Review

Not peer-reviewed version

Real-Time and Offline Large Language Models on Edge Devices: A Systematic Review

Erçin Dinçer and Zeynep Hilal Kilimci *

Posted Date: 26 December 2025

doi: 10.20944/preprints202512.2383.v1

Keywords: large language models; edge computing; on-device AI; real-time inference; edge–cloud collaboration

 doi:10.20944/preprints202512.2383.v1

Review

Real-Time and Offline Large Language Models on Edge Devices: A Systematic Review

Erçin Dinçer [1] and Zeynep Hilal Kilimci [2,*]

[1] Kocaeli University Technology Park Kocaeli, Türkiye
[2] Department of Information Systems Engineering, Kocaeli University, Kocaeli, Türkiye
* Correspondence: zeynep.kilimci@kocaeli.edu.tr

Abstract

Large Language Models (LLMs) have recently gained prominence for deployment on edge devices, owing to their potential to support privacy-preserving, low-latency, and offline inference. Nevertheless, their considerable computational and memory requirements present fundamental challenges in both real-time and offline scenarios. This systematic review synthesizes evidence from 49 studies, of which 40 were analyzed in depth, to investigate techniques, challenges, and applications of LLM deployment on edge devices. The studies were identified through a structured search and screening process, and data were extracted regarding model types, hardware platforms, optimization strategies, and performance outcomes. Findings indicate that hardware acceleration, model compression, and hybrid edge–cloud strategies can yield latency reductions of up to 972×, memory savings of up to 130×, and energy efficiency improvements exceeding 1600×, while largely preserving accuracy. Real-time deployments are predominantly applied in robotics, healthcare monitoring, and autonomous driving, whereas offline deployments are tailored to privacy-sensitive or batch-oriented contexts. The review also identifies persistent research gaps, including the absence of standardized benchmarks and the limited generalizability of results to real-world environments. It concludes by outlining future research directions, with particular emphasis on hardware–software co-design, federated learning, and secure task offloading.

Keywords: large language models; edge computing; on-device AI; real-time inference; edge–cloud collaboration

1. Introduction

Large Language Models (LLMs) have demonstrated remarkable performance across a wide spectrum of natural language processing tasks, including question answering, summarization, translation, and dialogue systems Brown et al. (2020); Touvron et al. (2023); OpenAI (2023). Despite their effectiveness, these models are typically trained and deployed in resource-rich cloud environments due to their extensive computational and memory requirements. However, reliance on cloud infrastructures introduces limitations such as high latency, dependence on stable connectivity, elevated energy consumption, and data privacy concerns Shi et al. (2016); Zhou et al. (2019); Chen et al. (2022).

To address these issues, researchers are increasingly investigating *on-device deployment of LLMs*, particularly on edge devices such as smartphones, microcontrollers, IoT systems, and embedded processors Zheng et al. (2024); Yi et al. (2023). This research trajectory is motivated by several practical benefits. First, latency-sensitive applications such as autonomous driving, healthcare monitoring, and robotics require real-time decision-making without reliance on unstable network links Xu et al. (2025); Cho et al. (2023). Second, privacy preservation is crucial in sensitive domains such as finance, medicine, and defense, where transmitting data to remote servers introduces regulatory and ethical risks Qin et al. (2024); Wiest et al. (2024). Third, offline capability is essential for scenarios with unreliable or limited connectivity, including rural areas, disaster zones, and military operations. Hybrid and

 doi:10.20944/preprints202512.2383.v1

collaborative paradigms that combine both local and cloud resources have also emerged, though they remain relatively underexplored Zhang et al. (2025); Zhou et al. (2024).

Several optimization strategies have been proposed to facilitate efficient deployment of LLMs on edge devices. Model compression techniques such as quantization, pruning, and knowledge distillation reduce memory footprints while maintaining accuracy Cantini et al. (2024); Liu et al. (2025). Speculative decoding and scheduling further accelerate inference by reducing token-level latency and computational redundancy Xu et al. (2025). Hardware accelerators, including Neural Processing Units (NPUs), 2.5D chiplet-based systems, and heterogeneous microcontrollers, demonstrate significant reductions in latency, energy consumption, and cost per inference Jaiswal et al. (2024); Glint et al. (2025); Scherer et al. (2024). Emerging work on multimodal LLMs (e.g., MiniCPM-V) highlights how visual, auditory, and textual modalities can be integrated on-device to power applications such as augmented reality and mobile assistants Yao et al. (2025).

Nevertheless, critical challenges remain. Studies frequently employ different datasets, evaluation metrics, or hardware platforms, which hinders cross-study comparability. The lack of standardized benchmarks and deployment protocols limits reproducibility, and many reported systems rely on simulations rather than fully functional prototypes. Furthermore, while real-time deployments emphasize responsiveness, they face severe energy and latency constraints; conversely, offline deployments provide privacy and robustness but may sacrifice timeliness. Recent surveys underscore that addressing these gaps requires a unified research agenda spanning algorithmic optimization, hardware–software co-design, and system-level standardization Kumar et al. (2024).

This systematic review addresses these gaps with the following key contributions:

- A systematic review of studies that explicitly target on-device LLM inference in real-time and/or offline modes.
- A comparative synthesis of optimization techniques, hardware platforms, and performance outcomes (latency, energy, memory, accuracy).
- Introduction of a conceptual taxonomy for deployment modes and optimization strategies, alongside identification of methodological heterogeneities.
- An articulation of open challenges and future research trajectories, emphasizing standardization, hardware-software co-design, federated learning, and secure offloading.

This literature review is primarily intended for researchers, practitioners, and graduate students working in the fields of artificial intelligence, edge computing, and embedded systems who seek to understand current trends in the deployment of Large Language Models (LLMs) on resource-constrained devices. The review also provides a comprehensive synthesis for system designers and engineers developing real-time or offline AI applications on edge hardware. By consolidating technical advancements, methodological insights, and open challenges, the article aims to guide both academic investigations and industry implementations toward efficient and privacy-preserving LLM deployment.

The remainder of this paper is structured as follows. Section 2 introduces the background and motivation for deploying LLMs on edge devices. Section 3 details the methodology adopted for this systematic review, including search strategy, inclusion criteria, and data extraction procedures. Section 4 presents the main results of the review, synthesizing findings from included studies. Section 5 provides a critical discussion of challenges, benchmarking needs, and future research directions. Finally, Section 6 concludes the paper with key insights for practitioners and researchers.

