LLM optimization in embedded hardware: a systematic review

Lúcio Vagner Carvalho[1], **Carlos Henrique Leite Bianchin**[1], **Marcos Lima Alves**[1], **João Carrett**[1], **Lucas Machado Dias**[1], **Fabricio Barbosa Viegas**[1], **Anderson Priebe Ferrugem**[1]

[1]Grupo de Aplicações de Inteligência Artificial (GAIA) – Universidade Federal de Pelotas (UFPel)
Pelotas, RS – Brasil

{lvcsouza, chlbianchin, mlalves, jcacarrett}@inf.ufpel.edu.br

{lmdias, fbviegas, ferrugem}@inf.ufpel.edu.br

Abstract. *Deploying Large Language Models (LLMs) on the edge faces the Von Neumann bottleneck and severe energy and memory footprint constraints. To map solutions to this challenge, this Systematic Literature Review analyzed 106 primary studies retrieved from IEEE Xplore and Scopus. Results reveal quantization (55.7%) as the dominant compression technique, driven by hardware-software co-design with FPGAs and NPUs to mitigate massive data traffic. Furthermore, this study highlights critical gaps in prefill phase optimization and the underrepresentation of the RISC-V ecosystem, providing an architectural roadmap for future Edge AI research.*

Resumo. *A implantação de Large Language Models (LLMs) na borda esbarra no gargalo de Von Neumann e em severas restrições energéticas e de memória. Para mapear as soluções a este desafio, esta Revisão Sistemática da Literatura analisou 106 estudos primários extraídos do IEEE Xplore e Scopus. Os resultados revelam a dominância da quantização (55,7%), impulsionada pelo co-design com FPGAs e NPUs, como a estratégia mais eficaz para contornar o tráfego massivo de dados. O estudo evidencia lacunas críticas na otimização da fase de prefill e a sub-representação do ecossistema RISC-V, delineando um roteiro arquitetural para futuras pesquisas em Edge AI.*

1. Introdução

A crescente adoção de inteligência artificial na borda (Edge AI) tem impulsionado a necessidade de executar modelos robustos localmente, reduzindo a dependência de infraestrutura em nuvem e garantindo menor latência e maior privacidade. No entanto, a implantação de Large Language Models (LLMs) e arquiteturas baseadas em Transformers em sistemas embarcados esbarra em limitações severas de hardware. Esses dispositivos possuem restrições críticas de área, consumo energético e capacidade de memória, inviabilizando a execução de modelos tradicionais sem adaptações profundas. O gargalo de Von Neumann, que separa fisicamente o processamento da memória [Backus 1978], torna-se o obstáculo central para a inferência eficiente na borda. Superar essas barreiras exige abordagens sofisticadas de co-design de hardware e software, capazes de otimizar o processamento e maximizar a eficiência energética.

Embora soluções em nuvem ofereçam alta capacidade computacional, o crescente interesse em aplicações de borda, como assistentes pessoais offline, dispositivos médicos vestíveis e automação industrial, demanda modelos que operem com pegada de memória reduzida e baixo consumo. Nesse contexto, técnicas de compressão de modelos, quantização de baixa precisão e aceleradores dedicados (FPGAs, NPUs, ASICs) têm emergido como alternativas promissoras. Contudo, a literatura ainda carece de uma visão sistemática e quantitativa sobre quais métodos e plataformas são mais eficazes, bem como sobre as métricas de desempenho verdadeiramente críticas para o ambiente embarcado.

O objetivo desta Revisão Sistemática da Literatura (RSL) é identificar e categorizar as técnicas de otimização de LLMs voltadas para a implementação em dispositivos de hardware embarcado com recursos limitados, buscando responder às seguintes questões de pesquisa (QP):

- QP1: Quais técnicas de compressão estão sendo aplicadas aos modelos implementados em hardware embarcado?
- QP2: Quais plataformas de hardware são os alvos predominantes para aceleração dessas otimizações?
- QP3: Quais métricas de desempenho são reportadas como mais críticas na literatura?

A partir da análise de 106 estudos primários, selecionados em bases como IEEE Xplore e Scopus.

