

Benchmark de diferentes modelos com Emergent Web Server e Digital Twins

F. B. Marques R. S. Dias L. F. Bittencourt R. R. Filho

Relatório Técnico - IC-PFG-25-34

Projeto Final de Graduação

2025 - Dezembro

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE COMPUTAÇÃO

The contents of this report are the sole responsibility of the authors.
O conteúdo deste relatório é de única responsabilidade dos autores.

Benchmark de diferentes modelos com Emergent Web Server e Digital Twins

Fernando Braunstein Marques* Ricardo Santos Dias†
Luiz Fernando Bittencourt‡ Roberto Rodrigues Filho§

Resumo

Este trabalho investiga o uso de diferentes modelos preditivos para tentar otimizar o desempenho do Emergent Web Server (EWS), um servidor web auto-adaptativo que utiliza aprendizado por reforço para ajustar dinamicamente suas configurações. O objetivo é realizar um benchmark entre abordagens de predição, avaliando o impacto de cada uma na estabilidade e eficiência do sistema. Para isso, o EWS foi integrado a um Gêmeo Digital, que simula o comportamento do sistema real por meio de uma arquitetura de coleta de dados e modelos baseados em XGBoost, Extra Trees, Histogram Gradient Boosting (HGB), LightGBM (LGBM), Random Forest e Multi-Layer Perceptron (MLP). Dessa forma, conseguimos fazer uma predição de auto-distribuição de recursos de forma mais barata, através do Gêmeo Digital, e posteriormente testar se a distribuição se mantém relevante no EWS de fato.

1 Introdução

O avanço da computação distribuída e o crescimento da complexidade dos sistemas modernos têm acelerado o desenvolvimento de soluções capazes de operar com maior autonomia e adaptabilidade. À medida que aplicações em cloud, serviços web e aplicativos tornam-se mais mutáveis, cresce também a necessidade de mecanismos que permitam o autoajuste em tempo real, reduzindo a dependência de administração humana direta [1]. Nesse contexto, surge o interesse por arquiteturas capazes de observar seu próprio comportamento, aprender com o ambiente e reagir a mudanças de forma coerente.

Considerando as diferentes arquiteturas relacionadas, destacam-se os sistemas auto-adaptativos e emergentes, projetados para ajustar sua configuração ou recursos alocados a partir de políticas e objetivos previamente definidos [2, 3]. Um exemplo notável é o *Emergent Web Server* (EWS), um servidor web experimental que aplica técnicas de aprendizado por reforço para modificar sua alocação de recursos e otimizar o desempenho sob

*Instituto de Computação, Universidade Estadual de Campinas, 13081-970 Campinas, SP.

†Instituto de Computação, Universidade Estadual de Campinas, 13081-970 Campinas, SP.

‡Instituto de Computação, Universidade Estadual de Campinas, 13081-970 Campinas, SP.

§Departamento de Ciência da Computação, Universidade de Brasília, 70910-900 Brasília, DF.

diferentes condições de carga [4]. A mutabilidade desse sistema o torna um ambiente ideal para experimentos relacionados a técnicas de predição e controle autônomo.

Paralelamente, os Gêmeos Digitais (*Digital Twins*) vêm se consolidando como uma ferramenta essencial para modelar e prever o comportamento de sistemas complexos em escalas menores, diminuindo os custos relacionados [5]. Através de uma réplica virtual sincronizada ao sistema real, é possível analisar o impacto da realocação de recursos e antecipar anomalias de forma segura e eficiente, antes de aplicar mudanças no ambiente real.

Diante desse cenário, este projeto propõe investigar o uso de diferentes modelos preditivos no contexto do EWS, avaliando como distintas técnicas de aprendizado podem representar e antecipar seu comportamento adaptativo. O objetivo é comparar modelos como Extra Trees, HGB, LGBM, MLP, Random Forest e XGBoost quanto à precisão, estabilidade e capacidade de generalização, buscando identificar quais abordagens oferecem maior benefício para sistemas auto-organizáveis. Essa análise fornece uma base para futuras arquiteturas que combinem o EWS com Gêmeos Digitais de forma mais robusta e preditiva.

2 Referencial Teórico

Sistemas modernos operam em ambientes altamente variáveis, nos quais fatores como carga de trabalho, disponibilidade de recursos e falhas imprevistas exigem mecanismos de adaptação contínua. A pesquisa em software autônomo e auto-adaptativo busca oferecer respostas a esses desafios, utilizando princípios de controle, aprendizado e modelagem digital. Nesta seção, são apresentados os principais conceitos que sustentam este trabalho: Computação Autônômica, Gêmeos Digitais e Sistemas Emergentes, com destaque para o caso do *Emergent Web Server* (EWS) como plataforma de experimentação.