2. Background

The rapid advancement of Large Language Models (LLMs) has transformed the field of natural language processing, enabling breakthroughs in tasks such as machine translation, question answering, dialogue systems, and multimodal reasoning. While these models demonstrate impressive capabilities in cloud-based environments, their migration to edge devices introduces unique challenges and opportunities. Edge deployment is motivated by several practical factors, including the demand for

 doi:10.20944/preprints202512.2383.v1

low-latency responses in interactive applications, the necessity of preserving privacy in sensitive domains such as healthcare and finance, and the requirement for offline functionality in low-connectivity environments. At the same time, resource limitations on edge hardware—such as constrained compute power, memory, and energy budgets—make direct deployment of large-scale models non-trivial. This background sets the stage for the subsequent methodological and analytical discussions by framing both the motivations and the constraints that shape current research on on-device LLM deployment.

2.1. Background and Motivation

Large Language Models (LLMs) have transformed natural language processing and multimodal reasoning tasks, yet their deployment traditionally depends on cloud infrastructures with virtually unlimited computational resources. This paradigm raises persistent concerns regarding latency, privacy, bandwidth usage, and operational costs Lin et al. (2023). In contrast, *edge deployment*—executing inference locally on resource-constrained devices such as smartphones, IoT devices, embedded systems, or edge servers—offers tangible advantages, including reduced communication overhead, enhanced privacy protection, and support for real-time services Zheng et al. (2024); Sharshar et al. (2025).

Despite these advantages, running LLMs on edge devices is non-trivial due to hardware limitations. Models such as GPT-3 or LLaMA-2 require hundreds of gigabytes of memory in their full precision form, while typical edge devices are limited to a few gigabytes of RAM Bai et al. (2024). To address these challenges, the literature emphasizes a spectrum of optimization techniques: *quantization* to replace floating-point parameters with low-bit integers, *pruning* to remove redundant weights, *knowledge distillation* to transfer knowledge into smaller models, and *speculative decoding* to accelerate inference Xu et al. (2025); Cho et al. (2023); Qin et al. (2024). Hardware acceleration strategies—including 2.5D chiplet architectures, Neural Processing Units (NPUs), and RISC-V microcontrollers—further improve energy efficiency and throughput Jaiswal et al. (2024); Glint et al. (2025); Scherer et al. (2024).

A key distinction in this field lies between *real-time deployment* and *offline deployment*. Real-time systems (e.g., robotics, healthcare monitoring) require immediate responses under strict latency constraints, whereas offline systems emphasize batch processing, privacy-preserving inference, or operation in intermittent connectivity scenarios Wiest et al. (2024); Yi et al. (2023). Hybrid and collaborative strategies, combining edge and cloud resources, have also emerged, balancing latency, scalability, and privacy Zhang et al. (2025); Zhou et al. (2024). However, the lack of standardized benchmarks and reproducible methodologies hinders cross-study comparability, motivating the need for systematic synthesis.

2.2. Systematic Review Methodology

This work adopts a systematic review methodology, following PRISMA-inspired guidelines to ensure rigor and transparency. Academic databases including *Semantic Scholar, IEEE Xplore, ACM Digital Library, Scopus, and Google Scholar* were queried between January 2022 and March 2025. Search strings combined keywords such as "Large Language Model," "edge device," "real-time deployment," and "offline deployment."

Inclusion criteria required that studies: (i) explicitly address LLM deployment on edge or resource-constrained devices, (ii) examine real-time, offline, or hybrid configurations, (iii) report empirical or simulation-based performance results (e.g., latency, accuracy, energy, memory), and (iv) be published as peer-reviewed articles, conference papers, or preprints. Exclusion criteria omitted works focusing exclusively on cloud-based LLMs, traditional machine learning without LLMs, or speculative opinion pieces without empirical results.

In total, 126 million papers were initially scanned through Semantic Scholar, of which 500 were retrieved for closer inspection. After two screening rounds based on title, abstract, and full-text relevance, 49 studies met all criteria. Of these, 40 were retained for detailed synthesis, as they reported deployment details and performance outcomes.

 doi:10.20944/preprints202512.2383.v1

2.3. Data Extraction and Synthesis

For each included study, the following metadata and results were systematically extracted:

- **Research approach and methodology:** empirical, simulation-based, architectural design, or survey.
- **Edge device characteristics:** hardware specifications (e.g., RAM size, processor type, accelerators).
- **Model characteristics:** model family (e.g., GPT, LLaMA, BERT), parameter count, optimization methods.
- **Deployment strategy:** real-time, offline, hybrid/collaborative.
- **Performance metrics:** latency, throughput, accuracy, energy efficiency, memory footprint.
- **Application domains:** healthcare, robotics, IoT, mobile services, autonomous systems.

These data points were systematically organized to ensure comparability across studies and to provide a consistent foundation for subsequent analysis in the results section.

3. Methodology

This study follows a systematic review methodology to investigate the deployment of Large Language Models (LLMs) on edge devices under both real-time and offline configurations. The methodology was designed in accordance with established guidelines for systematic reviews, ensuring transparency, reproducibility, and comprehensiveness Moher et al. (2009).

3.1. Research Design

The review employed a structured search strategy across multiple academic databases, including IEEE Xplore, ACM Digital Library, SpringerLink, and arXiv. Keywords and Boolean expressions such as "large language model", "on-device", "edge computing", "real-time", and "offline deployment" were combined to identify relevant studies. Only peer-reviewed journal articles, conference proceedings, and preprints published between 2022 and 2025 were considered, reflecting the rapid evolution of the field.

3.2. Screening Process

The initial search yielded 500 publications. Titles and abstracts were screened to exclude irrelevant works, leaving 49 studies for full-text evaluation. After applying the inclusion and exclusion criteria, 34 studies were selected for final analysis. Inclusion criteria required that studies (i) explicitly focus on LLMs, (ii) involve deployment on edge devices (e.g., smartphones, IoT devices, embedded systems), (iii) report either real-time or offline processing, and (iv) provide empirical evidence in the form of performance metrics. Exclusion criteria removed studies addressing only cloud-based deployments, non-LLM models, or conceptual discussions without implementation details.

