

OPTIFETCH: Um Agente de Prefetching Probabilístico para Apoiar a Experiência de Aprendizagem em Sistemas Tutores Inteligentes sob Conectividade Instável

Alex Almeida do Amaral¹, João Carlos Herculano da Silva Queiroz¹,
Huliane Medeiros da Silva², Geiser Chalco Chalco²

¹Programa de Pós-Graduação em Ciência da Computação (PPGCC)
Universidade Federal Rural do Semi-Árido (UFERSA)
Universidade do Estado do Rio Grande do Norte (UERN) – Mossoró – RN – Brasil

²Departamento de Engenharias e Tecnologia (DETEC)
Universidade Federal Rural do Semi-Árido (UFERSA) – Pau dos Ferros – RN – Brasil

{alex.amaral, joao.queiroz72216}@alunos.ufersa.edu.br

{huliane, geiser}@ufersa.edu.br

Abstract. *Intelligent Tutoring Systems based on remote language models may experience high latency under unstable connectivity. This paper presents OPTIFETCH, a prefetching agent that uses the error probability estimated by Bayesian Knowledge Tracing to anticipate hints. The proposal was evaluated through controlled simulations in Wi-Fi, 4G, and rural connectivity scenarios, compared with On-Demand and Aggressive Prefetching strategies. Under the simulated conditions, OPTIFETCH reduced perceived latency by 35.3% to 44.4% compared with On-Demand and achieved a lower aggregate cost than both reference strategies. The results indicate operational feasibility but do not provide evidence of an impact on learning.*

Resumo. *Sistemas Tutores Inteligentes baseados em modelos de linguagem remotos podem apresentar alta latência sob conectividade instável. Este trabalho apresenta o OPTIFETCH, um agente de prefetching que utiliza a probabilidade de erro estimada pelo Bayesian Knowledge Tracing para antecipar dicas. A proposta foi avaliada por simulações controladas em cenários Wi-Fi, 4G e rural, em comparação com as estratégias On-Demand e Prefetching Agressivo. Nas condições simuladas, o OPTIFETCH reduziu a latência percebida entre 35,3% e 44,4% frente ao On-Demand e apresentou menor custo agregado que as duas referências. Os resultados indicam viabilidade operacional, mas não evidenciam impacto sobre a aprendizagem.*

1. Introdução

A escola pública brasileira convive, há anos, com um descompasso silencioso: enquanto a pesquisa em *Artificial Intelligence in Education* (AIED) avança para experiências cada vez mais ricas e adaptativas, parte significativa dos estudantes acessa essas experiências por meio de redes 4G instáveis, conexões intermitentes ou infraestruturas escolares precárias. Esse contraste é particularmente sensível em regiões rurais e periféricas

do Sul Global, nas quais a conectividade não é apenas uma variável técnica, mas um fator determinante de quem consegue, de fato, participar de uma situação de aprendizagem mediada por tecnologia [Isotani et al. 2023, Rodrigues et al. 2023].

Os Sistemas Tutores Inteligentes (STI) ocupam, neste cenário, um papel duplamente delicado. De um lado, são sistemas voltados ao acompanhamento individual do aluno e à entrega de *feedback* adaptativo; de outro, têm absorvido cada vez mais componentes baseados em modelos de Linguagem de Grande Escala (LLMs), frequentemente integrados por meio de arquiteturas como *Retrieval-Augmented Generation* (RAG) [Swacha and Gracel 2025, Li et al. 2025, Dakshit 2024]. Esses componentes ampliam a riqueza pedagógica das dicas oferecidas, mas, ao mesmo tempo, introduzem uma dependência crítica de chamadas remotas a serviços de nuvem. Em ambientes de conectividade restrita, essa dependência tende a se manifestar exatamente onde o aluno mais precisa de continuidade: durante a tentativa de resolver um problema.

Quando uma dica demora a chegar, o problema não se restringe ao funcionamento técnico do sistema. A responsividade pode afetar a percepção de controle, o engajamento e a continuidade da interação, enquanto atrasos prolongados podem interromper o raciocínio durante a resolução de uma atividade [Tolia et al. 2006, Seow 2008, Zhao et al. 2017, Harmon et al. 2024]. Como a utilidade de uma dica também depende do momento em que ela é disponibilizada, a latência pode reduzir sua oportunidade temporal em relação à ação que o estudante está realizando. Essa relação não implica, contudo, que a redução da latência produza diretamente melhores resultados de aprendizagem.

O custo dessa dependência também ultrapassa a latência percebida: chamadas remotas à API envolvem consumo de bateria, transferência de dados e custos financeiros que podem dificultar a adoção e a manutenção de STIs baseados em LLMs, especialmente em iniciativas públicas e contextos com recursos limitados [Ko et al. 2017, Satyanarayanan 2017]. Estratégias de *prefetching*, investigadas em computação móvel, antecipam requisições com base em modelos preditivos e funções de custo que consideram latência, energia, dados e desperdício [Zhao et al. 2018a, Zhao et al. 2018b, Siris et al. 2014]. Diante disso, este trabalho investiga como adaptar essa lógica ao contexto educacional, decidindo quando antecipar uma dica com base tanto nas condições de infraestrutura quanto na probabilidade estimada de domínio da habilidade em exercício.

Este artigo apresenta o OPTIFETCH, um agente de *prefetching* probabilístico voltado especificamente a STIs que dependem de LLMs em ambientes de conectividade instável. O agente utiliza estimativas produzidas por um modelo de *Bayesian Knowledge Tracing* (BKT) para inferir a probabilidade de o aluno errar na próxima interação relativa a uma habilidade e, com base nessa probabilidade, decide dinamicamente se vale a pena pré-carregar uma dica. A decisão é guiada por uma Função de Custo Total que pondera latência percebida, bateria, dados, custo de API e desperdício de *prefetching*. Importa marcar, desde já, o escopo desta contribuição: não se trata de um estudo sobre melhoria de aprendizagem, e sim de uma proposta de infraestrutura pedagógica voltada a preservar a responsividade necessária para que experiências adaptativas continuem viáveis em contextos historicamente desfavorecidos. A avaliação aqui apresentada é exploratória, baseada em simulações controladas, e não substitui o estudo empírico com usuários reais.