2.1 Computação Autônômica

A computação autônômica foi proposta pela IBM no início dos anos 2000 como uma nova direção para a gestão de sistemas complexos [1]. Inspirada no funcionamento do sistema nervoso humano, esse conceito busca criar sistemas capazes de se autorregular, ajustando seu comportamento em resposta a variações externas ou internas.

O modelo conceitual mais conhecido é o ciclo *MAPE-K* (*Monitor, Analyze, Plan, Execute – Knowledge*), que define as etapas de monitoramento, análise, planejamento e execução de ações baseadas em conhecimento acumulado. Esse ciclo cria um laço de controle contínuo responsável por manter o sistema dentro de parâmetros desejáveis, mesmo sob condições mutáveis.

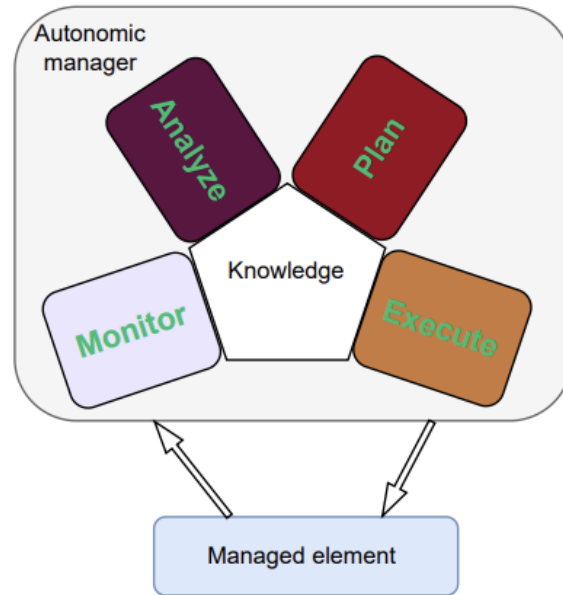


Figura 1: Modelo MAPE-K de controle autônomo. Adaptado de [1].

Sistemas auto-adaptativos implementam variações desse ciclo para ajustar recursos, reconfigurar módulos e reagir a falhas em tempo real [2, 3]. Esses princípios formam a base conceitual sobre a qual se apoiam tanto os Gêmeos Digitais quanto os sistemas emergentes analisados neste projeto.

2.2 Gêmeos Digitais (Digital Twins)

O conceito de Gêmeo Digital (*Digital Twin*) descreve a criação de uma réplica virtual de um sistema físico ou lógico, capaz de simular, monitorar e prever seu comportamento [5]. Diferente de simples modelos de simulação, o Gêmeo Digital mantém uma sincronização contínua com o sistema real, refletindo seu estado em tempo quase real e permitindo a realização de análises preditivas e prescritivas.

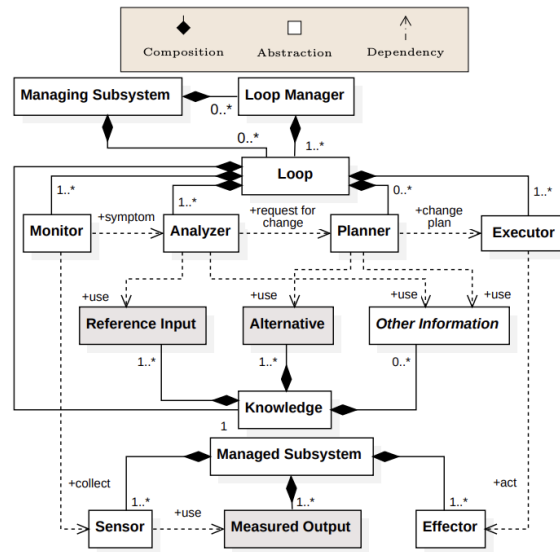


Figura 2: Arquitetura conceitual de Gêmeo Digital baseada no framework MODA. Adaptado de [5].

A utilização de Gêmeos Digitais têm se expandido para áreas como manufatura, redes de comunicação e sistemas autônomos, onde permitem prever falhas, testar cenários e otimizar decisões sem afetar o ambiente real. No contexto deste trabalho, o Gêmeo Digital é considerado como um mecanismo de apoio à camada de aprendizado e controle do EWS, fornecendo uma representação preditiva de seu comportamento, em que são criados gêmeos digitais de representações *menores* do EWS para tentar prever o balanceamento de recursos com um menor custo.