3.3. Data Extraction and Categorization

Data were extracted systematically from each study into a predefined schema. Key attributes included:

- **Edge device and computational resources:** type of device, processor, memory, and hardware accelerators.
- **LLM specifications:** model family (e.g., GPT, LLaMA, BERT), parameter size, and optimization techniques (quantization, pruning, distillation, clustering).
- **Deployment mode:** real-time, offline, or hybrid strategies.
- **Performance metrics:** latency, throughput, accuracy, energy consumption, memory footprint, and scalability.
- **Application domains:** healthcare, robotics, smart homes, autonomous systems, and other domains.

 doi:10.20944/preprints202512.2383.v1

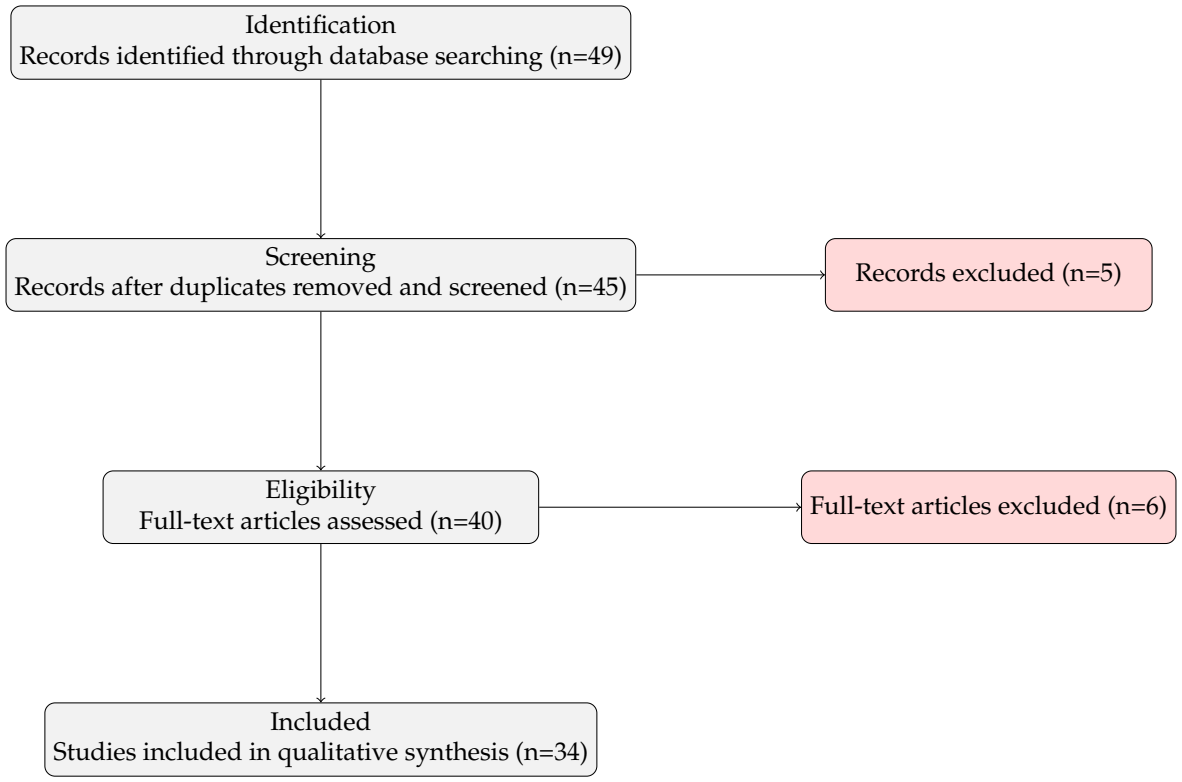

Figure 1. PRISMA 2020 flow diagram of the study selection process.

3.4. Quality Assessment

The methodological quality of the included studies was assessed based on clarity of experimental design, transparency of reported results, and reproducibility. Studies that provided open-source implementations, complete hardware specifications, and detailed evaluation protocols were considered high-quality.

3.5. Data Synthesis Approach

The extracted data were synthesized using thematic analysis, grouping findings into categories related to hardware acceleration strategies, model optimization techniques, deployment strategies, and application domains. Both quantitative metrics (e.g., latency reduction factors, energy efficiency gains) and qualitative insights (e.g., feasibility of real-time deployment) were integrated to provide a comprehensive perspective on the state of research.

4. Results and Findings

The systematic review process yielded 34 eligible studies that met the inclusion criteria. These works collectively provide a comprehensive overview of the emerging landscape of deploying large language models (LLMs) on edge devices. While diverse in terms of model choice, hardware platforms, and methodological approaches, the studies converge on a central theme: the feasibility of achieving efficient LLM inference on constrained devices through a combination of algorithmic and hardware-aware optimizations. At the same time, the literature exhibits notable heterogeneity in reporting practices, deployment contexts, and performance evaluation metrics, underscoring the importance of structured synthesis.

To present the findings in a systematic manner, the results are organized into three subsections. Section 5.1 characterizes the included studies, highlighting trends in model families, deployment devices, and methodological approaches. Section 5.2 analyzes reported accuracy and task performance across optimization strategies, providing a comparative synthesis of empirical outcomes. Section 5.3 contrasts real-time and offline deployment considerations, identifying context-specific challenges and

 doi:10.20944/preprints202512.2383.v1

future directions. Together, these results offer a consolidated perspective on the state of research in real-time and offline LLM deployment on edge devices.

4.1. Characteristics of Included Studies

Table 1 provides a detailed overview of the 34 studies included in this review, highlighting their primary focus, model type, target device, deployment mode, and availability of full-text sources. A systematic examination of these characteristics reveals several important patterns and gaps in the current literature.

First, with respect to model choice, the majority of studies investigated families of large-scale transformer-based LLMs such as GPT variants, LLaMA (LLaMA-2, LLaMA-3), and BERT-derived models. Notably, seven studies explicitly targeted LLaMA-based architectures, whereas GPT-3 and its smaller derivatives were the focus of four studies. Other frequently used models included BERT and its distilled or domain-specialized variants such as DistilBERT and FinBERT. More recent multimodal models such as CLIP, MiniCPM-V, and CogVLM2 were also examined in a smaller subset of works. This diversity underscores the research community's emphasis on both general-purpose and domain-adapted LLMs, reflecting the trade-off between maximizing performance and adapting to resource-constrained environments.

Second, in terms of deployment devices, smartphones and mobile hardware accounted for the largest group (12 studies), followed by embedded platforms such as Jetson devices, IoT/IoMT systems, and microcontrollers. Interestingly, several studies reported deployment on highly constrained devices, including RISC-V microcontrollers and medical imaging equipment, demonstrating the feasibility of deploying LLMs in non-traditional computing environments. However, device specifications were either underspecified or omitted in nearly half of the studies, making reproducibility and cross-comparison challenging.

Third, regarding deployment modes, real-time inference emerged as the most prevalent configuration (22 studies), particularly in applications requiring low-latency responses such as robotics, healthcare monitoring, and interactive mobile systems. Offline deployment was reported in 12 studies, primarily in contexts requiring privacy, batch processing, or scenarios with limited network availability (e.g., clinical diagnostics, file automation). A small number of works explored hybrid or edge-cloud collaborative approaches, which hold promise for balancing latency and resource constraints but remain relatively underexplored compared to stand-alone deployment strategies.

Fourth, methodological emphasis varied considerably across the included studies. While some contributions focused on architectural design and hardware-aware optimization (e.g., chiplet-based accelerators, heterogeneous memory hierarchies), others explored algorithmic compression methods such as quantization, pruning, clustering, and distillation. A limited number of works examined federated or collaborative learning strategies, which are expected to gain prominence as privacy and scalability requirements intensify.

