autovalores maiores do que 1. Neste caso, a Figura 1(a) sugere a retenção de 5 fatores. Após a determinação do número de fatores, a análise fatorial é realizada com rotação Varimax. A rotação Varimax é aplicada para simplificar a interpretação dos fatores, maximizando a variância das cargas quadradas de cada fator, o que resulta em fatores mais distintos e mais facilmente interpretáveis.



(a) Scree Plot dos Autovalores

(b) Avaliação de k: SSE e Silhouette

Figura 1. Comparativo entre o Scree Plot e a avaliação de clustering.

Com base nos *loadings* da AFE, a clusterização KMeans é aplicada para identificar agrupamentos de métricas com comportamento correlacionado, visando selecionar uma métrica

representativa para cada agrupamento e, assim, simplificar ainda mais a análise. A avaliação do número ótimo de agrupamentos (k) é uma etapa crítica. São utilizados dois critérios: o critério de Pham (PS) e o Coeficiente de Silhueta. A análise revelou que o "Melhor k (Pham): 6"e o "Melhor k (Silhouette): 7", conforme a Figura 1.

Para a execução do KMeans, o valor de k=6 (baseado no critério de Pham) é utilizado. Para cada um dos 6 agrupamentos identificados pelo KMeans, a métrica mais próxima do centroide do agrupamento é selecionada como a métrica representativa. A Tabela 3 lista as métricas representativas selecionadas para cada agrupamento.

Tabela 3. Métricas representativas por agrupamento.

Agrupamento	Métrica Representativa
0	Asynchronous I/O Delay (ms)
1	Read and Write Throughput (MB/s)
2	Synchronous I/Os (I/Os/s)
3	Pending Time: Command Response Delay (ms)
4	Pages Used in Buffer Pool (GiB)
5	Synchronous I/O Delay (ms)

4.3. Seleção de Parâmetros de Configuração via Regressão LASSO

A seleção dos parâmetros de configuração mais influentes na métrica-alvo é uma etapa crucial para a otimização de SGBDs. Para este fim, a regressão LASSO que é uma técnica conhecida por sua capacidade de realizar seleção de *features* ao penalizar a magnitude dos coeficientes, é empregada, potencialmente zerando os de menor importância.

A análise inicia com a execução do modelo LASSO, que calcula o caminho de regularização, fornecendo os valores de *alphas* (forças de regularização) e os *coefs* (coeficientes correspondentes para cada parâmetro em diferentes valores de *alpha*) conforme observado na Figura 2. Assim, com base no "Passo de Ativação", que indica em que ponto ao longo do caminho de regularização o coeficiente de um parâmetro se torna significativamente diferente de zero, os parâmetros são classificados quanto à sua importância. A Tabela 4 apresenta o resumo da análise LASSO, incluindo a classificação e a decisão final (manter ou remover) para cada parâmetro.

Tabela 4. Classificação de caminho LASSO.

Parâmetro	Passo de Ativação	Classificação de Importância	Decisão
VPSEQT	1	Precoce	Manter
VPUSE	4	Precoce	Manter
VPPSEQT	20	Precoce	Manter
VDWQT	32	Moderada	Manter
VPMIN	48	Moderada	Manter
VPMAX	51	Tardia	Manter
DWQT	55	Tardia	Manter

O resultado da Análise de Regressão LASSO foi segmentado em três categorias, conforme o número de iterações necessárias para ativação de cada parâmetro: *precoce* ($\leq$