2.3 Sistemas Emergentes e o Emergent Web Server (EWS)

Os sistemas emergentes estendem o paradigma auto-adaptativo ao permitir que a estrutura interna de um software se reorganize dinamicamente, em vez de apenas ajustar parâmetros de execução [6, 7]. Essa característica é obtida pela combinação de componentes modulares que podem ser reconfigurados em tempo de execução, permitindo que o sistema evolua conforme as condições do ambiente.

O *Emergent Web Server* (EWS) exemplifica essa abordagem. Ele utiliza o framework PAL (*Perception-Assembly-Learning*), que integra monitoramento, recomposição e aprendizado de reforço em um ciclo de adaptação contínuo. O servidor é capaz de gerar e avaliar múltiplas configurações, aprendendo quais oferecem melhor desempenho sob diferentes condições.

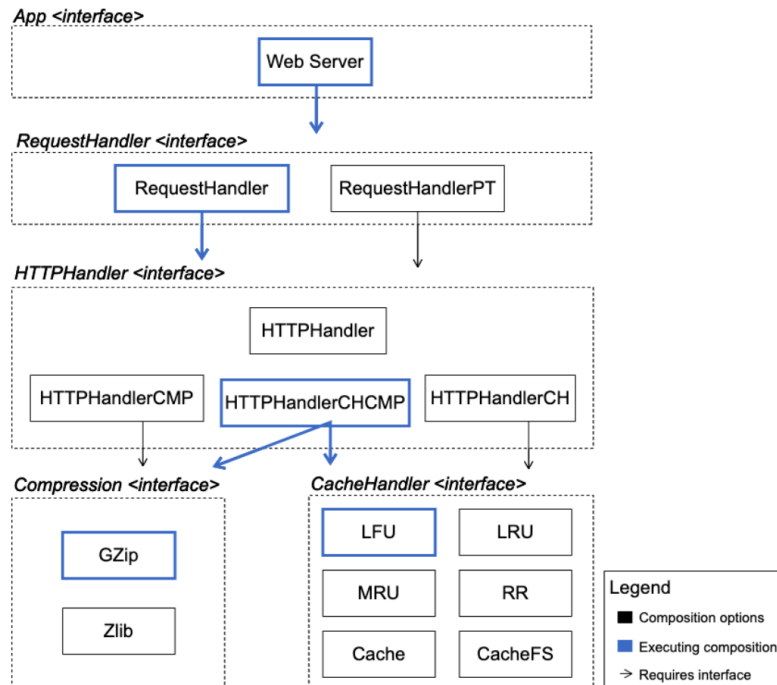


Figura 3: Arquitetura conceitual do framework PAL e do EWS. Adaptado de [7].

Essa capacidade de aprendizado estrutural torna o EWS um ambiente experimental ideal para avaliar a eficácia de modelos preditivos. Ao testar algoritmos de predição, é possível comparar a qualidade de decisões automatizadas e o impacto de cada modelo na estabilidade e eficiência do sistema.

2.4 Convergência entre Computação Autônoma, Gêmeos Digitais e Sistemas Emergentes

A união entre os conceitos de computação autônoma, gêmeos digitais e sistemas emergentes aponta para uma nova geração de sistemas auto-evolutivos. Enquanto o MAPE-K define os mecanismos de controle, os Gêmeos Digitais fornecem capacidade de previsão e simulação, e os sistemas emergentes trazem a adaptabilidade estrutural. A junção de todos esses elementos possibilita a criação de sistemas que não apenas se ajustam, mas também aprendem e se transformam com base em sua própria experiência [4, 5].

No contexto deste trabalho, essa convergência é explorada de forma prática, testando diferentes modelos preditivos como Extra Trees, HGB, LGBM, MLP, Random Forest e XGBoost na camada de decisão do EWS. O objetivo é identificar qual modelo apresenta melhor desempenho ao prever estados futuros e apoiar o processo de autoajuste.

3 Objetivo

O principal objetivo deste trabalho é realizar um *benchmark* entre diferentes modelos de predição aplicados ao *Emergent Web Server* (EWS), explorando como técnicas de aprendizado podem apoiar sua camada de autoajuste e contribuir para uma operação mais estável e eficiente com foco em comparar modelos preditivos distintos, e não em provar a integração entre EWS e Gêmeo Digital. A proposta envolve o uso de um Gêmeo Digital (*Digital Twin*) que simula o comportamento do servidor real, permitindo testar estratégias de adaptação em um ambiente controlado antes de aplicá-las no sistema físico.