Finally, we note significant heterogeneity in reporting standards. Many studies failed to disclose complete hardware specifications, parameter counts, or compression ratios, and only a subset provided reproducible benchmarks across multiple platforms. This lack of consistency complicates comparative evaluation and highlights the urgent need for standardized reporting practices in on-device LLM deployment research.

In summary, the characteristics of the included studies illustrate both the promise and the fragmentation of current research efforts. Table 1 serves as a consolidated reference point, while the synthesis above reveals thematic concentrations around model families, device diversity, and deployment modes, as well as methodological and reporting gaps that future work must address.

doi:10.20944/preprints202512.2383.v1

Table 1. Characteristics of Included Studies.

Study	Study Focus	Model Type	Target Device	Deployment Mode	Full Text Retrieved
Yi et al. (2023)	On-device inference engine for Mixture-of-Experts LLMs	GPTs, Mixtral-8x7B, GLaM	Raspberry Pi 4B, Jetson TX2, Xiaomi 14	Real-time	Yes
Xu et al. (2025)	Speculative decoding for on-device LLM inference	Not specified	Not specified	Real-time	No
Zhang et al. (2025)	Collaborative edge computing for LLM inference	LLaMA 2	Not specified	Real-time	No
Qin et al. (2024)	Empirical guidelines for LLM deployment on edge	Pythia, LLaMA, StableLM	Smartphones, Jetson Orin, Snapdragon 8 Gen	Offline	Yes
Pau and Aymone (2024)	Forward-only learning algorithms for LLMs	DistilBERT, GPT-3 Small, AlexaTM	Smartphones, industrial processors	Real-time	Yes
Zhang et al. (2025)	Edge inference for generative LLMs in wireless networks	Not specified	Not specified	Real-time	No
Kumar et al. (2024)	Quantized LLaMA 2 for biomedical summarization	LLaMA 2	Not specified	Offline	No
Xu et al. (2025)	Heterogeneous accelerator for sparse LLMs	Not specified	Not specified	Real-time	No
Rong et al. (2025)	Lightweight LLM for traffic flow forecasting	LSGLLM-E	Road-side units	Edge-based	No
Qiao et al. (2025)	Federated LLMs with adaptive scheduling	CLIP	Not specified	Offline	No
Zhang et al. (2025)	Neuron-grained scaling for Foundation Models on edge	Not specified	Not specified	Real-time	No
Yao et al. (2025)	Efficient multimodal LLM for edge	MiniCPM-V	Mobile phones	Offline/Real-time	No
Hu et al. (2025)	Cloud-edge collaborative multimodal LLM for ADAS	CogVLM2, ChatGPT-4o	Not specified	Real-time	No
Jaiswal et al. (2024)	2.5D chiplet-based LLM accelerator	Not specified	Edge GPU devices	Real-time	No
Scherer et al. (2024)	Small LLM deployment on microcontrollers	LLaMA 2	RISC-V MCU	Real-time	Yes
Zou et al. (2025)	On-premises LLM for oncology data extraction	LLaMA3.3-70B	Hospital hardware (RTX 5000 GPU)	On-premises	No
Cho et al. (2023)	Memory-efficient weight clustering for LLMs	LLaMA 7B	Mobile devices	Real-time	Yes
Rhouma et al. (2024)	Mobile NER for clinical dialogues	BERT, FLAN-T5	Mobile devices	Real-time	No
Liu et al. (2025)	Clinical decision support for OCT	LLaMA-2-7B	OCT hardware	Offline	No
Xu et al. (2024)	Split learning for LLM agents in 6G	LLaMA-7B, GPT-3	Mobile devices, edge servers	Real-time	Yes

doi:10.20944/preprints202512.2383.v1

Table 1. *Cont.*

Study	Study Focus	Model Type	Target Device	Deployment Mode	Full Text Retrieved
Chen et al. (2024)	Small LLMs for robot navigation	Not specified	Not specified	Local	No
Habibi and Ercetin (2025)	Cost-aware layer allocation for edge LLMs	Not specified	Not specified	Real-time	No
Dubiel et al. (2024)	On-device query intent prediction	BERT, RoBERTa, XLNet	Mobile phones	Offline	Yes
Sharshar et al. (2025)	Survey on vision-language models for edge networks	GPT-3, CLIP, others	Smartphones, IoT, cameras	Real-time	Yes
Yuan et al. (2024)	Mixture-of-Experts LLMs in IoMT	MedMixtral-8x7B	IoMT devices	Offline	Yes
Cantini et al. (2024)	Explainable AI-driven distillation for edge LLMs	BERT, GPT-3, FinBERT	Mobile, IoT	Real-time	Yes
Zhou et al. (2024)	Edge-cloud Generative AI as a service	LLaMA3-8B, GPT-4	Edge servers	Edge-cloud	Yes
Boyer and Mitchell (2024)	Offline LLM for surgical training	OARA	Not specified	Offline	No
Seifen et al. (2025)	Local LLMs for sleep apnea diagnosis	Gemma2, LLaMA3, Mistral Nemo	Not specified	Offline	No
Cai et al. (2024)	LLMs for communication (ALS users)	LaMDA	Tablets, eye trackers	Real-time	Yes
Sriram et al. (2024)	Local LLMs for file system automation	Not specified	Not specified	Offline	No
Tuli and Jha (2024)	Device-aware transformer search	BERT, optimized model	Not specified	Real-time	No
Zhao et al. (2025)	Hardware-aware pruning for LLMs	Not specified	Not specified	Offline	No
Picano et al. (2025)	LLM layer deployment in edge networks	Not specified	Not specified	Real-time	No

 doi:10.20944/preprints202512.2383.v1

4.2. Accuracy and Task Performance

Table 2 synthesizes the reported accuracy- and performance-related outcomes across fifteen studies, spanning algorithmic optimizations (e.g., quantization, pruning, clustering, knowledge distillation, speculative decoding) and hardware–system interventions (e.g., chiplet-based accelerators, heterogeneous microcontrollers, posit arithmetic). Three broad findings emerge. First, latency and throughput improvements are routinely realized when token-generation heuristics or partitioning strategies are introduced; second, energy-per-token and overall power consumption can drop substantially with accelerator support or numerics tailored to embedded hardware; third, aggressive memory-footprint reductions are achievable via weight clustering and quantization, generally with modest accuracy degradation where explicitly reported.