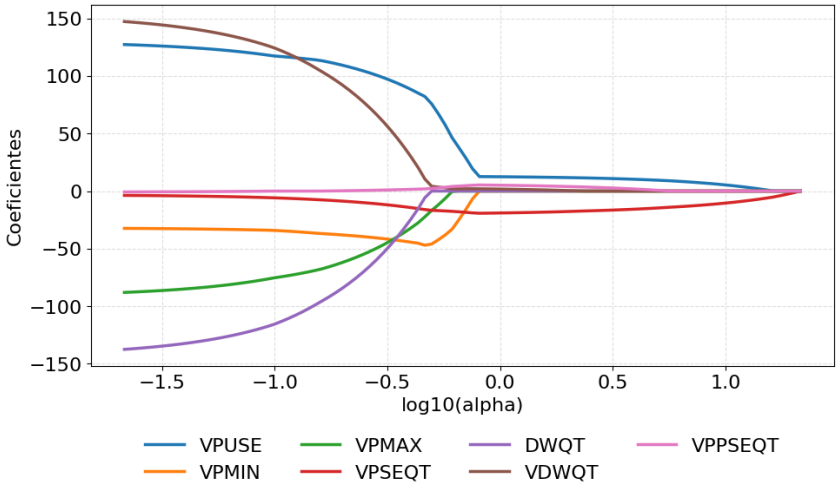


Figura 2. Caminho dos Coeficientes - LASSO PATH

25 iterações), *moderada* (26–50 iterações) e *tardia* ($>$ 50 iterações). A priorização dos parâmetros de ativação precoce maximiza o impacto operacional, reduzindo a variabilidade das intervenções e facilitando a rastreabilidade causa-efeito em ambientes de banco de dados. Embora o LASSO forneça um *ranking* objetivo de importância, a seleção final cabe ao DBA, garantindo flexibilidade na definição do escopo de ajustes. Para este experimento, optou-se por manter os sete parâmetros iniciais, equilibrando robustez analítica e governança de mudanças.

4.4. Regressão com Processo Gaussiano (GPR) e Otimização

Esta seção detalha a aplicação do modelo de GPR para prever o comportamento da métrica-alvo e as estratégias de aquisição utilizadas para recomendar configurações de parâmetros otimizadas. O GPR é uma abordagem não-paramétrica que fornece não apenas previsões pontuais, mas também uma medida de incerteza associada a essas previsões, o que é valioso em contextos de otimização.

A construção da amostra ajustada é a etapa final de preparação de dados, antes do treinamento do modelo de predição. Este processo consolida os resultados das fases anteriores de redução de dimensionalidade e seleção de *features*, combinando as métricas representativas e os parâmetros de configuração selecionados em um único *DataFrame* otimizado. Portanto, a amostra ajustada inclui apenas as colunas essenciais para a modelagem: as colunas temporal e de categorização do *buffer pool*, as seis métricas representativas identificadas pelo KMeans, e os sete parâmetros de configuração mantidos pelo LASSO. A amostra ajustada final tem uma dimensão de 178.215 linhas e 16 colunas.

Em ambientes não transacionais, a filtragem de *outliers* revelou-se crítica e deve ser continuamente refinada. Ao contrário de cenários transacionais de missão crítica, com fluxos de trabalho padronizados e objetivos bem definidos, *data warehouses* e ambientes de desenvolvimento (como o adotado neste experimento) costumam gerar valores extremos decorrentes de testes e falhas na modelagem, os quais não refletem as necessidades operacionais reais.

A escolha do método de remoção de outliers foi fundamentada em uma avaliação comparativa entre abordagens clássicas, incluindo *Z-Score* ($\pm 3\sigma$), *Interquartile Range* (IQR), Percentil (1-99%) e a combinação Percentil+IQR. Os resultados demonstraram que, embora o método Z-Score tenha removido apenas 0,22% dos registros e mantido alta variância residual, e o Percentil (1-99%) tenha eliminado 1,81%, ambos se mostraram pouco eficazes na redução

de variabilidade indesejada. O método IQR isolado apresentou maior impacto (14,22% de remoção), mas ainda preservou valores extremos. A estratégia combinada Percentil+IQR foi escolhida por proporcionar o melhor equilíbrio entre rigor estatístico e retenção de dados relevantes, removendo 15,43% dos registros e reduzindo a variância pós-limpeza para $73,21$, valor significativamente inferior aos demais métodos avaliados. Assim, este procedimento resultou em uma amostra final mais homogênea e adequada para as etapas subsequentes de modelagem, totalizando 150.712 registros após a limpeza.

Para reduzir o custo computacional do GPR, intensivo em grandes *datasets*, adotou-se amostragem de 20.000 registros (11,22% do conjunto original) para o treinamento. Observou-se desempenho similar com amostras entre 20.000 e 40.000 registros, o que indica potencial para metodologias de amostragem avançadas que capturem a representatividade sem processar toda a carga de trabalho. Por fim, os dados foram divididos em conjuntos de treino e validação, reservando 20% para validação, o que resultou em 16.000 registros de treino e 4.000 de validação, atendendo aos requisitos de generalização do modelo.