De forma mais específica, pretende-se: comparar o desempenho de modelos de aprendizado supervisionado na tarefa de prever métricas de desempenho do EWS e avaliar a estabilidade dos resultados obtidos pelo Gêmeo Digital frente a diferentes padrões de carga e composição de componentes.

O estudo busca responder a duas principais teses: em que medida modelos de predição podem representar com fidelidade o comportamento adaptativo do EWS e até que ponto a utilização do Gêmeo Digital contribui para atenuar oscilações de desempenho observadas durante os ciclos de autoajuste do sistema.

4 Metodologia

A metodologia deste trabalho fundamenta-se na integração entre os paradigmas de Sistemas de Software Emergentes e Gêmeos Digitais, utilizando uma abordagem experimental para validar a eficácia de modelos preditivos na estabilização de servidores web auto-adaptativos. O desenvolvimento seguiu o ciclo de vida de sistemas autônômicos, abrangendo desde a coleta de dados não intrusiva até a atuação dinâmica no sistema real.

4.1 Arquitetura do Sistema e Integração

A arquitetura proposta baseia-se na convergência de dois frameworks conceituais: o **PAL** (*Perception-Assembly-Learning*) [4], que provê as capacidades de auto-adaptação do Emergent Web Server (EWS), e o **MODA** (*Models and Data*) [5], que estrutura a implementação do Gêmeo Digital.

Para viabilizar a adaptação sem a penalidade de desempenho típica da exploração *online*, desenvolveu-se uma camada de *Digital Shadow*. Esta camada atua como um espelho de dados, coletando métricas em tempo real através de agentes de monitoramento que leem os logs de acesso e registram o tempo de resposta do servidor. Diferentemente da abordagem nativa do EWS, que realiza testes intrusivos no ambiente de produção, nossa metodologia utiliza esses dados históricos para treinar modelos *offline*, permitindo que o "custo" do aprendizado seja pago no ambiente virtual.

4.2 Definição de Cenários e Carga de Trabalho

Para avaliar a robustez dos modelos sob situações variadas, foram definidos dois cenários de tráfego distintos, simulados por clientes desenvolvidos especificamente para este fim:

- **Cenário de Texto (Leve):** Utiliza requisições a recursos estáticos de baixo tamanho (arquivos HTML e scripts simples). Este cenário simula uma carga de trabalho onde o gargalo tende a ser a latência de rede e o *overhead* do processamento de requisições (I/O bound de baixa intensidade). O objetivo é verificar se o modelo consegue otimizar o *throughput* em situações de alta concorrência e baixo tempo de serviço.
- **Cenário de Imagem (Pesado):** Envolve o processamento e transferência de arquivos de imagem de alta resolução. Este cenário impõe uma carga significativa de CPU e largura de banda ao servidor. A utilização deste tipo de tráfego é crucial para validar a capacidade do Gêmeo Digital em prever degradações severas de desempenho, que são mais frequentes e custosas em operações pesadas.

4.3 Modelagem Preditiva e Algoritmos

O núcleo de decisão do Gêmeo Digital substitui a exploração aleatória por inferências baseadas em aprendizado de máquina supervisionado. Foram selecionados e avaliados seis algoritmos distintos, escolhidos por sua capacidade de lidar com dados tabulares e relações não-lineares entre as configurações do servidor e o tempo de resposta:

1. **XGBoost (eXtreme Gradient Boosting):** Um algoritmo baseado em *gradient boosting* conhecido por sua eficiência e escalabilidade [8]. Foi selecionado como o modelo principal devido à sua capacidade comprovada em competições de ciência de dados e sua robustez contra *overfitting* através de regularização avançada.
2. **Random Forest (RF):** Um método de *ensemble* do tipo *bagging* que constrói múltiplas árvores de decisão durante o treinamento. Sua inclusão deve-se à sua estabilidade e baixa tendência a *overfitting*, servindo como um comparativo robusto para os métodos de *boosting*.
3. **LightGBM (LGBM):** Similar ao XGBoost, mas otimizado para velocidade e eficiência de memória, utilizando uma abordagem de crescimento de árvore baseada em folhas (*leaf-wise*). Foi testado visando cenários onde o retreinamento rápido do Gêmeo Digital seja crítico.
4. **HistGradientBoosting (HGB):** Uma implementação de *gradient boosting* otimizada para grandes conjuntos de dados, que utiliza binagem de histogramas para acelerar o treinamento.
5. **Extra Trees (Extremely Randomized Trees):** Uma variação do Random Forest que introduz mais aleatoriedade na escolha dos cortes das árvores, visando reduzir ainda mais a variância do modelo.
6. **Multi-layer Perceptron (MLP):** Uma rede neural artificial *feedforward*. Foi incluída para contrastar a performance de métodos baseados em árvores com abordagens conexionistas, avaliando se a complexidade de uma rede neural traz benefícios para este tipo específico de problema de regressão.