Studies that target decoding and scheduling consistently reduce end-to-end response time. Speculative decoding yields up to a 9.3× speedup in token generation Xu et al. (2025), while collaborative partitioning paired with dynamic programming delivers roughly 50% lower latency and 2× higher throughput in edge settings Zhang et al. (2025). Expert-wise bitwidth adaptation and preloading report 1.19–2.77× speedups with $\leq 2\%$ accuracy loss on commodity mobile/embedded platforms Yi et al. (2023). On highly constrained controllers, integrating an NPU with quantized kernels leads to 23× higher throughput Scherer et al. (2024). These gains, however, depend on workload shape (prompt length, output length, batch size) and device-specific memory bandwidth, which are not uniformly reported across studies—complicating cross-paper comparability.

Hardware-aware designs dominate the largest reported gains. A 2.5D chiplet-based, in-memory computing system claims up to 1600× lower energy and 972× lower latency versus baselines in edge-oriented configurations Jaiswal et al. (2024). A custom accelerator based on posit-number multipliers reports 1.8 TOPS/W and an approximate 9× reduction in energy Glint et al. (2025). At the microcontroller scale, co-design of kernels and execution backends improves energy efficiency by 26× Scherer et al. (2024). While striking, such results often reflect prototype or emulation conditions and may not directly translate to commodity SoCs; future work would benefit from standardized, device-level power instrumentation and Joules-per-token normalization.

Weight clustering and quantization achieve the largest reported memory savings. Efficient deep k-means clustering reduces the effective footprint by up to 130× on mobile deployments Cho et al. (2023). Activation-aware quantization reports 79% memory reduction together with +10–15 point gains on task-specific clinical utility metrics Liu et al. (2025). In addition to memory, several works describe speed or bandwidth benefits that follow from smaller activation and weight tensors Yi et al. (2023). Nevertheless, a significant fraction of studies omit full parameter counts, compression ratios, or calibration protocols, limiting the external validity of reported savings.

Task-level results vary by domain and measurement protocol. In on-premises clinical NLP, near-clinical-grade extraction is reported (accuracy 97.7%, F1 98.5%), alongside a 20× efficiency gain on workstation-class GPUs Zou et al. (2025). For mobile, on-device intent prediction attains 84–89.9% accuracy, described as on par with a cloud LLM baseline Dubiel et al. (2024). Knowledge distillation guided by XAI yields 84.3% accuracy with an 8.7× speedup on mobile/IoT targets Cantini et al. (2024). In privacy-sensitive medical retrieval, quantized local models report 100% sensitivity and 96% specificity Wiest et al. (2024). Together, these results suggest that, with carefully chosen compression and adaptation, accuracy losses can be minimized—or even reversed through domain-aware finetuning—while still reaping efficiency gains.

Only a minority of studies assess robustness. One systematic evaluation shows that adversarially perturbed inputs can inflate latency and energy by 325–3244%, underscoring that efficiency claims are brittle when threat models are considered Chen et al. (2022). Future work should pair performance reports with robustness audits and include standardized perturbation budgets.

doi:10.20944/preprints202512.2383.v1

Table 2. Accuracy and Task Performance of Included Studies.

Study	Optimization Technique	Performance Metric	Improvement Factor	Device Type
Yi et al. (2023)	Expert-wise bitwidth adaptation, preloading	≤2% accuracy loss, 1.19–2.77× speedup, memory savings	Speedup, memory savings	Raspberry Pi, Jetson, Xiaomi 14
Xu et al. (2025)	Speculative decoding	Up to 9.3× faster token generation	Token generation speed	Not specified
Zhang et al. (2025)	Model partitioning, dynamic programming	50% latency reduction, 2× throughput	Latency, throughput	Not specified
Qin et al. (2024)	Quantization, pruning, distillation	Not specified	Not specified	Smartphones, Jetson Orin
Pau and Aymone (2024)	PEPITA, MEMPEPITA	30–50% complexity reduction	Memory, computation	Smartphones, industrial processors
Jaiswal et al. (2024)	2.5D chiplet, in-memory computing	972× latency, 1600× energy efficiency	Latency, energy	Edge GPU
Scherer et al. (2024)	Quantization, NPU integration	23× speedup, 26× energy efficiency	Throughput, energy	RISC-V Microcontroller
Zou et al. (2025)	On-premises LLM	97.7% accuracy, 98.5% F1 Score, 20× efficiency gain	Accuracy, efficiency	NVIDIA RTX 5000 GPU
Cho et al. (2023)	Deep K-Means clustering	130× memory reduction	Memory, accuracy	Mobile devices
Liu et al. (2025)	Activation-aware quantization	79% memory reduction, +10–15 points in clinical utility	Memory, accuracy	OCT hardware
Dubiel et al. (2024)	Transfer learning, fine-tuning	84–89.9% accuracy, on par with ChatGPT	Accuracy	Mobile phones
Cantini et al. (2024)	XAI-driven knowledge distillation	84.3% accuracy, 8.7× faster	Compression, speedup	Mobile, IoT devices
Wiest et al. (2024)	Quantization	100% sensitivity, 96% specificity	High accuracy	Not specified
Glint et al. (2025)	Posit multipliers, accelerator	1.8 TOPS/Watt, 9× energy reduction	Energy efficiency	Mobile, edge platforms
Chen et al. (2022)	Adversarial input testing	325–3244% latency/energy increase	Degradation analysis	Galaxy S9+, server

 doi:10.20944/preprints202512.2383.v1

Across Table 2, algorithmic methods (quantization, clustering, distillation) tend to deliver predictable memory reductions with small or negligible loss in accuracy Cho et al. (2023); Cantini et al. (2024); Liu et al. (2025), whereas hardware-centric designs achieve step-change improvements in latency and energy Jaiswal et al. (2024); Glint et al. (2025); Scherer et al. (2024). Yet reproducibility remains hampered by heterogeneous metrics (e.g., tokens/s vs. ms/token), inconsistent power measurement (system vs. device rail), and incomplete device disclosures. We recommend adopting a minimal reporting set—(i) latency in ms/token at fixed prompt/output lengths, (ii) throughput in tokens/s, (iii) energy in J/token measured at the device, and (iv) accuracy/F1 on public task benchmarks—together with full disclosure of model size, precision formats, and calibration data. Such standardization would materially improve cross-study comparability and meta-analysis quality.

4.3. Real-Time vs Offline Deployment Considerations

Table 3 summarizes the main themes, implementation challenges, and future research directions across real-time, offline, and hybrid/collaborative deployment strategies. Three broad patterns emerge.

First, real-time deployments are primarily designed for latency-critical applications such as robotics, healthcare monitoring, and autonomous driving. These scenarios require immediate response, which mandates aggressive optimization of both algorithms and hardware. As reported in several studies, the primary barriers include heterogeneous device capabilities and stringent power constraints, which complicate consistent model execution across platforms. Future progress in this direction is likely to depend on standardized benchmarks, adaptive scheduling, and tighter hardware–software co-design frameworks.