Posicionamos o OPTIFETCH no diálogo com a agenda de *AIED Unplugged*

[Isotani et al. 2023], que defende sistemas educacionais baseados em IA capazes de operar com conectividade limitada ou inexistente. O trabalho oferece três contribuições: (i) uma formulação do problema de *prefetching* de dicas que articula uma estimativa probabilística de domínio e variáveis operacionais; (ii) uma Função de Custo Total parametrizável, cuja calibração para diferentes contextos e infraestruturas deverá ser investigada empiricamente; e (iii) uma análise exploratória, por simulação, dos *trade-offs* entre responsividade e consumo de recursos sob distintos cenários de conectividade.

Questões de Pesquisa. As contribuições da avaliação exploratória do OPTIFETCH é organizada em torno das seguintes três questões de pesquisa: **QP1:** Em que medida o OPTIFETCH reduz a latência percebida em comparação com estratégias tradicionais de obtenção de dicas?; **QP2:** Como o OPTIFETCH afeta o custo operacional total em diferentes cenários de conectividade?; e **QP3:** Como a utilização da probabilidade de erro estimada pelo BKT influencia os *trade-offs* entre responsividade e consumo de recursos?

A questão QP1 confronta a latência percebida do OPTIFETCH com as estratégias *On-Demand* e de *prefetching* agressivo nos três cenários de rede. A QP2 observa o custo operacional dessas mesmas estratégias, verificando se o agente sustenta um consumo controlado de recursos. A QP3 analisa de que forma o uso da estimativa de erro posiciona o OPTIFETCH no espaço de compromisso entre responsividade e custo.

2. Referencial Teórico

Latência, Feedback e Continuidade da Aprendizagem. A latência é o intervalo entre a ação do usuário e a resposta percebida; em STIs, ela determina quando o *feedback* chega ao aluno, e o valor pedagógico de uma intervenção depende de sua oportunidade temporal. Estudos mostram que tanto a latência média quanto o *jitter* afetam a qualidade percebida e o desempenho [Van Damme et al. 2024], o que se conecta ao conceito de *feedback immediacy*, segundo o qual o intervalo entre ação e resposta modula o efeito instrucional. Atrasos inferiores a $\sim 150\text{ms}$ sustentam uma experiência fluida, enquanto acima de $\sim 1\text{s}$ cresce a insatisfação [Tolia et al. 2006]; a faixa de 7 a 10s caracteriza um regime *captive*, com risco de ruptura do fluxo [Seow 2008].

A espera altera ainda a percepção do tempo [Zhao et al. 2017] e pode induzir desatenção que degrada o desempenho [Harmon et al. 2024], interpretada na aprendizagem como *cognitive interruption*. Há sustentação robusta para a relação entre latência elevada e degradação de experiência, engajamento e percepção de controle, mas não para causalidade direta sobre resultados de aprendizagem. Tomamos, portanto, a latência percebida como *proxy* de qualidade de experiência e continuidade pedagógica, sem reivindicar que sua redução, por si só, melhore o aprendizado.

AIED Unplugged e Computação em Conectividade Restrita. A *Inteligência Artificial Desconectada* (AIED Unplugged) refere-se a sistemas de IA projetados para operar com baixa ou nenhuma dependência de conectividade contínua [Isotani et al. 2023]. Tratar essas restrições como exceção, e não como requisito de projeto, aprofunda a desigualdade digital, sobretudo no Sul Global [Rodrigues et al. 2023, Isotani et al. 2023], e princípios como *edge computing* e processamento local [Satyanarayanan 2017] são posicionamentos sobre quem terá acesso à AIED

em uso real. O OPTIFETCH se filia a essa agenda de forma cautelosa: em vez de substituir as chamadas remotas por modelos locais, modula sua frequência e momento, antecipando-as quando há sinal de utilidade da dica, como camada intermediária entre a dependência total da nuvem e a operação desconectada.

Bayesian Knowledge Tracing (BKT). O *Bayesian Knowledge Tracing* (BKT) [Corbett and Anderson 1994] é um modelo probabilístico amplamente utilizado em STIs para estimar, ao longo do tempo, a probabilidade de domínio de uma habilidade pelo estudante. Uma habilidade, também denominada *skill* ou *Knowledge Component*, corresponde a uma unidade de competência considerada pelo sistema. A cada interação com uma questão, o modelo recebe uma evidência observável, representada por acerto ou erro, e atualiza a estimativa de domínio da habilidade correspondente.

Formalmente, o BKT representa o domínio de uma habilidade por meio de uma variável latente binária L_t e utiliza quatro parâmetros: $P(L_0)$, probabilidade inicial de domínio; $P(T)$, probabilidade de transição do estado de não domínio para o estado de domínio após uma oportunidade de aprendizagem; $P(G)$, probabilidade de acerto sem domínio da habilidade (*guess*); e $P(S)$, probabilidade de erro apesar do domínio da habilidade (*slip*) [Baker et al. 2008]. A probabilidade de erro é calculada por:

$$P(\text{erro}_t) = P(L_t)P(S) + (1 - P(L_t))(1 - P(G)). \quad (1)$$

Essa probabilidade é utilizada pelo OPTIFETCH como sinal de decisão. Quando a probabilidade estimada de erro é elevada, aumenta a possibilidade de que uma dica venha a ser necessária na interação seguinte; quando é baixa, a antecipação da dica apresenta maior possibilidade de resultar em *prefetching* não utilizado. Essa relação constitui uma hipótese operacional do agente, e não uma afirmação sobre o conhecimento global, as intenções ou as necessidades pedagógicas do estudante.