Todos os modelos foram submetidos a uma etapa de otimização de hiperparâmetros utilizando o framework *Optuna*, com 100 iterações de busca para minimizar o Erro Absoluto Médio (MAE).

4.4 Processamento de Dados e Análise Estatística

A análise dos resultados brutos de tempo de resposta em sistemas distribuídos frequentemente sofre com a presença de ruídos exógenos (e.g., latência transiente de rede, pausas de *Garbage Collection*). Para garantir que a avaliação refletisse o comportamento do algoritmo de adaptação e não anomalias de infraestrutura, aplicou-se um método rigoroso de limpeza de dados.

4.4.1 Remoção de Outliers

Foi implementada uma filtragem estatística para remover *outliers* extremos nos gráficos de resultados. A justificativa para esta remoção são as seguintes:

1. **Visualização da Convergência:** Picos isolados de latência (que podem ser ordens de grandeza superiores à média) distorcem a escala dos gráficos, mascarando a tendência de estabilização do sistema. A remoção permite observar com clareza o momento em que o EWS converge para a configuração ótima.
2. **Foco no Algoritmo:** O objetivo do trabalho é avaliar a política de decisão do Gêmeo Digital. Variações extremas e esporádicas geralmente não são causadas pela configuração escolhida, mas sim por fatores externos, devendo ser descartadas para uma comparação justa entre os modelos.

Os dados apresentados nos gráficos foram agrupados em janelas temporais (agrupamentos de requisições a cada 5 segundos ou 100ms, conforme a densidade), plotando-se a média do tempo de resposta dentro dessa janela. Isso suaviza a variância natural do tráfego de rede e destaca o comportamento macroscópico de cada modelo durante as fases de exploração e exploração.

5 Resultados Experimentais

Esta seção apresenta a análise comparativa de desempenho entre o Emergent Web Server (EWS) operando em modo nativo (abordagem *baseline*) e a arquitetura proposta assistida por Gêmeos Digitais. A avaliação foca na capacidade dos modelos preditivos em mitigar as oscilações de tempo de resposta (*Response Time*) durante as fases de adaptação.

Para garantir uma análise mais robusta das tendências centrais de desempenho, foram removidos outliers extremos dos dados brutos, permitindo uma visualização mais clara do comportamento de convergência dos modelos. Os experimentos foram divididos em dois cenários de carga de trabalho: requisições de texto (baixa carga computacional) e requisições de imagem (alta carga computacional).

5.1 Cenário 1: Tráfego de Texto

Para as requisições de texto, o desempenho do sistema foi avaliado comparando o baseline contra seis modelos preditivos distintos: XGBoost, Random Forest (RF), Multi-layer Perceptron (MLP), LightGBM (LGBM), HistGradientBoosting (HGB) e Extra Trees (ET).

5.1.1 Análise do Baseline

A Figura 4 ilustra o comportamento do EWS operando localmente sem o auxílio do Gêmeo Digital. Mesmo após o tratamento de dados, nota-se uma variabilidade expressiva no tempo de resposta, característica do algoritmo de *Reinforcement Learning* nativo, que penaliza o desempenho do sistema com picos de latência enquanto explora configurações sub-ótimas.

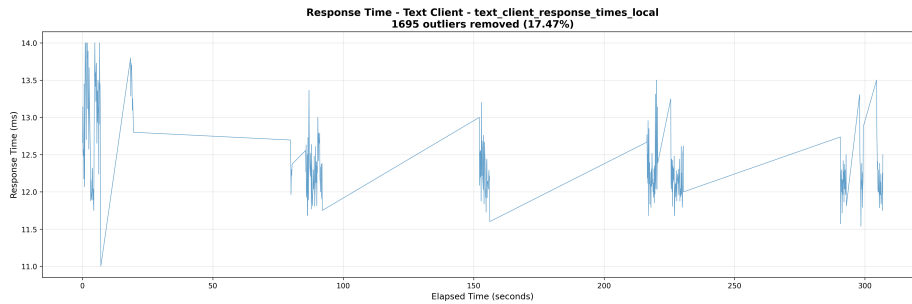


Figura 4: Baseline: Oscilação de desempenho no tráfego de texto sem Gêmeo Digital.