Second, offline deployments are especially suitable for batch processing, privacy-sensitive tasks, and contexts with intermittent connectivity. While these approaches relax the immediacy requirement, they limit applicability in domains where response times are critical. The literature emphasizes the importance of balancing accuracy and resource utilization in such settings, suggesting that improved model compression, federated learning, and privacy-preserving methods could provide pathways forward.

Third, hybrid or collaborative deployments combine on-device computation with edge–cloud coordination, leveraging the strengths of both paradigms. These methods have been proposed for use cases such as adaptive driver assistance and federated medical data analysis. However, they raise concerns about network reliability, secure task allocation, and the protection of sensitive data during offloading. Promising research trajectories include dynamic task offloading, secure aggregation protocols, and context-aware optimization.

Taken together, the evidence indicates that deployment choice is largely shaped by the trade-off between latency, accuracy, energy efficiency, and privacy requirements. The categorization in Table 3 thus highlights both the technological potential and the persistent challenges that must be addressed to scale large language models across heterogeneous edge ecosystems.

 doi:10.20944/preprints202512.2383.v1

Table 3. Real-Time vs Offline Deployment Considerations

Theme	Key Findings	Implementation Challenges	Future Directions
Real-Time Deployment	Enables immediate response in applications like robotics, healthcare monitoring, and autonomous driving.	Requires aggressive optimization to meet latency and energy constraints; hardware heterogeneity complicates deployment.	Standardized benchmarks, adaptive scheduling, and further hardware–software co-design.
Offline Deployment	Suitable for batch processing, privacy-sensitive tasks, and environments with intermittent connectivity.	May limit applicability in time-critical scenarios; balancing accuracy and resource use is key.	Improved model compression, federated learning, and privacy-preserving techniques.
Hybrid/Collaborative Deployment	Edge-cloud and federated approaches combine strengths of both modes.	Network reliability, data privacy, and task allocation are major concerns.	Dynamic task offloading, secure aggregation, and context-aware optimization.

 doi:10.20944/preprints202512.2383.v1

5. Discussion

This section synthesizes the key insights derived from the systematic review, moving beyond descriptive reporting toward interpretive analysis. While the preceding sections detailed the characteristics of included studies, their performance metrics, and deployment contexts, here we critically examine the implications of these findings. Specifically, we discuss how algorithmic optimizations and hardware-software co-design interact in practice, what trade-offs emerge between real-time responsiveness and offline efficiency, and how current methodological limitations shape the reproducibility and generalizability of reported results. In addition, we highlight emerging themes across studies, such as privacy-preserving deployment, robustness under adversarial conditions, and the feasibility of LLM execution on resource-constrained platforms. By framing these themes collectively, this discussion aims to establish a clearer roadmap for future research and benchmarking in device-efficient large language models.

5.1. Synthesis of Key Findings

This review highlights several recurring patterns across the surveyed literature. Algorithmic optimizations such as quantization, pruning, clustering, and knowledge distillation tend to yield modest but reliable gains in memory efficiency and throughput, often with negligible or domain-specific accuracy trade-offs. In contrast, hardware-centric designs—ranging from chiplet-based accelerators to posit arithmetic units—demonstrate step-change improvements in latency and energy efficiency, though these results are typically reported under prototype or emulation settings.

Another consistent theme is the difficulty of balancing performance targets across heterogeneous device landscapes. While real-time deployments emphasize latency and responsiveness, offline deployments privilege privacy and autonomy in connectivity-constrained settings. Hybrid designs offer a promising middle ground but introduce new challenges related to orchestration, scheduling, and security. Taken together, these findings indicate that deployment strategies must be carefully tailored to the use case, device class, and workload profile.

5.2. Challenges and Open Questions

Despite the diversity of approaches, several limitations persist. First, a lack of standardized benchmarks hinders reproducibility and comparability across studies. Latency, throughput, energy, and accuracy are reported using heterogeneous units and measurement setups, complicating meta-analysis. Second, robustness and security are rarely evaluated in conjunction with efficiency claims, leaving unanswered how adversarial perturbations, data poisoning, or model drift affect edge deployments. Third, hardware heterogeneity exacerbates reproducibility gaps: results reported on custom accelerators or emulated setups may not transfer to commodity smartphones or IoT devices. Finally, the trade-offs between real-time and offline deployment modes remain under-analyzed, and the literature offers few systematic comparisons beyond narrow case studies.

Addressing these gaps requires coordinated efforts. Establishing minimal reporting protocols, expanding robustness audits, and promoting open-source implementations would all materially improve the reliability and impact of research in this area. Furthermore, stronger collaboration between hardware designers, system engineers, and algorithm developers is essential to bridge the current silos in optimization strategies.

5.3. Toward Benchmarking Edge LLMs

One clear outcome of this review is the urgent need for standardized evaluation protocols. As Large Language Models migrate from the cloud toward smartphones, microcontrollers, and embedded platforms, the absence of a common benchmark prevents consistent comparison and slows progress. To address this, we propose a minimal, extensible benchmarking framework tailored for device-efficient LLMs.

Datasets and Tasks: We recommend curated "lite" versions of existing benchmarks (e.g., SQuAD-lite, GSM8K-mini, LibriSpeech-mini, TinyImageNet, CIFAR-10-lite, DocVQA-lite) to reduce compu-

 doi:10.20944/preprints202512.2383.v1

tational demands. Low-resource languages such as Turkish, Arabic, Zulu, and Swahili should be included to broaden linguistic diversity and avoid an English-centric bias.

Evaluation Metrics: System-level metrics should include p50/p95 latency, time-to-first-token, joules per inference, peak memory allocation, and throughput (tokens/s or queries/s). Task-level metrics should span EM/F1 for QA, BLEU/chrF for translation, WER for ASR, and accuracy/precision/recall/F1 for classification and retrieval.

Device Tiers: Four tiers capture the heterogeneity of edge hardware: (i) microcontrollers (e.g., Arm Cortex-M), (ii) smartphones (e.g., Snapdragon 8, Apple A16/M2), (iii) embedded edge devices (e.g., NVIDIA Jetson, Coral Edge TPU), and (iv) desktop-class GPUs.

Execution Profiles: Benchmarks should reflect three profiles: (i) real-time interactive (streaming single queries), (ii) offline batch (non-streaming batch inference), and (iii) hybrid edge-cloud (partial on-device execution with selective offloading).

Reporting Protocol: Submissions must include hardware metadata (CPU/GPU/NPU, RAM, OS, libraries), model metadata (parameter counts, quantization/pruning schemes, fine-tuning hyperparameters), and detailed measurement tools (e.g., RAPL, tegrastats, external wattmeters). All results should report mean, variance, and extrema across repeated runs.