A escolha do BKT, em vez de alternativas como *Deep Knowledge Tracing* (DKT) [Piech et al. 2015], *Performance Factors Analysis* (PFA) [Pavlik et al. 2009] ou heurísticas baseadas em erros recentes, decorre dos requisitos operacionais do cenário investigado. Embora abordagens baseadas em redes neurais possam apresentar maior capacidade preditiva em determinados conjuntos de dados, elas geralmente exigem maior volume de dados e recursos computacionais e apresentam menor interpretabilidade. O PFA também oferece uma modelagem interpretável do desempenho, mas depende de um histórico adequadamente estruturado de oportunidades, acertos e erros por habilidade. Heurísticas baseadas apenas em erros recentes, por sua vez, não mantêm explicitamente uma estimativa probabilística de domínio.

O BKT foi adotado por combinar interpretabilidade probabilística, baixo custo computacional e possibilidade de execução local em dispositivos com recursos limitados. Essas características permitem utilizar sua estimativa diretamente na política de *prefetching*, sem exigir treinamento contínuo ou comunicação permanente com serviços remotos. Apesar de suas simplificações e de não representar fatores afetivos, motivacionais ou contextuais, o BKT permanece como uma abordagem consolidada de modelagem probabilística do domínio de habilidades em sistemas educacionais [Pelánek 2017].

Sistemas Tutores Inteligentes (STI). STIs são sistemas educacionais que usam

IA para fornecer instrução personalizada e que incorporam explicitamente um modelo do aluno para inferir seu estado de conhecimento [VanLehn 2006]. Sua arquitetura costuma ser descrita por um *laço interno*, que acompanha as ações do aluno e entrega *feedback* imediato, e um *laço externo*, que seleciona tarefas com base no conhecimento estimado [VanLehn 2006]. O OPTIFETCH atua no laço interno: não altera a sequência de problemas, mas a forma como as dicas geradas por LLMs são antecipadas e entregues, considerando a infraestrutura e a conectividade como requisitos centrais de projeto [Rodrigues et al. 2023].

3. Trabalhos Relacionados

A pesquisa sobre o OPTIFETCH se situa na interseção entre Sistemas Tutores Inteligentes, otimização de responsividade em sistemas interativos e operação de IA em conectividade restrita, articulando frentes que a literatura trata de forma majoritariamente independente. A agenda de *AIED Unplugged* [Isotani et al. 2023, Rodrigues et al. 2023] defende sistemas educacionais que operem com conectividade limitada, mas não propõe uma política de antecipação de dicas guiada por estimativas probabilísticas de domínio de habilidades. A modelagem do aluno em STIs [VanLehn 2006], em especial o BKT [Corbett and Anderson 1994, Baker et al. 2008, Pelánek 2017], consolida a estimativa probabilística de domínio de habilidades, mas para fins de sequenciamento, não de pré-busca. Sob a ótica de infraestrutura, a *edge computing* [Satyanarayanan 2017] e o *prefetching* em computação móvel [Zhao et al. 2018a, Zhao et al. 2018b, Siris et al. 2014] reduzem latência e custo por meio de modelos preditivos sensíveis ao custo, porém não utilizam estimativas de domínio de habilidades como sinal de decisão. Por fim, a integração de LLMs em STIs via RAG [Swacha and Gracel 2025, Li et al. 2025, Dakshit 2024, Granata et al. 2025] amplia a riqueza das dicas, mas raramente trata sua operação sob conectividade instável. A Tabela 1 sintetiza esse panorama.

Tabela 1. Comparação entre trabalhos relacionados e o OPTIFETCH.

Linha de trabalho			Usa LLM	Model. aluno	Prefetch.	Conect. limit.?
AIED	Unplugged	[Isotani et al. 2023, Rodrigues et al. 2023]	Não	—	Não	Sim
Knowledge tracing		[Corbett and Anderson 1994, Piech et al. 2015, Pavlik et al. 2009]	Não	Sim	Não	Não
Prefetch. móvel		[Zhao et al. 2018a, Siris et al. 2014]	Não	—	Sim (genérica)	Sim
RAG em STI		[Swacha and Gracel 2025, Li et al. 2025]	Sim	—	Não	Não
OPTIFETCH			Sim	BKT	Probab.	Sim

O cruzamento das colunas evidencia a lacuna investigada: os modelos de estudante estimam o domínio de habilidades, mas essas estimativas não são normalmente utilizadas para decidir sobre a pré-busca de dicas; por sua vez, as estratégias de *prefetching* móvel não empregam estimativas de domínio como sinal de decisão. Além disso, os trabalhos sobre infraestrutura e RAG não oferecem um mecanismo de antecipação de dicas guiado pela probabilidade de erro estimada pelo BKT. O OPTIFETCH ocupa essa lacuna ao articular essa estimativa com uma estratégia de *prefetching* probabilístico voltada a contextos de AIED com conectividade limitada, sem propor uma nova arquitetura de STI nem um novo modelo de estudante.

4. Metodologia

A metodologia adotada é exploratória e baseia-se em simulações computacionais controladas¹. O objetivo não é demonstrar ganhos de aprendizagem, e sim avaliar a viabilidade operacional do OPTIFETCH em comparação com dois comportamentos de referência, sob diferentes cenários de conectividade. Reconhecemos, desde já, que essa estratégia metodológica é compatível com uma prova de conceito e não substitui um estudo empírico com alunos reais (Seção 6).

4.1. Visão Geral das Simulações

A avaliação foi conduzida por meio de simulações computacionais controladas, cujo objetivo foi analisar a viabilidade operacional do OPTIFETCH em diferentes condições de conectividade. Cada simulação representa uma sessão de interação entre um estudante simulado e um STI baseado em LLM, na qual o estudante resolve uma sequência de questões e pode solicitar dicas durante a resolução.

Para cada habilidade, o BKT mantém uma estimativa probabilística de domínio, atualizada a partir dos acertos e erros observados ao longo da sessão. A partir dessa estimativa, calcula-se a probabilidade de erro na interação seguinte, utilizada pelo OPTIFETCH como sinal para decidir sobre o pré-carregamento da dica.