2. Metodologia

Para mapear o estado da arte de um campo em rápida evolução e com vocabulários fragmentados entre as comunidades de Inteligência Artificial e Engenharia de Hardware, esta Revisão Sistemática da Literatura (RSL) adotou um protocolo rigoroso. O desenho do estudo foi estruturado em três fases principais: definição do escopo metodológico, estratégia de busca iterativa e um rigoroso filtro de qualidade acadêmica.

O escopo foi definido através da estrutura PICOC [Booth et al. 2016]:

- Population (P): Large Language Models, incluindo arquiteturas baseadas em Transformers e variantes como GPT e LLaMA.
- Intervention (I): Técnicas de otimização física e em nível de hardware, como quantização, pruning e aceleração via FPGA ou ASICs.
- Comparison (C): Comparação com modelos em nuvem ou implementações padrão sem otimização.
- Outcome (O): Extração de métricas críticas para a viabilidade na borda, com foco em eficiência energética, latência, throughput e memory footprint.
- Context (C): Arquiteturas e sistemas de computação de borda (Edge Computing) e dispositivos embarcados.

2.1. String de Busca

A string de busca foi intencionalmente construída para capturar a convergência entre IA Generativa e metodologias de projeto de silício (CAD). Ao combinar termos como *("Large Language Models"OR "LLM")* com *("Optimization"OR "Software CAD"OR "Hardware description")* e *("FPGA"OR "Hardware acceleration"OR "Embedded")*, o

protocolo garantiu a captura de publicações focadas em fluxos de síntese lógica de alto nível, cujo vocabulário técnico frequentemente diverge da literatura tradicional de algoritmos de aprendizado de máquina.

As buscas foram executadas nas bases IEEE Xplore e Scopus, sendo esta última fundamental para englobar os anais dos principais simpósios internacionais da área (como DAC, DATE e o FPGA Symposium da ACM/SIGDA).

2.2. Critérios de Seleção

Para garantir que o corpus final refletisse exclusivamente os avanços tangíveis em hardware, aplicamos critérios de elegibilidade restritivos. Foram incluídos apenas artigos completos, revisados por pares, que propusessem ou avaliassem otimizações voltadas especificamente para LLMs. Além disso, o estudo precisava detalhar, obrigatoriamente, a implementação em nível de hardware utilizando aceleradores dedicados (como FPGAs, ASICs, SoCs ou NPUs).

Para a seleção dos estudos primários, foram definidos os seguintes Critérios de Inclusão (CI):

- CI1: O artigo propõe, avalia ou revisa técnicas de otimização voltadas especificamente para Large Language Models (LLMs).
- CI2: O estudo detalha a implementação em nível de hardware, abordando o uso de aceleradores dedicados, incluindo FPGAs, ASICs, Sistemas em Chip (SoCs) customizados ou Unidades de Processamento Neural (NPUs).

Em contrapartida, excluímos da análise qualquer literatura cinzenta (como preprints e tutoriais) para manter a integridade científica da revisão. Trabalhos focados exclusivamente em otimização de software para data centers (nuvem) ou que abordassem apenas modelos tradicionais de Machine Learning (como CNNs) foram sumariamente removidos, assegurando o foco absoluto na arquitetura de Transformers na borda.

Estudos foram removidos com base nos seguintes Critérios de Exclusão (CE):

- CE1: Foco exclusivo em otimização de software para execução em nuvem ou GPUs de alta performance para data centers.
- CE2: Estar fora da definição de um artigo científico completo revisado por pares
- CE3: Foco em modelos tradicionais de Machine Learning, não abordando arquiteturas baseadas em Transformers ou LLMs.

A avaliação de qualidade dos estudos primários não se limitou à aderência temática, cada trabalho foi submetido a um rigoroso checklist de avaliação de 1ualidade. O objetivo foi mensurar o valor científico e o rigor experimental das implementações físicas encontradas. Para cada critério, os artigos foram pontuados de forma categórica: Sim (atendimento pleno), parcialmente (atendimento moderado) ou Não (critério ausente ou negligenciado).

A avaliação de qualidade dos estudos selecionados foi baseada em um checklist composto por: Clareza Metodológica, Rigor nas Métricas Físicas, Baselines e Comparação justa, Trade-off de Acurácia, Reprodutibilidade e Detalhamento Técnico.