Observa-se na Figura 4 a presença de rampas de subida ou descida gradual no tempo de resposta. Essas rampas não indicam estabilidade, mas sim o fato de que o algoritmo de *Reinforcement Learning* mantém a mesma configuração por algum tempo durante a exploração. Como os dados são agregados por janelas, o tempo médio de resposta aumenta ou diminui de maneira contínua até que o algoritmo troca de configuração, produzindo a queda brusca observada no gráfico.

Mesmo que os valores absolutos permaneçam próximos (por exemplo, entre 10 e 14 ms), a instabilidade se revela na variação entre janelas consecutivas. No baseline, essas variações são abruptas, pois o RL alterna constantemente entre configurações boas e ruins. Assim, a sua variabilidade refere-se ao comportamento temporal e não ao valor médio em si.

5.1.2 Desempenho com Gêmeos Digitais

A introdução do Gêmeo Digital alterou drasticamente o perfil de resposta, promovendo uma estabilidade superior. As Figuras 9 e 12 apresentam os resultados comparativos dos seis modelos.

A estabilidade observada com o uso do Gêmeo Digital ocorre porque o sistema real recebe configurações ruins com menor frequência. Como essas configurações são filtradas no ambiente virtual, o EWS deixa de entrar em regimes de degradação progressiva, observando assim comportamentos mais planos na maior parte dos testes com modelos com DTs.



Figura 5: DT com XGBoost (Texto).



Figura 6: DT com Random Forest (Texto).

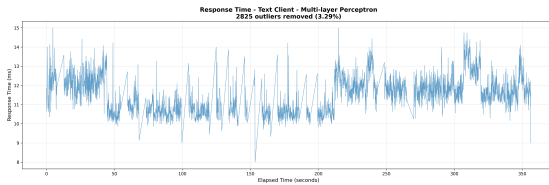


Figura 7: DT com MLP (Texto).

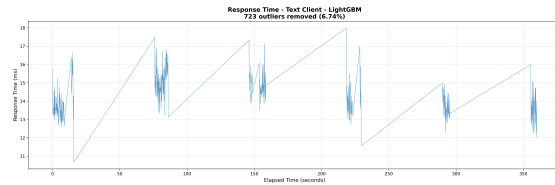


Figura 8: DT com LightGBM (Texto).

Figura 9: Comparativo: XGBoost, RF, MLP e LightGBM para texto.

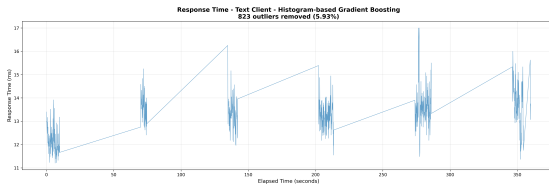


Figura 10: DT com HistGradientBoosting.

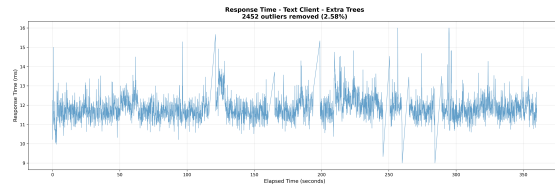


Figura 11: DT com Extra Trees.

Figura 12: Modelos adicionais de ensemble para texto.

Os modelos baseados em *gradient boosting* apresentaram diferenças importantes entre si. O XGBoost destacou-se como o modelo mais estável, convergindo rapidamente para tempos de resposta baixos e mantendo uma variância mínima ao longo do experimento. Já o LightGBM, embora eficiente na redução da latência média, apresentou rampas residuais e oscilações periódicas, indicando que sua estabilidade é inferior à do XGBoost. O modelo Random Forest manteve um comportamento consistentemente estável, com baixa variação inter-janelas, enquanto o MLP exibiu maior ruído no início da execução, comportamento esperado de redes neurais ajustando-se a dados em fluxo contínuo.

Dessa forma, a distinção essencial entre o baseline e os modelos com DT está no comportamento temporal: enquanto o baseline apresenta rampas e quedas bruscas devido à exploração, a maioria dos modelos preditivos mantêm o servidor em estados estáveis por longos períodos. Mesmo com valores absolutos semelhantes, o DT reduz drasticamente a variabilidade entre janelas.