Foram comparadas três estratégias de obtenção de dicas: (i) o *STI On-Demand*, no qual a dica é requisitada somente após a solicitação do estudante; (ii) o *STI com Prefetching Agressivo*, no qual todas as dicas são pré-carregadas; e (iii) o OPTIFETCH, que decide dinamicamente pelo pré-carregamento quando a probabilidade estimada de erro ultrapassa um limiar configurável.

As três estratégias foram avaliadas utilizando as mesmas sequências simuladas de respostas, os mesmos parâmetros do BKT, os mesmos custos operacionais e as mesmas condições de conectividade. Esse procedimento favorece a comparabilidade entre as estratégias, permitindo que as diferenças observadas sejam relacionadas à política de obtenção das dicas, e não a variações nas condições experimentais.

4.2. Comportamentos de Referência

STI On-Demand. No comportamento *On-Demand*, estritamente reativo, a dica só é obtida após solicitação explícita, de modo que toda a latência de rede e de processamento é percebida durante a interação. Tem uso parcimonioso de recursos, mas vulnerabilidade direta à instabilidade de rede, sendo a linha de base reativa do experimento.

STI com Prefetching Agressivo (limite superior comparativo). A segunda referência pré-carrega todas as dicas antes da sessão: a latência percebida é praticamente eliminada, mas o consumo de recursos (bateria, dados, API) é maximizado, mesmo quando muitas dicas nunca são usadas. Esse comportamento não representa o STI médio da literatura, e sim um *upper bound* comparativo, que expressa a pré-busca levada ao extremo sem modulação. A escolha isola o efeito da decisão dinâmica do OPTIFETCH frente a uma alternativa que privilegia responsividade a qualquer custo, sem afirmar que sistemas reais operem assim.

¹Artefatos experimentais utilizados neste estudo: <https://tinyurl.com/4bb64sd2>

OPTIFETCH: Modelo Proposto. Conforme a Figura 1, o agente atua como camada de decisão no laço interno do STI: a cada interação, observa a probabilidade de erro estimada pelo BKT para a habilidade em exercício e decide pré-carregar a dica quando $P(\text{erro}_t) \geq \tau$, obtendo-a sob demanda caso contrário, sendo τ um limiar configurável.

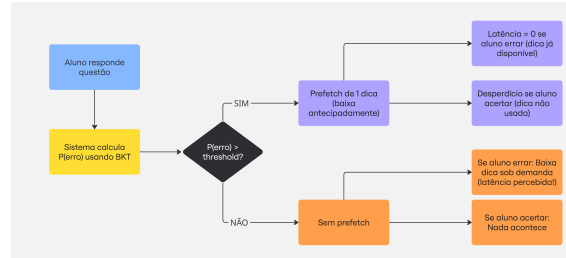


Figura 1. Modelo conceitual do OPTIFETCH..

O comportamento do OPTIFETCH é formalizado segundo o modelo PEAS (*Performance, Environment, Actuators, Sensors*): a *performance* é minimizar a Função de Custo Total (latência percebida, bateria, dados, custo de API e desperdício de *prefetching*); o *environment* é um STI baseado em LLMs sob conectividade instável típica do Sul Global; o *actuator* é realizar ou não o *prefetching* da dica antes da solicitação explícita; e os *sensors* são a probabilidade de erro estimada pelo BKT, a latência média da rede e os parâmetros operacionais do sistema.

4.3. Estimativa de Domínio das Habilidades pelo BKT

Em todas as condições experimentais, o BKT estima, separadamente para cada habilidade, a probabilidade de domínio do estudante simulado. Após cada resposta observada, essa estimativa é atualizada considerando os parâmetros de aprendizagem, acerto sem domínio (*guess*) e erro apesar do domínio (*slip*).

A partir da probabilidade estimada de domínio, calcula-se $P(\text{erro}_t)$, que representa a probabilidade de erro na interação seguinte. Essa probabilidade constitui o sinal de decisão utilizado pelo OPTIFETCH. O modelo não pretende representar o conhecimento do estudante em toda a sua complexidade, tampouco contempla fatores afetivos, motivacionais ou contextuais.

Nas estratégias de referência, a estimativa produzida pelo BKT é mantida para garantir condições experimentais comparáveis, mas não influencia a decisão de obtenção das dicas.

4.4. Modelagem de Custos e Métrica de Avaliação

A avaliação do OPTIFETCH considera cinco componentes operacionais: latência percebida (L), consumo de bateria (B), uso de dados móveis (D), custo financeiro da API (A) e desperdício de *prefetching* (W), correspondente às dicas antecipadas que não foram utilizadas. Esses componentes da Função de Custo Total é definida como:

$$C_{\text{Total}} = w_L \cdot L + w_B \cdot B + w_D \cdot D + w_A \cdot A + w_W \cdot W. \quad (2)$$

Os pesos w_L , w_B , w_D , w_A e w_W controlam a contribuição relativa de cada componente para a Função de Custo Total. Os componentes não foram normalizados: cada um

é agregado em sua unidade original (latência em segundos, bateria em unidades, dados em KB, custo de API em dólares e desperdício em número de dicas), ponderado apenas pelo respectivo peso. Essa informação é necessária porque latência, energia, dados, custo financeiro e desperdício são expressos em unidades e escalas distintas.

Os valores apresentados na Tabela 2 constituem uma configuração de referência adotada para a prova de conceito. Eles não resultam de calibração empírica com estudantes, professores ou instituições educacionais e, portanto, não devem ser interpretados como uma ponderação universal ou ótima.

Tabela 2. Configuração de referência utilizada na Função de Custo Total.

Peso	Valor	Papel na configuração de referência
w_L	1,0	Estabelece a latência percebida como componente de referência da função.
w_B	0,05	Atribui penalidade ao consumo energético associado à obtenção das dicas.
w_D	0,05	Atribui penalidade ao volume de dados móveis transferidos.
w_A	5,0	Atribui maior penalidade relativa ao custo financeiro das chamadas à API.
w_W	0,5	Penaliza dicas pré-carregadas que não são utilizadas.