O checklist foi composto pelos seguintes pilares fundamentais:

- Clareza Metodológica: Avalia se o artigo define de forma explícita os seus objetivos de pesquisa, a arquitetura do modelo-alvo e o fluxo de co-design ou síntese lógica adotado.
- Rigor nas Métricas Físicas: Verifica se há o reporte compulsório e transparente de dados tangíveis de hardware obtidos por simulação precisa ou medição em silício, tais como eficiência energética (GOPs/W ou TOPS/W), latência e pegada de memória.
- Baselines e Comparação Justa: Examina se a proposta é confrontada com soluções de estado da arte consolidadas ou implementações padrão sem otimização sob as mesmas condições de contorno, evitando comparações assimétricas.
- Trade-off de Acurácia: Analisa se o impacto das técnicas de compressão agressivas (como a quantização de baixíssima precisão ou pruning) na capacidade de generalização e qualidade do modelo (ex: perplexidade) é devidamente mensurado e discutido.
- Reprodutibilidade e Detalhamento Técnico: Mede o grau de especificação da plataforma de hardware utilizada (especificações da FPGA, NPU ou ASIC), descrição das ferramentas de CAD e se há menção à abertura do código ou dos fluxos de mapeamento.

3. Resultados

3.1. Seleção dos Estudos

A execução da string de busca nas bases IEEE Xplore e Scopus retornou 489 trabalhos. Após remoção de 42 duplicatas, 447 artigos foram submetidos à triagem por títulos e resumos. Aplicando os critérios de inclusão/exclusão (com ênfase em implementação real de hardware), 156 artigos avançaram para leitura íntegra. Na etapa final, com base no checklist de qualidade (exigindo pelo menos quatro critérios satisfeitos), 106 estudos primários foram aprovados. A Tabela 1 detalha o fluxo por base de dados.

Tabela 1. Resultados do processo de triagem e seleção

Base de Dados	Retornados	Duplicatas Removidas	Analisados	Incluídos
Scopus	353	21	332	91
IEEE Xplore	136	21	115	15
Total	**489**	**42**	**447**	**106**

3.2. Caracterização dos Estudos

A análise dos 106 estudos incluídos revela uma explosão de publicações muito recentes na área, refletindo a urgência de executar modelos massivos na borda: aproximadamente 94% dos trabalhos aprovados foram publicados entre 2024 e 2026. Quanto aos veículos de publicação, o campo é predominantemente impulsionado por conferências e simpósios (cerca de 65% do corpus), seguidos por periódicos científicos da área de hardware e sistemas digitais.

A evolução dos modelos-alvo demonstrou-se evidente. Os trabalhos pioneiros e mais antigos da janela temporal concentravam-se predominantemente na otimização da arquitetura BERT (identificada em 7 implementações), voltada a tarefas de classificação e extração de características na borda,redirecionou seu foco quase absoluto para a geração

de texto ponta a ponta. A arquitetura LLaMA consolidou-se de forma isolada como a principal referência para validações físicas, liderando com 19 implementações em hardware embarcado. Na sequência, destacam-se os modelos da família GPT (presentes em 8 trabalhos) e as arquiteturas compactas comerciais da família Qwen (com 6 trabalhos). Arquiteturas emergentes eficientes de espaço de estados (SSMs), como o Mamba, também começam a aparecer nas acelerações reportadas [Kim et al. 2025].

4. Discussão

A partir do mapeamento quantitativo consolidado na seção anterior, esta seção aprofunda a análise qualitativa das estratégias de aceleração, cruzando as técnicas algorítmicas com as restrições físicas do silício para responder às Questões de Pesquisa sob uma ótica arquitetural.

4.1. Panorama das Técnicas de Otimização (QP1)

A análise do corpus revela que a quantização é a técnica de compressão dominante absoluta, presente em 59 dos 106 artigos aceitos (55,7%). A progressão observada na literatura segue uma trajetória de precisão decrescente: os trabalhos mais antigos concentram-se em INT8, enquanto as propostas mais recentes avançam para INT4, precisões mistas (Mixed-Precision) e esquemas cientes de valores atípicos (Outlier-Aware Quantization).