O modelo de Regressão com Processo Gaussiano foi implementado na biblioteca `gpytorch`, por meio da classe `ExactGPModel`. Inicialmente, a função de média foi definida como constante e foram avaliadas diferentes funções de covariância: `RBFKernel`, `MaternKernel` ($\nu = 2{,}5$), `SpectralMixtureKernel`, e combinações entre elas. O treinamento utilizou o otimizador Adam com taxa de aprendizado inicial de $0{,}01$ ao longo de 200 épocas, com monitoramento contínuo do desempenho no conjunto de validação. Após avaliação sistemática de todas as alternativas, o melhor desempenho foi obtido com o `MaternKernel` ($\nu = 2{,}5$), que atingiu coeficiente de determinação $R^2 = 20{,}3\%$ na validação, superando as demais abordagens testadas. Assim, a configuração baseada no kernel de Matérn foi mantida nas etapas posteriores por apresentar maior robustez e acurácia no cenário avaliado.

As estratégias de aquisição são componentes fundamentais na otimização bayesiana, utilizadas para guiar a busca por novas configurações de parâmetros que prometem a maior melhoria na função objetivo.Três estratégias são implementadas e avaliadas: *Expected Improvement* (EI), *Probability Improvement* (PI) e *Upper Confidence Bound* (UCB).

Assim, utilizou-se estratégia de otimização, iterando sobre cada *buffer pool* individualmente para encontrar as configurações de parâmetros otimizadas. Para cada um, um conjunto de 2.000 amostras de novas configurações candidatas é gerado, e as estratégias de aquisição são aplicadas para selecionar a configuração mais promissora ao longo de 20 iterações. Escolhendo a estratégia de aquisição de melhor desempenho.

5. Resultados

Ao final do processo de treinamento, o modelo foi avaliado no conjunto de validação, apresentando alguns indicadores de desempenho. O erro quadrático médio (RMSE) foi de 0,893 ms, sugerindo que, em média, o desvio padrão dos resíduos é inferior a 1 ms. O erro médio absoluto (MAE) alcançou 0,641 ms. O coeficiente de determinação (R^2) foi de 0,203, revelando que o modelo explica apenas 20,3% da variância da métrica-alvo, o que reforça a necessidade de aprimoramento na modelagem dos padrões. Por fim, a densidade preditiva negativa (NLPD) resultou em 1,309, sinalizando calibração moderada na distribuição preditiva e possíveis oportunidades de ajustes na modelagem de incerteza.

A Tabela 5 apresenta uma visão consolidada dos resultados da otimização para cada *buffer pool*, incluindo as configurações originais dos parâmetros (Orig.), as configurações re-

comendadas pelo modelo (Rec.), e os respectivos ganhos observados na métrica-alvo.

Tabela 5: Resultados da aquisição de parâmetros por *buffer pool*.