Como os pesos influenciam diretamente o valor de C_{Total} e a seleção do limiar τ^* , os resultados apresentados neste estudo são condicionados à configuração de referência adotada. Não foi conduzida, nesta prova de conceito, uma análise de sensibilidade capaz de demonstrar que as comparações entre as estratégias se mantêm sob outras ponderações. Consequentemente, os resultados devem ser interpretados como evidências de viabilidade operacional nas condições simuladas, e não como demonstração de uma configuração universalmente ótima.

Como trabalhos futuros, serão investigadas: (i) a variação sistemática dos pesos, com o objetivo de analisar seus efeitos sobre o limiar τ^* e sobre o desempenho relativo das estratégias; (ii) a calibração empírica da função com base em medições de uso real; e (iii) a adaptação contextual dos pesos às prioridades pedagógicas e às restrições de infraestrutura de cada cenário educacional.

4.5. Configuração e Procedimento Experimental

O Apêndice A apresenta os parâmetros utilizados nas simulações. Esses valores foram mantidos constantes entre as três estratégias comparadas, de forma que apenas a política de obtenção das dicas variasse entre as condições experimentais.

Cada condição experimental foi simulada uma única vez, de forma determinística, com semente aleatória fixa igual a 42, agregando 60 estudantes simulados. Cada execução simulou uma sessão composta por 25 questões (75 interações), associadas a uma única habilidade. Foram considerados três perfis de estudante (baixa, média e alta habilidade); para cada perfil, o BKT foi inicializado, respectivamente, com $P(L_0) = 0,25; 0,50; 0,75$, $P(T) = 0,12; 0,10; 0,08$, $P(G) = 0,25; 0,22; 0,18$ e $P(S) = 0,10; 0,08; 0,05$.

A partir do estado latente simulado e dos parâmetros de *guess* e *slip*, foram geradas as respostas observáveis de acerto ou erro. Após cada resposta, a estimativa de domínio da habilidade correspondente foi atualizada pelo BKT. Em seguida, calculou-se a probabilidade de erro na próxima interação, utilizada como sinal de decisão pelo OPTIFETCH.

As estratégias *STI On-Demand*, *STI com Prefetching Agressivo* e *OPTIFETCH* foram submetidas às mesmas sequências simuladas de respostas e às mesmas condições de conectividade. Os cenários Wi-Fi, 4G e rural foram representados, respectivamente, por uma distribuição Normal ($\mu = 0,20$ s; $\sigma = 0,05$ s), uma Log-normal ($\mu = 0,80$ s; $\sigma = 0,30$ s) e uma Weibull ($c = 1,5$; $\mu = 2,50$ s; $\sigma = 0,80$ s). Para cada chamada ao LLM, foram contabilizados 50 KB de dados transferidos, 1,0 unidade de consumo energético e US\$ 0,05 de custo de API, valores base multiplicados por fatores específicos de cada cenário ($\times 1$ no Wi-Fi, $\times 2$ em dados e $\times 3$ em energia no 4G, e $\times 6$ em ambos no cenário rural).

Os três perfis representam condições distintas de uso de um STI. O Wi-Fi corresponde ao cenário de menor latência, como em laboratórios escolares conectados; o 4G representa o acesso por redes móveis; e o cenário rural representa condições de maior latência e instabilidade, recorrentes em regiões remotas. Como esses perfis são estilizados e não derivam de medições de campo, os resultados devem ser interpretados como indicativos do comportamento relativo das estratégias, não sendo diretamente generalizáveis para redes reais.

No *OPTIFETCH*, a dica foi pré-carregada quando a probabilidade de erro estimada pelo BKT satisfizesse a condição $P(\text{erro}_t) \geq \tau$. Caso contrário, a dica permaneceu disponível para obtenção sob demanda após uma eventual solicitação do estudante. O limiar τ foi discretizado no intervalo de 0,01 a 0,99, utilizando incrementos de 0,01. Para cada cenário de conectividade, definiu-se τ^* como o valor que apresentou o menor custo total médio entre as 18 execuções.

Durante as simulações, foram registrados a latência percebida, o consumo de bateria, o volume de dados transferidos, o custo de API, a quantidade de dicas pré-carregadas e não utilizadas e o valor agregado da Função de Custo Total. Os resultados foram sintetizados por meio da média sobre os estudantes e perfis simulados; por se tratar de uma simulação determinística com semente fixa, não foi calculada uma medida de dispersão entre execuções independentes. Esses resultados representam observações operacionais produzidas em um ambiente simulado e não constituem evidências de impacto sobre a aprendizagem.

4.6. Aspectos Éticos

Este estudo baseia-se exclusivamente em simulações computacionais e não envolveu coleta de dados humanos, interação com participantes nem uso de dados pessoais ou sensíveis, sendo todos os dados gerados sinteticamente pelo ambiente de simulação. Por não haver pesquisa com seres humanos, não foi necessária submissão a Comitê de Ética em Pesquisa; eventuais estudos futuros com alunos reais serão precedidos da devida apreciação ética.

5. Resultados e Discussão

Os resultados são apresentados como análise exploratória de *trade-offs*, baseada em simulações controladas, caracterizando o comportamento operacional do *OPTIFETCH* frente às duas referências sob diferentes cenários de conectividade.

5.1. OPTIFETCH e o cenário de Prefetching Agressivo

A primeira comparação confronta o OPTIFETCH com o *prefetching* agressivo, que carrega todas as dicas antes da sessão: elimina a latência percebida na interação, mas ao preço de carregar muitas dicas potencialmente não utilizadas, com impacto sobre bateria, dados, custo de API e desperdício.

Nas simulações, o OPTIFETCH apresentou economia substancial em bateria, dados e custo de API, com reduções entre 82,9% e 83,9% em todos os cenários. O desperdício de dicas pré-carregadas e não utilizadas também foi consistentemente menor: 73,6% no 4G, 68,4% no rural e 73,0% no Wi-Fi. Isso indica que a decisão dinâmica baseada no BKT evita uma fração relevante das dicas que o cenário agressivo carregaria de qualquer forma, ao preço de uma latência percebida um pouco maior, já que parte das dicas continua sendo obtida sob demanda.