5.2 Cenário 2: Tráfego de Imagem

O processamento de imagens impõe maior estresse ao servidor, tornando o custo da exploração mais severo no sistema real. A Figura 13 demonstra como o baseline sofre com

latências elevadas e instáveis durante a busca por configurações.

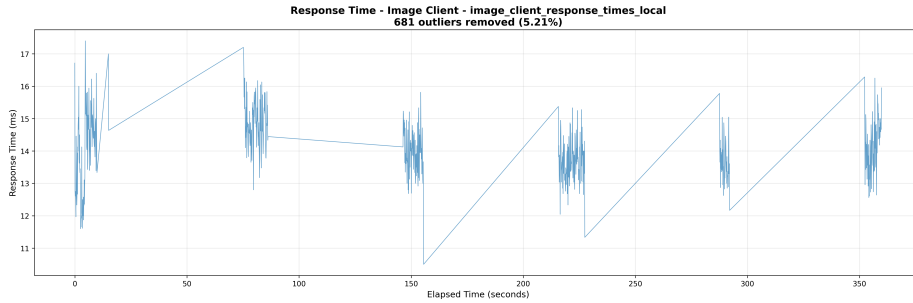


Figura 13: Baseline: Alta latência e instabilidade no tráfego de imagem.

A aplicação do Gêmeo Digital foi avaliada com os mesmos seis modelos, conforme ilustrado nas Figuras 18 e 21.



Figura 14: DT com XGBoost (Imagem).

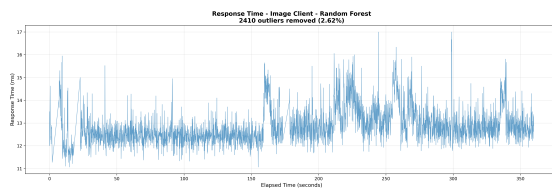


Figura 15: DT com Random Forest (Imagem).

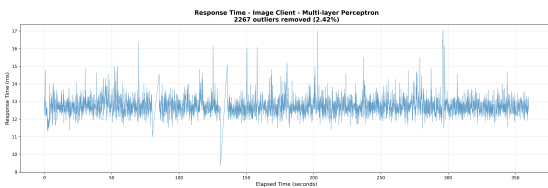


Figura 16: DT com MLP (Imagem).



Figura 17: DT com LightGBM (Imagem).

Figura 18: Comparativo principal de modelos para tráfego de imagem.

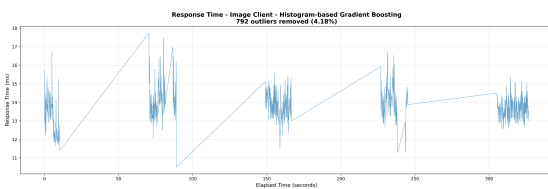


Figura 19: DT com HistGradientBoosting.



Figura 20: DT com Extra Trees.

Figura 21: Modelos adicionais de ensemble para imagem.

É possível observar claramente como os Gêmeos Digitais afetaram o tempo de latência. Mesmo com comportamentos diferentes, vemos padrões de latência muito mais constantes

do que o baseline. O modelo **XGBoost** (Fig. 14) novamente se destacou pela estabilidade, mantendo a linha de tempo de resposta próxima a plana e com uma latência baixa.

O modelo **MLP** (Fig. 16) e o **Extra Trees** (Fig. 20) apresentaram pequenas oscilações residuais, mas ainda assim garantiram um desempenho superior à exploração nativa. Isso confirma a hipótese de que a transferência da fase de aprendizado para o ambiente virtual protege o sistema real das penalidades de desempenho associadas a configurações ineficientes.

6 Conclusão

O objetivo central deste trabalho foi validar se a incorporação de modelos preditivos, através de Gêmeos Digitais, é capaz de auxiliar e otimizar o desempenho do *Emergent Web Server* (EWS). A investigação focou em determinar se a substituição da exploração nativa (intrusiva) por uma exploração virtual (preditiva) traria ganhos tangíveis de estabilidade e tempo de resposta.

A análise dos resultados experimentais permite responder afirmativamente a essa questão, levando às seguintes conclusões sobre o impacto dos modelos no comportamento do servidor:

6.1 Impacto na Estabilidade e Latência

A utilização dos modelos preditivos transformou fundamentalmente o perfil de operação do EWS. Enquanto o sistema operando sozinho (*baseline*) sofreu com degradações severas de desempenho, causadas pela necessidade de testar configurações ineficientes em ambiente de produção, o sistema assistido pelos modelos eliminou essas oscilações.