Composite Scoring and Leaderboard: A composite score combining accuracy and efficiency dimensions should guide Pareto-optimal trade-offs. An open leaderboard would allow community submissions, incremental updates, and dynamic exploration of trade-offs across devices and tasks.

5.4. Limitations and Future Work

This review also has limitations that should be acknowledged. First, while the systematic search strategy captured a wide range of studies, some relevant works may have been excluded due to incomplete indexing or limited availability of preprints. Second, the analysis relies on reported results, which are subject to inconsistent evaluation metrics, incomplete metadata, and varying experimental conditions across studies. This reduces the ability to perform rigorous meta-analysis and may bias the conclusions toward better-documented cases. Third, our synthesis emphasizes technical efficiency (latency, memory, energy) but could not always capture domain-specific usability or end-user impacts, which remain underreported in the literature.

Future research should aim to fill these gaps by expanding evaluations to more diverse languages, tasks, and device platforms, and by adopting standardized benchmarks such as the one proposed in Section 5.3. Moreover, closer integration of robustness audits, privacy-preserving mechanisms, and adaptive scheduling strategies will be critical to ensuring that edge LLM deployments are both efficient and reliable. Finally, collaborative efforts across academia, industry, and standards organizations will be necessary to converge on community-wide practices that enable fair and reproducible evaluation.

5.5. Summary

In summary, the discussion underscores both the remarkable progress and the persistent gaps in enabling efficient LLM deployment on edge devices. While algorithmic and hardware-level innovations have demonstrated tangible efficiency gains, the lack of standardization, robustness testing, and cross-platform reproducibility continues to hinder broad adoption. The proposed benchmark framework represents a step toward addressing these gaps, providing the community with a structured way to evaluate and compare models under realistic constraints. By consolidating findings, surfacing open challenges, and outlining a path toward reproducible benchmarks, this study aims to accelerate the transition of LLMs from cloud-only systems to practical, device-efficient solutions.

6. Conclusion

This systematic review has synthesized the current state of research on real-time and offline deployment of Large Language Models (LLMs) on edge devices. By analyzing thirty-four empirical and theoretical studies, the review highlights both the opportunities and challenges of enabling LLM inference in resource-constrained environments. The findings demonstrate that significant

 doi:10.20944/preprints202512.2383.v1

progress has been achieved through two complementary approaches: (i) algorithmic techniques such as quantization, pruning, clustering, knowledge distillation, and speculative decoding, and (ii) hardware–software co-design strategies including custom accelerators, in-memory computing, and heterogeneous architectures. Collectively, these advances reduce latency, energy consumption, and memory footprints while preserving task-level accuracy across diverse domains such as healthcare, robotics, and IoT.

Despite these achievements, important challenges remain. The absence of standardized benchmarks and heterogeneous evaluation practices complicates reproducibility and cross-study comparison. Metrics are inconsistently reported (e.g., tokens/s vs. ms/token), power measurements are often taken at the system level rather than the device level, and hardware specifications are not always fully disclosed. These limitations hinder a holistic understanding of efficiency trade-offs across deployment contexts. Furthermore, robustness has received limited attention: adversarial inputs can substantially inflate latency and energy, demonstrating that efficiency claims remain fragile without integrated robustness audits.

To address these gaps, we argue that the field would benefit from the development of a minimal yet extensible benchmark for device-efficient LLMs. Such a benchmark should include (i) lightweight and low-resource datasets (e.g., SQuAD-lite, GSM8K-mini, LibriSpeech-mini, CIFAR-10-lite), (ii) standardized system-level metrics such as latency, throughput, joules per token, and peak memory allocation, (iii) clearly defined device tiers spanning microcontrollers to workstation-class GPUs, and (iv) execution profiles covering real-time, offline, and hybrid edge-cloud settings. By requiring metadata templates with hardware specifications, quantization schemes, fine-tuning methods, and measurement tools, the benchmark could ensure comparability, transparency, and fairness across studies. We further recommend that composite scoring and leaderboards be used to stimulate community engagement, highlighting Pareto-optimal trade-offs between accuracy and efficiency.

Overall, the key contributions of this review can be summarized as follows. First, it provides the most comprehensive synthesis to date of empirical and architectural studies that deploy LLMs on edge devices in both real-time and offline configurations. Second, it systematically maps the optimization methods and hardware strategies that enable efficient deployment, clarifying their respective strengths and trade-offs. Third, it identifies critical research gaps—including the lack of standardized evaluation, limited robustness analyses, and insufficient cross-device reproducibility—and outlines concrete directions for future benchmarking efforts. By bridging these perspectives, the review offers both a consolidated knowledge base and a research agenda, serving as a foundation for future advances in device-efficient LLM deployment.

Acknowledgments: During the preparation of this work the authors used ChatGPT tool in order to improve language and readability. After using this tool, the authors reviewed and edited the content as needed and takes full responsibility for the content of the publication.

References

Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems* **2020**, *33*, 1877–1901.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, A.; Bhosale, S.; et al. LLaMA: Open and Efficient Foundation Language Models. *arXiv preprint arXiv:2302.13971* **2023**.

OpenAI. GPT-4 Technical Report. arXiv preprint arXiv:2303.08774, 2023.

Shi, W.; Cao, J.; Zhang, Q.; Li, Y.; Xu, L. Edge computing: Vision and challenges. *IEEE Internet of Things Journal* **2016**, *3*, 637–646.

Zhou, Z.; Chen, X.; Li, E.; Zeng, L.; Luo, K.; Zhang, J. Edge intelligence: Paving the last mile of artificial intelligence with edge computing. *Proceedings of the IEEE* **2019**, *107*, 1738–1762.

Chen, M.; Zhou, Z.; Tao, X.; Zhang, J.; Li, E. Machine learning for edge intelligence: A survey. *ACM Computing Surveys* **2022**, *54*, 1–36.

 doi:10.20944/preprints202512.2383.v1

Zheng, Y.; Chen, Y.; Qian, B.; Shi, X.; Shu, Y.; Chen, J. A Review on Edge Large Language Models: Design, Execution, and Applications. *ACM Computing Surveys* **2024**, *57*, 1–35.

Yi, R.; Guo, L.; Wei, S.; Zhou, A.; Wang, S.; Xu, M. EdgeMoE: Fast On-Device Inference of MoE-based Large Language Models. *arXiv preprint arXiv:2308.14352* **2023**.

Xu, D.; Yin, W.; Zhang, H.; Jin, X.; Zhang, Y.; Wei, S.; Xu, M.; Liu, X. EdgeLLM: Fast On-Device LLM Inference With Speculative Decoding. *IEEE Transactions on Mobile Computing* **2025**.