Considerada a Função de Custo Total, o OPTIFETCH reduziu o custo agregado em até 39,8% no cenário 4G e 55,5% no cenário rural em comparação com o *prefetching* agressivo (Figura 2, à esquerda). A leitura cautelosa desses números é importante: eles não dizem que o aluno aprendeu mais com o OPTIFETCH, e sim que, dentro da modelagem de custos adotada, o agente encontrou um ponto de operação que preserva responsividade enquanto reduz substancialmente o uso de recursos.

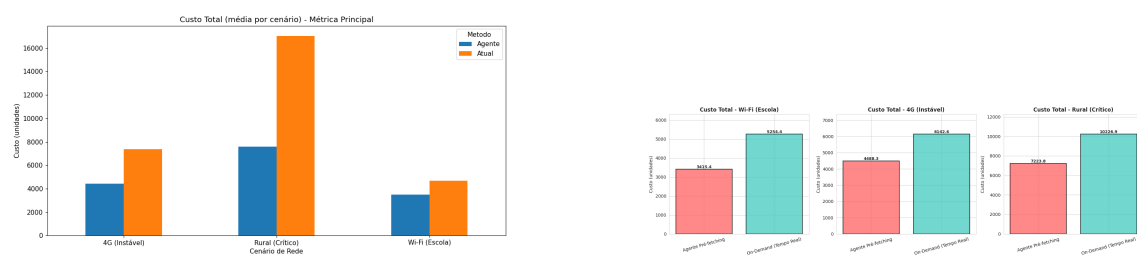


Figura 2. Custo total do OPTIFETCH frente ao *prefetching* agressivo (à esquerda; limite superior comparativo, não operação típica de STIs reais) e ao *On-Demand* (à direita) nos cenários de rede.

5.2. OPTIFETCH e o comportamento On-Demand

A segunda comparação confronta o OPTIFETCH com o *On-Demand*, que não antecipa dicas: elimina o desperdício de *prefetching*, mas paga em latência percebida e custos elevados em redes instáveis, pois cada dica é solicitada e aguardada durante a interação.

Em latência, o OPTIFETCH reduziu 35,3% no 4G, 44,4% no rural e 41,1% no Wi-Fi. Como discutido na Seção ??, ao antecipar dicas quando o BKT estima alta probabilidade de erro, o agente reduz a janela de espera, favorecendo a manutenção do fluxo de interação. Em recursos, o *On-Demand* foi mais econômico em bateria, dados e custo de API, pois baixa dicas apenas quando o estudante erra, enquanto o OPTIFETCH, por antecipar parte das dicas, faz mais chamadas ao LLM. Ainda assim, a redução expressiva da latência percebida — componente dominante da Função de Custo Total — faz com que o OPTIFETCH apresente menor custo agregado, com reduções de 26,9% no 4G, 29,4% no rural e 35,0% no Wi-Fi frente ao *On-Demand*.

O OPTIFETCH apresenta, naturalmente, algum desperdício de dicas: 22,1% no 4G, 25,0% no cenário rural e 23,1% no Wi-Fi. Esse desperdício é o preço pago pela antecipação probabilística e, no balanço da Função de Custo Total, é compensado pela redução substancial dos demais componentes (Figura 2, à direita).

5.3. Síntese das Respostas às Questões de Pesquisa

Com base exclusivamente nos resultados de simulação, a Tabela 3 retoma as três questões de pesquisa. As respostas restringem-se às evidências obtidas e não se estendem a impactos sobre a aprendizagem, que não foram medidos.

Tabela 3. Síntese das respostas às questões de pesquisa.

Questão	Evidência encontrada	Resposta
QP1	Frente ao <i>On-Demand</i> , a latência percebida caiu 35,3% (4G), 44,4% (rural) e 41,1% (Wi-Fi).	Reduz a latência percebida, com ganho maior sob conectividade adversa.
QP2	O custo agregado caiu de 23,8% a 26,5% frente ao <i>On-Demand</i> e de 39,8% a 55,5% frente ao <i>prefetching</i> agressivo, com economia de 82,9%–83,9% em bateria, dados e API.	Mantém o custo operacional controlado nos três cenários, evitando o desperdício da pré-busca indiscriminada.
QP3	Usar a probabilidade de erro do BKT como gatilho posiciona o agente entre o <i>On-Demand</i> e o <i>prefetching</i> agressivo em ambas as métricas, ao preço de algum desperdício (22,1%–23,1%).	A pré-busca seletiva equilibra responsividade e consumo, em vez de otimizar um único eixo.

5.4. Discussão Pedagógica

Os resultados posicionam o OPTIFETCH como uma estratégia intermediária entre o *Prefetching* Agressivo, que privilegia a responsividade com maior consumo de recursos, e o *On-Demand*, que evita antecipações, mas expõe o estudante à latência da rede. Esse equilíbrio é relevante do ponto de vista educacional porque a oportunidade temporal do feedback integra a experiência de interação com um STI.

Quando a espera por uma dica se aproxima de regimes associados à interrupção cognitiva e à ruptura do fluxo de interação [Tolia et al. 2006, Seow 2008, Harmon et al. 2024], a dica, ainda que adequada, pode perder sua oportunidade temporal em relação à atividade que o estudante está realizando. A redução da espera pode, portanto, favorecer a continuidade da interação com o tutor. Entretanto, neste estudo, a latência foi utilizada como uma medida operacional da qualidade da interação, e não como indicador direto de aprendizagem.

A contribuição do OPTIFETCH é, portanto, infraestrutural. O agente busca tornar a dica disponível em tempo oportuno, mas não determina quando ela deve ser apresentada nem substitui as decisões instrucionais do STI. Uma dica pré-carregada deve permanecer oculta até ser solicitada pelo estudante ou liberada segundo a política pedagógica do tutor. Essa separação é importante para preservar oportunidades de elaboração autônoma e evitar intervenções excessivas.