Buffer	**VPMIN**	**VPUSE**	**VPMAX**	**VPSEQT**	**VDWQT**	**DWQT**	**VPPSEQT**	**Ganho**
BP0 Ori.	10000	200000	200000	10	5	30	50	–
BP0 Rec.	164427	273045	480412	30	71	76	98	2%
BP1 Ori.	1048576	2097152	2097152	30	5	30	50	–
BP1 Rec.	59552	59552	649657	31	23	11	40	30%
BP11 Ori.	10000	10000	200000	80	0	30	50	–
BP11 Rec.	133589	133589	469819	26	32	14	4	10%
BP12 Ori.	10000	200000	200000	10	5	30	50	–
BP12 Rec.	208077	208077	0	10	91	30	19	6%
BP16K0 Ori.	2500	50000	50000	10	0	30	50	–
BP16K0 Rec.	14326	14326	489860	51	36	10	10	8%
BP16K1 Ori.	2500	50000	50000	80	1	35	50	–
BP16K1 Rec.	151033	151033	228912	35	51	35	12	2%
BP16K2 Ori.	2500	50000	50000	10	0	35	50	–
BP16K2 Rec.	21323	108180	740040	10	61	35	4	8%
BP2 Ori.	10000	200000	200000	10	5	30	50	–
BP2 Rec.	0	12357	541115	36	17	22	16	10%
BP23 Ori.	5000	5000	262144	30	0	50	50	–
BP23 Rec.	21191	64622	64622	11	12	32	34	16%
BP26 Ori.	5000	65536	262144	30	0	10	50	–
BP26 Rec.	200591	384074	512948	84	0	33	2	2%
BP3 Ori.	10000	200000	200000	30	5	10	50	–
BP3 Rec.	18880	18880	467743	30	4	29	14	16%
BP32K Ori.	10000	25000	100000	80	0	30	50	–
BP32K Rec.	0	5000	192380	40	12	30	20	6%
BP32K1 Ori.	10000	100000	100000	80	0	30	50	–
BP32K1 Rec.	0	79157	79157	39	12	30	9	6%
BP32K2 Ori.	100000	105000	500000	10	5	30	50	–
BP32K2 Rec.	96311	96311	98562	30	24	24	1	2%
BP32K7 Ori.	10000	76288	540000	99	85	90	50	–
BP32K7 Rec.	201727	333527	333527	91	0	33	8	10%
BP32K9 Ori.	5000	16384	262144	30	0	10	50	–
BP32K9 Rec.	100971	100971	550356	65	0	35	4	4%
BP4 Ori.	10000	154000	200000	30	5	10	50	–
BP4 Rec.	119738	119738	0	33	71	32	2	2%
BP5 Ori.	10000	10000	200000	80	5	87	50	–
BP5 Rec.	124420	581875	581875	34	0	87	8	4%
BP6 Ori.	10000	156000	200000	30	5	23	50	–
BP6 Rec.	192518	192518	215652	30	17	23	12	0%
BP7 Ori.	0	86000	0	99	85	90	50	–
BP7 Rec.	0	45608	408057	82	91	35	10	8%
BP8 Ori.	10000	200000	200000	30	5	10	50	–
BP8 Rec.	0	345216	400856	56	0	35	17	0%
BP8K0 Ori.	5000	48000	100000	80	0	31	50	–
BP8K0 Rec.	0	5000	305880	34	13	31	57	2%
BP8K1 Ori.	5000	100000	100000	80	0	34	50	–

Continua na próxima página

Continuação da Tabela 5

Buffer	VPMIN	VPUSE	VPMAX	VPSEQT	VDWQT	DWQT	VPPSEQT	Ganho
BP8K1 Rec.	0	60928	60928	10	14	34	21	22%
BP8K2 Ori.	5000	5000	100000	80	0	30	50	–
BP8K2 Rec.	8314	8314	584042	90	61	30	2	8%
BP9 Ori.	0	10000	0	10	0	30	50	–
BP9 Rec.	539287	49840	667579	74	0	38	12	18%

6. Conclusões e Trabalhos Futuros

Este estudo apresentou uma solução para otimização automática de parâmetros em SGBD com uso de AM. A solução processa dados brutos, reduz a dimensionalidade de métricas e parâmetros e aplica Regressão com GPR para sugerir configurações otimizadas. As etapas de pré-processamento mostraram-se eficazes, e a redução de dimensionalidade com AFE e regressão LASSO foi validada. O KMeans identificou métricas representativas, reduzindo a complexidade da análise. A limpeza de amostras para remoção de *outliers*, aliada à análise exploratória, revelou padrões relevantes, como variações diurnas e noturnas na latência de I/O síncrono, evidenciando o comportamento dinâmico das cargas. A regressão LASSO mostrou que todos os parâmetros influenciam a métrica-alvo, indicando alta interdependência. Embora o modelo GPR tenha convergido, apresentou baixo R^2, alinhado a estudos que desaconselham seu uso em ambientes não transacionais, reforçando a hipótese de que fatores externos comprometem sua predição. A heterogeneidade dos ganhos entre *buffer pools* ressalta o caráter contextual do problema e a necessidade de estratégias adaptativas. Ainda assim, a aplicação do fluxo proposto resultou em ganhos de desempenho em ambiente real, demonstrando sua viabilidade. Ressalta-se que a atuação do DBA, especialmente na validação técnica das recomendações, permanece essencial para garantir a segurança e aplicabilidade das configurações em produção.

Para evoluções futuras, destacam-se quatro frentes principais. A primeira envolve o uso de modelos de (*ensembles*) para aprimorar o R^2 e a estabilidade da regressão. A segunda propõe a adoção de um ciclo de realimentação contínuo, no qual as recomendações são testadas em ambiente controlado e os resultados retornam ao modelo para refinamento incremental. A terceira frente diz respeito à formulação de objetivos múltiplos, buscando não apenas reduzir latência, mas também balancear outros aspectos como *throughput*, uso de CPU e memória. Por fim, a quarta direção considera a melhoria das estratégias de aquisição, com foco em áreas do espaço paramétrico na quais a incerteza do GPR é elevada, ampliando a efetividade da exploração.