O ganho de desempenho foi confirmado pela capacidade dos modelos em filtrar configurações ruins antes que elas fossem aplicadas ao sistema real. No cenário crítico de tráfego de imagens, embora os valores absolutos de latência permaneçam próximos ao baseline, a maior parte dos modelos com DT eliminou as rampas de degradação características da exploração nativa e reduziram significativamente a variação entre janelas. Como consequência, o EWS deixou de passar longos períodos em configurações ineficientes, resultando em curvas mais regulares e previsíveis ao longo do tempo, evidenciando a importância do apoio preditivo em cenários sensíveis à latência.

6.2 Benchmark de Eficácia dos Modelos

Apesar do ganho de desempenho evidente na utilização dos modelos, ao comparar os diferentes algoritmos para esta finalidade específica, conclui-se que nem todos os modelos oferecem o mesmo grau de auxílio:

- **Modelos de Boosting (XGBoost, LightGBM):** O XGBoost mostrou-se o modelo mais eficaz, apresentando a convergência mais rápida e o comportamento mais estável entre todos os avaliados. O LightGBM também apresentou bom desempenho em termos de latência média, mas manteve rampas e oscilações que o tornam menos estável que o XGBoost.

- **Modelos de Bagging (Random Forest, Extra Trees):** Ofereceram um auxílio robusto e muito superior ao baseline, embora com uma curva de aprendizado ligeiramente menos agressiva que os modelos de boosting em certos cenários.
- **Redes Neurais (MLP):** Embora funcionais, provaram ser a opção menos eficiente para este contexto. A maior variância nas previsões do MLP resultou em uma estabilização mais lenta do EWS comparada aos métodos baseados em árvores, indicando que a complexidade deste modelo não se traduz em ganho de desempenho proporcional para dados tabulares de sistemas web.

6.3 Considerações Finais

Conclui-se que a estratégia de utilizar modelos preditivos não apenas auxilia, mas é um componente crítico para a performance de sistemas emergentes. O "custo" computacional de manter e consultar esses modelos é irrelevante se comparado ao ganho obtido pela eliminação das penalidades de exploração no ambiente físico. Portanto, a abordagem valida-se como um mecanismo eficaz para garantir que o EWS possa se auto-adaptar continuamente sem comprometer a qualidade de serviço (QoS) entregue aos usuários.

Referências

- [1] KEPHART, J. O.; CHESS, D. M. *The vision of autonomic computing*. Computer, v. 36, n. 1, p. 41–50, 2003. DOI: 10.1109/MC.2003.1160055.
- [2] SALEHIE, M.; TAHVILDARI, L. *Self-adaptive software: Landscape and research challenges*. ACM Transactions on Autonomous and Adaptive Systems, v. 4, n. 2, p. 1–42, 2009. DOI: 10.1145/1516533.1516538.
- [3] ELKHODARY, A.; ESFAHANI, N.; MALEK, S. *FUSION: A framework for engineering self-tuning self-adaptive software systems*. In: Proceedings of the 18th ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE '10). Santa Fe, NM, USA: ACM, 2010. p. 7–16. DOI: 10.1145/1882291.1882296.
- [4] RODRIGUES FILHO, R. et al. *Emergent Software Systems: Theory and Practice*. In: Minicursos do XXXIX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC'21). Porto Alegre: SBC, 2021. p. 1–50.
- [5] ERAMO, R. et al. *Conceptualizing Digital Twins*. IEEE Software, v. 39, n. 2, p. 39–46, 2022. DOI: 10.1109/MS.2021.3130755.
- [6] PORTER, B.; FILHO, R. R. *Losing Control: The Case for Emergent Software Systems Using Autonomous Assembly, Perception, and Learning*. In: IEEE 10th International Conference on Self-Adaptive and Self-Organizing Systems (SASO), Augsburg, Germany, 2016. IEEE, 2016. p. 40–49. DOI: 10.1109/SASO.2016.10.
- [7] RODRIGUES FILHO, R. *Emergent Software Systems*. Tese (Doutorado em Ciência da Computação) – Lancaster University, Lancaster, Reino Unido, 2018.

- [8] CHEN, T.; GUESTRIN, C. *XGBoost: A scalable tree boosting system*. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16). San Francisco, EUA: ACM, 2016. p. 785–794. DOI: 10.1145/2939672.2939785.