Também deve ser considerado o risco de que a disponibilidade antecipada das dicas influencie seu modo de apresentação. O custo já incorrido na geração de uma dica não deve induzir o sistema a exibi-la apenas para evitar desperdício, pois isso poderia estimular dependência ou comportamentos de uso estratégico das ajudas, como avançar

nas atividades sem envolvimento suficiente com a resolução. Assim, a otimização operacional deve permanecer subordinada aos objetivos pedagógicos e às regras de interação definidas pelo STI.

Essas implicações dialogam com a perspectiva de AIED Unplugged [Isotani et al. 2023], ao tratar a conectividade limitada como requisito de projeto. Contudo, as simulações realizadas não permitem concluir que a redução da latência melhora a aprendizagem, o engajamento ou o desempenho. Estudos futuros com estudantes e professores deverão avaliar se os ganhos operacionais se traduzem em benefícios percebidos, considerando indicadores como tempo e frequência de solicitação de dicas, persistência na tarefa, satisfação, percepção de controle, desempenho e possíveis padrões de dependência das ajudas.

6. Limitações e Ameaças à Validade

Cabe explicitar as limitações do trabalho. Primeiro, a avaliação é integralmente baseada em simulações controladas, sem usuários reais: as reduções de latência e custo refletem o comportamento do modelo nas condições simuladas e não constituem evidência de impacto pedagógico. Segundo, a Função de Custo Total constitui uma escolha de projeto, e os pesos utilizados representam uma configuração de referência, não calibrada empiricamente. Alterações nesses pesos, sobretudo em w_L e w_A , podem modificar o valor de τ^* e o desempenho relativo das estratégias. Como não foi conduzida uma análise de sensibilidade, não se pode afirmar que os resultados se mantêm sob outras ponderações. Terceiro, o *Prefetching Agressivo* é um limite superior comparativo, não um STI real em produção. Quarto, o BKT, embora consolidado, é uma simplificação que não captura fatores afetivos, motivacionais e contextuais do aluno. Quinto, a modelagem de conectividade usa distribuições estilizadas para Wi-Fi, 4G e cenário rural, ao passo que condições reais envolvem variabilidade adicional. Essas limitações delimitam o alcance do estudo: trata-se de prova de conceito sobre viabilidade operacional, não de estudo sobre efeito instrucional.

7. Conclusão e Trabalhos Futuros

Este trabalho apresentou o OPTIFETCH, um agente de *prefetching* probabilístico voltado a Sistemas Tutores Inteligentes que dependem de LLMs em ambientes de conectividade instável. A contribuição central consiste em articular a probabilidade de erro estimada pelo BKT com uma estratégia de antecipação de dicas orientada por uma Função de Custo Total que considera latência percebida, consumo de bateria, uso de dados, custo de API e desperdício de *prefetching*.

Nas condições simuladas e sob a configuração de pesos adotada, o OPTIFETCH reduziu a latência percebida em relação à estratégia *On-Demand* e apresentou menor custo agregado que as duas estratégias de referência. Esses resultados fornecem evidências iniciais de viabilidade operacional do mecanismo, mas não permitem inferir efeitos sobre aprendizagem, engajamento ou desempenho. Sua generalização também é limitada pelos parâmetros sintéticos de conectividade, pela ausência de usuários reais e pela falta de análise de sensibilidade dos pesos da Função de Custo Total.

Como trabalhos futuros, pretende-se realizar a análise de sensibilidade e a calibração empírica da Função de Custo Total; avaliar o OPTIFETCH com estudantes

e professores em contextos com diferentes condições de conectividade; e investigar indicadores como percepção de responsividade, persistência, solicitação e uso de dicas, satisfação e desempenho. Também se pretende integrar o agente a estratégias complementares de AIED Unplugged, como execução local de modelos menores e sincronização eventual com a nuvem.

Declaração sobre Uso de Inteligência Artificial

Em conformidade com as diretrizes do SBIE 2026, declara-se que ferramentas de Inteligência Artificial generativa baseadas em LLM foram utilizadas como apoio na revisão de redação e na organização textual do manuscrito, bem como em auxílio à codificação dos *scripts* de simulação. Todas as decisões científicas, a concepção do agente, o desenho experimental e a análise e interpretação dos resultados foram conduzidas e revisadas integralmente pelos autores, que assumem total responsabilidade pelo conteúdo. A Inteligência Artificial não é autora deste trabalho.

Agradecimentos

Este trabalho foi realizado com apoio financeiro da Fundação Itaú, no âmbito do Edital IA para Educação de 2024, por meio do projeto “Sistema Tutor Inteligente Unplugged com IA Generativa para o Ensino de Física”.

Referências

- Baker, R. S. J. d., Corbett, A. T., and Aleven, V. (2008). More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. In Woolf, B. P., Aïmeur, E., Nkambou, R., and Lajoie, S., editors, *Intelligent Tutoring Systems*, pages 406–415, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Corbett, A. T. and Anderson, J. R. (1994). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4(4):253–278.
- Dakshit, S. (2024). Faculty perspectives on the potential of rag in computer science higher education.
- Granata, F., Poggi, F., and Mongiovì, M. (2025). Enhancing retrieval-augmented generation with entity linking for educational platforms.
- Harmon, K. A., Lee, H., Khasraghi, B. J., Parmar, H. S., and Walden, E. A. (2024). Delays in information presentation lead to brain state switching, which degrades user performance, and there may not be much we can do about it. *MIS Quarterly*, 48(1):273–298.
- Isotani, S., Bittencourt, I. I., Chalco, G. C., Dermeval, D., and Mello, R. F. (2023). Aied unplugged: Leapfrogging the digital divide to reach the underserved. In Wang, N., Rebolledo-Mendez, G., Dimitrova, V., Matsuda, N., and Santos, O. C., editors, *Artificial Intelligence in Education. Posters and Late Breaking Results, Workshops and Tutorials, Industry and Innovation Tracks, Practitioners, Doctoral Consortium and Blue Sky*, pages 772–779, Cham. Springer Nature Switzerland.
- Ko, S.-W., Huang, K., Kim, S.-L., and Chae, H. (2017). Live prefetching for mobile computation offloading. *IEEE Transactions on Wireless Communications*, 16(5):3057–3071.