Referências

[Aken et al. 2017] Aken, D. V. et al. (2017). Automatic dbms tuning through large-scale ml. In *Proc. ACM ICDE*, pages 1009–1024.

[Aken et al. 2021] Aken, D. V. et al. (2021). Ml-based automatic config tuning on real dbmss. *Proc. VLDB Endow.*, 14(7):1241–1253.

[Ansel et al. 2014] Ansel, J. et al. (2014). Opentuner: an extensible program auto-tuning framework. In *Proc. Int. Conf. Parallel Archit. Compilation*, pages 303–316. ACM.

[Cai et al. 2022] Cai, Y. et al. (2022). Hunter: an online cloud db hybrid tuner. *Proc. VLDB Endow.*, pages 646–659.

[Casella and Berger 2002] Casella, G. and Berger, R. L. (2002). *Statistical Inference*. Duxbury, 2nd edition.

[Cereda et al. 2021] Cereda, A. et al. (2021). Cgptuner: contextual gp bandit approach. *Proc. VLDB Endow.*, 14(8):1401–1413.

[Duarte and Ståhl 2019] Duarte, D. and Ståhl, N. (2019). Machine learning: a concise overview. In *Data Science in Practice*, volume 46, pages 27–58. Springer.

[Joshi et al. 2020] Joshi, K. K. et al. (2020). Ml–learning techniques, cnn, languages & apis. *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.*, 5(3):23–30.

[Kanellis et al. 2022] Kanellis, V. et al. (2022). Llamatune: sample-efficient dbms configuration tuning. arXiv:2203.05128.

[Lee 2019] Lee, W.-M. (2019). *Python Machine Learning*. Wiley.

[Li et al. 2020] Li, J.-P. et al. (2020). Ml & credit-ratings prediction. *Technol. Forecast. Soc. Change*, 161:120309.

[Rasmussen and Williams 2005] Rasmussen, C. E. and Williams, C. K. I. (2005). *Gaussian Processes for Machine Learning*. MIT.

[Shahriari et al. 2016] Shahriari, B. et al. (2016). A review of bayesian optimization. *Proc. IEEE*.

[Shaveta 2023] Shaveta (2023). A review on machine learning. *Int. J. Sci. Res. Arch.*, 9(1):281–285.

[Storm and Surendra 2006] Storm, A. and Surendra, S. (2006). Adaptive self-tuning memory in db2. *VLDB Endow.*

[Sullivan 2012] Sullivan, R. (2012). Machine-learning techniques. In *Intro. Data Mining Life Sci.*, pages 363–454. Humana.

[Tan et al. 2019] Tan, J. et al. (2019). ibtune: individualized buffer tuning. *Proc. VLDB Endow.*, 12(10):1221–1234.

[Tibshirani 1996] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. B*, 58(1):267–288.

[Trendafilov and Hirose 2023] Trendafilov, N. and Hirose, K. (2023). Exploratory factor analysis. In *Int. Encycl. Educ.*, pages 600–606. Elsevier, 4th edition.

[Trummer 2021] Trummer, I. (2021). Db-bert: a database tuning tool that 'reads the manual'.

[Vasyliev 2024] Vasyliev, A. (2024). Pgtune: configuring postgresql for max hardware performance. Online.

[Vyawahare 2022] Vyawahare, H. R. (2022). Machine learning: solution for complex problems. *Int. J. Sci. Res. Eng. Manag.*, 6(4):1–6.

[Yong and Pearce 2013] Yong, A. and Pearce, J. (2013). A beginner's guide to factor analysis: focusing on efa. *Tutorials Quant. Methods Psychol.*, 9(2):79–94.

[Zhang et al. 2021] Zhang, X. et al. (2021). Restune: meta-learning for cloud dbs. *Proc. VLDB Endow.*, pages 2102–2114.

[Zhang et al. 2022] Zhang, X. et al. (2022). Towards dynamic & safe configuration tuning for cloud dbs. *Proc. VLDB Endow.*, pages 631–645.

[Zhao et al. 2023] Zhao, Y. et al. (2023). Auto-config tuning for dbs: survey. *VLDB J.*, 32:835–862.