A segunda técnica mais recorrente é o pruning e esparsidade, presente em 28 artigos (26,4%). A binarização (1-bit) e a otimização de atenção aparecem com 8 artigos cada. Já a destilação de conhecimento é notavelmente sub-representada, com apenas 2 artigos.

4.2. Plataformas de Hardware Predominantes (QP2)

A distribuição dos artigos por plataforma de hardware-alvo reflete a acentuada heterogeneidade da computação de borda. Ressalta-se que a soma das contagens por plataforma supera o total absoluto de 106 artigos do corpus, uma vez que estudos fundamentados em co-design heterogêneo estrito (como arquiteturas híbridas baseadas no particionamento de tarefas entre núcleos CPU ARM e aceleradores programáveis em lógica) são contabilizados em múltiplas frentes simultaneamente. A configuração identificada é apresentada na figura 1:

4.3. Métricas de Desempenho Críticas (QP3)

Enquanto a inferência em nuvem prioriza o *throughput* massivo, a literatura de hardware embarcado foca em métricas delimitadoras físicas. A **eficiência energética** (quantificada em GOPs/W ou TOPS/W) desponta como a métrica suprema para viabilizar operações em dispositivos alimentados por bateria. A latência é avaliada de forma bipartida: *Time-to-First-Token* (TTFT) para a predição inicial e milissegundos por token para a geração autorregressiva. A pegada de memória (*Memory Footprint*) define o limiar de viabilidade da implantação, ditando as agressivas taxas de compressão observadas.

4.4. Análise Cruzada: Técnica Vs Hardware

- **Quantização em FPGAs (O par dominante):** A sinergia decorre da capacidade das FPGAs de realizar operações INT4 e INT2 sem penalidade de ciclo, utilizando empacotamento inteligente de blocos DSP [Wu et al. 2026]. Fluxos de mapeamento de ponta a ponta e arquiteturas orientadas à memória demonstraram

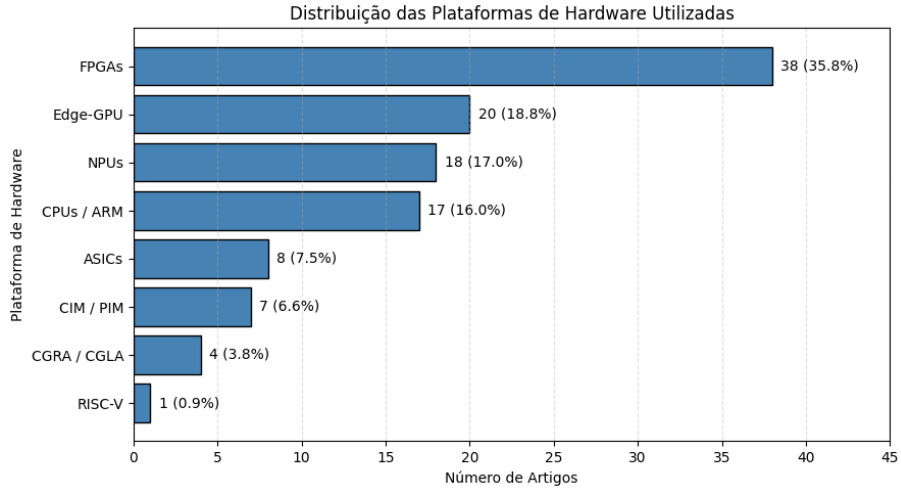


Figura 1. Distribuição de Plataformas de Hardware Utilizadas

ganhos maciços de eficiência [Zeng et al. 2024, Ma et al. 2025, Xu et al. 2024b]. Abordagens recentes também endereçam especificamente o problema de *outliers* de ativação com custo de hardware controlado [Yu et al. 2025, Tang et al. 2023, Liu et al. 2025].

- **Pruning Estruturado em FPGAs e CGRAs:** O pruning estruturado alinha-se naturalmente com arrays sistólicos e pipelines de dataflow espaciais, viabilizando aceleração direta sem custo elevado de indexação [Cao et al. 2023, Zhang et al. 2025].
- **Quantização em NPUs:** O desafio central nas NPUs comerciais não são as multiplicações de matriz (já otimizadas para INT8), mas sim as operações não-lineares, o que tem levado a propostas de hardware focado em aproximações polinomiais de baixo custo [Sadheerthan et al. 2026, Wei et al. 2025, Raha et al. 2025].