- Li, Z., Wang, Z., Wang, W., Hung, K., Xie, H., and Wang, F. L. (2025). Retrieval-augmented generation for educational application: A systematic survey. *Computers and Education: Artificial Intelligence*, 8:100417.
- Pavlik, P. I., Cen, H., and Koedinger, K. R. (2009). Performance factors analysis – a new alternative to knowledge tracing. In *Proceedings of the 14th International Conference on Artificial Intelligence in Education (AIED)*, pages 531–538.
- Pelánek, R. (2017). Bayesian knowledge tracing, logistic models, and beyond: an overview of learner modeling techniques. *User Modeling and User-Adapted Interaction*, 27(3):313–350.
- Piech, C., Bassen, J., Huang, J., Ganguli, S., Sahami, M., Guibas, L., and Sohl-Dickstein, J. (2015). Deep knowledge tracing. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 28.
- Rodrigues, L., Guerino, G., Chalco, G., Veloso, T., Oliveira, L., Penha, R., Melo, R., Vieira, T., Marinho, M., Macario, V., Bittencourt, I., Isotani, S., and Dermeval, D. (2023). Teacher-centered intelligent tutoring systems: Design considerations from brazilian, public school teachers. In *Anais do XXXIV Simpósio Brasileiro de Informática na Educação*, pages 1419–1430, Porto Alegre, RS, Brasil. SBC.
- Satyanarayanan, M. (2017). The emergence of edge computing. *Computer*, 50(1):30–39.
- Seow, S. C. (2008). *Designing and Engineering Time: The Psychology of Time Perception in Software*. Addison-Wesley Professional.
- Siris, V., Anagnostopoulou, M., and Dimopoulos, D. (2014). Improving mobile video streaming with mobility prediction and prefetching in integrated cellular-wifi networks.
- Swacha, J. and Gracel, M. (2025). Retrieval-augmented generation (rag) chatbots for education: A survey of applications. *Applied Sciences*, 15(8).
- Tolia, N., Andersen, D. G., and Satyanarayanan, M. (2006). Quantifying interactive user experience on thin clients. *Computer*, 39(3):46–52.
- Van Damme, S., Sameri, M. J., Schwarzmann, S., Wei, Q., Trivisonno, R., De Turck, F., and Torres Vega, M. (2024). Impact of latency on qoe, performance, and collaboration in interactive multi-user virtual reality. *Applied Sciences*, 14(6):2290.
- VanLehn, K. (2006). The behavior of tutoring systems. *International Journal of Artificial Intelligence in Education*, 16(3):227–265.
- Zhao, W., Ge, Y., Qu, W., Zhang, K., and Sun, X. (2017). The duration perception of loading applications in smartphone: Effects of different loading types. *Applied Ergonomics*, 65:223–232.
- Zhao, Y., Laser, M. S., Lyu, Y., and Medvidovic, N. (2018a). Leveraging program analysis to reduce user-perceived latency in mobile applications. In *Proceedings of the 40th International Conference on Software Engineering, ICSE ’18*, page 176–186. ACM.
- Zhao, Y., Wat, P., Laser, M. S., and Medvidović, N. (2018b). Empirically assessing opportunities for prefetching and caching in mobile apps. In *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering, ASE ’18*, page 554–564. ACM.

A. Parâmetros utilizados nas simulações computacionais

Parâmetro	Valor	Descrição
Execuções por condição	1 (60 estudantes)	Execução única determinística por condição, agregando 60 estudantes simulados.
Questões por sessão	25 (75 interações)	Número de questões apresentadas em cada sessão simulada (3 passos por questão).
Número de habilidades	1	Quantidade de habilidades acompanhadas separadamente pelo BKT.
$P(L_0)$	0,25/0,50/0,75	Probabilidade inicial de domínio (baixa / média / alta habilidade).
$P(T)$	0,12/0,10/0,08	Probabilidade de aprendizagem após uma oportunidade de prática.
$P(G)$	0,25/0,22/0,18	Probabilidade de acerto sem domínio da habilidade (<i>guess</i>).
$P(S)$	0,10/0,08/0,05	Probabilidade de erro apesar do domínio da habilidade (<i>slip</i>).
Valores de τ	Otimização Bayesiana em [0,01; 0,99]	τ^* obtido por Otimização Bayesiana (Processo Gaussiano, <i>kernel</i> Matérn $\nu = 2,5$), com 3 pontos iniciais e 15 iterações.
Latência no Wi-Fi	Normal ($\mu = 0,20$ s; $\sigma = 0,05$ s)	Distribuição e parâmetros do cenário de menor latência.
Latência no 4G	Log-normal ($\mu = 0,80$ s; $\sigma = 0,30$ s)	Distribuição e parâmetros do cenário de conectividade móvel.
Latência no cenário rural	Weibull ($c = 1,5$; $\mu = 2,50$ s; $\sigma = 0,80$ s)	Distribuição e parâmetros do cenário de maior instabilidade.
Dados por chamada	50 KB ($\times 1$ / $\times 2$ / $\times 6$)	Volume base por dica, multiplicado por 1 (Wi-Fi), 2 (4G) e 6 (rural).
Consumo energético	1,0 un. ($\times 1$ / $\times 3$ / $\times 6$)	Custo base por dica, multiplicado por 1 (Wi-Fi), 3 (4G) e 6 (rural).
Custo por chamada à API	US\$ 0,05	Custo financeiro atribuído à geração de cada dica.
Semente aleatória	42	Semente fixa (<code>numpy</code> e <code>random</code>) para reprodução das execuções.