4.5. Lacunas e Direções de Pesquisa

A análise cruzada expõe três lacunas sistemáticas no estado da arte atual:

- **L1 Sub-representação de RISC-V:** Apenas 1 artigo foca em RISC-V. Esta lacuna é crítica pois o modelo de licenciamento aberto desta arquitetura viabiliza um co-design profundo entre a ISA e extensões vetoriais (RVV) customizadas especificamente para operações INT4/INT8, algo inviável em ecossistemas fechados como ARM.
- **L2 Otimização da Fase de Prefill:** A vasta maioria dos aceleradores foca exclusivamente na fase de decodificação (*memory-bound*). A fase inicial de processamento do prompt (*prefilling*), que é estritamente delimitada pela capacidade computacional (*compute-bound* via multiplicação de matrizes densas), carece de arquiteturas dedicadas para mitigar o seu gargalo nos limitados recursos da borda [Xu et al. 2024a, Qiao et al. 2026].
- **L3 Falta de avaliação multimétrica simultânea:** A ausência de benchmarks padronizados compromete comparações justas. Uma avaliação madura de aceleradores na borda requer o reporte compulsório e simultâneo da eficiência (GOPs/W),

latência absoluta (P99) e a métrica de degradação da qualidade do modelo pós-quantização (perplexidade), o que raramente ocorre no corpus analisado.

5. Conclusão

Esta Revisão Sistemática da Literatura analisou 106 estudos primários focados na otimização de Large Language Models (LLMs) em hardware embarcado, respondendo a três questões de pesquisa sobre técnicas de compressão, plataformas alvo e métricas críticas. Os principais achados confirmam que a quantização é a técnica dominante (presente em 55,7% dos estudos), especialmente quando aliada a FPGAs e NPUs, onde o co-design de hardware e software é mandatório para acomodar o empacotamento eficiente de blocos lógicos e reduzir o tráfego de dados. Além disso, a urgência em mitigar o gargalo de largura de banda de memória (*Memory Wall*) tem impulsionado soluções disruptivas, como as arquiteturas de *Compute-in-Memory* (CIM).

A principal contribuição deste trabalho, no entanto, reside em expor as fraturas entre o comportamento dos modelos autorregressivos e o estado atual dos aceleradores de hardware. Ao cruzar o panorama técnico com as 106 implementações físicas, identificamos explicitamente o vácuo de cobertura na fase de prefill e o expressivo potencial inexplorado das plataformas abertas RISC-V.

Como direcionamentos concretos para trabalhos futuros com base nas lacunas documentadas, recomenda-se fortemente:

1. O desenvolvimento e a proposição de aceleradores de hardware dedicados à fase inicial de processamento do prompt (fase de prefill);
2. A concepção arquitetural de silício capaz de realizar o roteamento dinâmico de valores atípicos (outliers de ativação) sem impor penalizações críticas de latência ao sistema;
3. A exploração profunda do ecossistema e das extensões vetoriais da arquitetura RISC-V para viabilizar a inferência de ponta a ponta em dispositivos IoT de altíssima restrição.

6. Nota Ética

Ressaltamos que ferramentas de Large Language Models (LLMs) foram empregadas como suporte no processo de escrita e organização textual. Todo o conteúdo técnico, análises e conclusões foram elaborados e validados pelos autores.

Referências

Backus, J. (1978). Can programming be liberated from the von neumann style?: a functional style and its algebra of programs. *Communications of the ACM*, 21(8):613–641.

Booth, A., Sutton, A., and Papaioannou, D. (2016). *Systematic Approaches to a Successful Literature Review*. Sage Publications, London, UK, 2nd edition.

Cao, J., Lin, X., Zhang, M., Shi, K., Yu, J., and Wang, K. (2023). PP-Transformer: Enable efficient deployment of transformers through pattern pruning. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE/ACM.

Kim, J., Lee, J., Lin, J., Kanani, A., Miao, S., Ogras, U., and Park, J. (2025). eMamba: Efficient acceleration framework for mamba models in edge computing. *ACM Transactions on Embedded Computing Systems*.

Liu, F., Yang, N., Wang, Z., Zhu, X., Yao, H., Xiong, X., Sun, Q., and Jiang, L. (2025). OPS: Outlier-aware precision-slice framework for LLM acceleration. In *Proceedings of Design, Automation and Test in Europe (DATE)*. IEEE.

Ma, W., Yang, X., Zeng, S., Liu, T., Shen, L., Wang, H., Li, S., Wang, J., Zhang, Y., Guo, H., Li, J., Zhang, Z., Zhu, Z., Ning, X., Ho, T.-Y., Dai, G., and Wang, Y. (2025). FMC-LLM: Enabling FPGAs for efficient batched decoding of 70b+ LLMs with a memory-centric streaming architecture. In *Proceedings of the 2025 ACM/SIGDA International Symposium on Field Programmable Gate Arrays (FPGA)*. ACM.

Qiao, Y., Chen, Z., Zhang, Y., Wang, Y., and Huang, S. (2026). TeLLMe: An efficient end-to-end ternary LLM prefill and decode accelerator with table-lookup matmul on edge FPGAs. In *Proceedings of the 2026 ACM/SIGDA International Symposium on Field Programmable Gate Arrays (FPGA)*. ACM.

Raha, A., Kundu, S., Sridhar, S. N., Kundu, S., Ghosh, S. K., Palla, A., Das, A., Crews, D., and Mathaikutty, D. A. (2025). LLM-NPU: Towards efficient foundation model inference on low-power neural processing units. In *2025 IEEE International Conference on Omni-Layer Intelligent Systems (COINS)*. IEEE.

Sadheerthan, S., Hsu, M.-J., Huang, C.-H., and Wang, Y.-T. (2026). Attention distribution-aware softmax for NPU-accelerated on-device inference of LLMs: An edge-oriented approximation design. *Electronics*, 15(6):1312.

Tang, J., Guo, C., Leng, J., Liu, Y., Zhang, C., Guo, M., Hu, W., Yang, F., and Zhu, Y. (2023). OliVe: Accelerating large language models via hardware-friendly outlier-victim pair quantization. In *Proceedings of the 50th Annual International Symposium on Computer Architecture (ISCA)*. ACM.

Wei, C.-C., Wang, C.-H., Pong, E.-Y., and Chen, C.-H. (2025). Low-cost non-linear function approximation for transformer deployment on edge NPU. In *2025 1st International Conference on Consumer Technology (ICCT-Pacific)*. IEEE.

Wu, J., Zheng, Z., Gong, L., Wang, C., and Zhou, X. (2026). UDP: A universal DSP packing framework for low-bitwidth MAC acceleration on FPGAs. In *Proceedings of the 2026 ACM/SIGDA International Symposium on Field Programmable Gate Arrays (FPGA)*. ACM.

Xu, D., Zhang, H., Yang, L., Liu, R., Xu, M., and Liu, X. (2024a). WiP: Efficient LLM prefilling with mobile NPU. In *Proceedings of the 2024 Workshop on Edge and Mobile Foundation Models (EdgeFM)*. ACM.

Xu, H., Li, Y., and Ji, S. (2024b). LlamaF: An efficient Llama2 architecture accelerator on embedded FPGAs. In *2024 IEEE 10th World Forum on Internet of Things (WF-IoT)*. IEEE.

Yu, Z., Ji, H., Cao, Y., Wu, J., Yan, X., Zheng, L., and Zou, Z. (2025). DuoQ: A DSP utilization-aware and outlier-free quantization for FPGA-based LLMs acceleration. In *Proceedings of the 62nd Design Automation Conference (DAC)*. IEEE.

Zeng, S., Liu, J., Dai, G., Yang, X., Fu, T., Wang, H., Ma, W., Sun, H., Li, S., Huang, Z., Dai, Y., Li, J., Wang, Z., Zhang, R., Wen, K., Ning, X., and Wang, Y. (2024). FlightLLM: Efficient large language model inference with a complete mapping flow on FPGAs. In *Proceedings of the 2024 ACM/SIGDA International Symposium on Field Programmable Gate Arrays (FPGA)*. ACM.

Zhang, Y., Wang, H., Zhang, G., Chen, J., Yu, J., and Wang, K. (2025). SiST: Token similarity and sparsity aware optimization for transformers on FPGA. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE/ACM.