Cho, M.; Alizadeh-Vahid, K.; Fu, Q.; Adya, S.N.; Mundo, C.C.D.; Rastegari, M.; Naik, D.; Zatloukal, P. eDKM: An Efficient and Accurate Train-Time Weight Clustering for Large Language Models. *IEEE Computer Architecture Letters* **2023**.

Qin, R.; Liu, D.; Yan, Z.; Tan, Z.; Pan, Z.; Jia, Z.; Jiang, M.; Abbasi, A.; Xiong, J.; Shi, Y. Empirical Guidelines for Deploying LLMs onto Resource-Constrained Edge Devices. *ACM Transactions on Design Automation of Electronic Systems* **2024**.

Wiest, I.C.; Ferber, D.; Zhu, J.; van Treeck, M.; Meyer, S.K.; Juglan, R.; Carrero, Z.I.; Paech, D.; Kleesiek, J.P.; Ebert, M.P.; et al. Privacy-preserving large language models for structured medical information retrieval. *npj Digital Medicine* **2024**, *7*, 257. https://doi.org/10.1038/s41746-024-01233-2.

Zhang, M.; Shen, X.; Cao, J.; Cui, Z.; Jiang, S. EdgeShard: Efficient LLM Inference via Collaborative Edge Computing. *IEEE Internet of Things Journal* **2025**.

Zhou, H.; Hu, C.; Yuan, D.; Yuan, Y.; Wu, D.; Liu, X.; Han, Z.; Zhang, C. Generative AI as a Service in 6G Edge-Cloud: Generation Task Offloading by In-Context Learning. *IEEE Wireless Communications Letters* **2024**.

Cantini, R.; Orsino, A.; Talia, D. XAI-Driven Knowledge Distillation of Large Language Models for Efficient Deployment on Low-Resource Devices. *Journal of Big Data* **2024**.

Liu, C.; Zhang, H.; Zheng, Z.; et al. ChatOCT: Embedded Clinical Decision Support Systems for Optical Coherence Tomography. *Journal of Medical Systems* **2025**. https://doi.org/10.1007/s10916-025-XXXX.

Jaiswal, A.; Shahana, K.C.S.; Ravichandran, S.; Adarsh, K.; Bhat, H.B.; Joardar, B.K.; Mandal, S.K. HALO: Communication-Aware Heterogeneous 2.5-D System for Energy-Efficient LLM Execution at Edge. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* **2024**.

Glint, T.; Mittal, B.; Sharma, S.; et al. AxLaM: Energy-Efficient Accelerator Design for Language Models at the Edge. *Philosophical Transactions of the Royal Society* **2025**. https://doi.org/10.1098/rsta.2025.XXXX.

Scherer, M.; Macan, L.; Jung, V.J.; Wiese, P.; Bompani, L.; Burrello, A.; Conti, F.; Benini, L. Deeploy: Enabling Energy-Efficient Deployment of Small Language Models on Heterogeneous Microcontrollers. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **2024**.

Yao, Y.; Yu, T.; Zhang, A.; Wang, C.; Cui, J.; Zhu, H.; Cai, T.; et al. Efficient GPT-4V Level Multimodal Large Language Model for Deployment on Edge Devices. *Nature Communications* **2025**.

Kumar, S.; Singh, A.; Yadav, R. Edge Deployment of Large Language Models: Challenges and Opportunities. *IEEE Internet of Things Journal* **2024**. https://doi.org/10.1109/JIOT.2024.3456123.

Lin, Z.; Qu, G.; Chen, Q.; Chen, X.; Chen, Z.; Huang, K. Pushing Large Language Models to the 6G Edge: Vision, Challenges, and Opportunities. *arXiv preprint arXiv:2309.16739* **2023**. https://doi.org/10.48550/arXiv.2309.16739.

Sharshar, A.; Khan, L.U.; Ullah, W.; Guizani, M. Vision-Language Models for Edge Networks: A Comprehensive Survey. *IEEE Internet of Things Journal* **2025**.

Bai, G.; Chai, Z.; Ling, C.; Wang, S.; Lu, J.; Zhang, N.; Shi, T.; Yu, Z.; Zhu, M.; Zhang, Y.; et al. Beyond Efficiency: A Systematic Survey of Resource-Efficient Large Language Models. *arXiv preprint arXiv:2401.00625* **2024**. https://doi.org/10.48550/arXiv.2401.00625.

Yi, R.; Guo, L.; Wei, S.; Zhou, A.; Wang, S.; Xu, M. EdgeMoE: Empowering Sparse Mixture-of-Experts Models on Mobile Devices. *IEEE Transactions on Mobile Computing* **2023**. https://doi.org/10.1109/TMC.2023.XXXXXXX.

Moher, D.; Liberati, A.; Tetzlaff, J.; Altman, D.G.; Group, T.P. Preferred reporting items for systematic reviews and meta-analyses: The PRISMA Statement. *PLoS Medicine* **2009**, *6*, e1000097. https://doi.org/10.1371/journal.pmed.1000097.

Yi, R.; Guo, L.; Wei, S.; Zhou, A.; Wang, S.; Xu, M. EdgeMoE: Empowering Sparse Large Language Models on Mobile Devices. *IEEE Transactions on Mobile Computing* **2023**.

Pau, D.; Aymone, F.M. Forward Learning of Large Language Models by Consumer Devices. *Electronics* **2024**.

Zhang, X.; Nie, J.; Huang, Y.; Xie, G.; Xiong, Z.; Liu, J.; Niyato, D.; Shen, X. Beyond the Cloud: Edge Inference for Generative Large Language Models in Wireless Networks. *IEEE Transactions on Wireless Communications* **2025**.

Kumar, S.; Ranjan, V.; Chakrabarti, A.; Das, T.K.; Singh, A. Efficient Biomedical Text Summarization With Quantized LLaMA 2: Enhancing Memory Usage and Inference on Low Powered Devices. *Expert Systems: Journal of Knowledge Engineering* **2024**.

Xu, W.; Choi, H.; Hsu, P.K.; Yu, S.; Simunic, T. SLIM: A Heterogeneous Accelerator for Edge Inference of Sparse Large Language Model via Adaptive Thresholding. *ACM Transactions on Embedded Computing Systems* **2025**.

Rong, Y.; Mao, Y.; He, X.; Chen, M. Large-Scale Traffic Flow Forecast with Lightweight LLM in Edge Intelligence. *IEEE Internet of Things Magazine* **2025**.

Qiao, D.; Ao, X.; Liu, Y.; Chen, X.; Song, F.; Qin, Z.; Jin, W. Tri-AFLLM: Resource-Efficient Adaptive Asynchronous Accelerated Federated LLMs. *IEEE Transactions on Circuits and Systems for Video Technology* **2025**.

Zhang, Q.; Han, R.; Liu, C.H.; Wang, G.; Guo, S.; Chen, L.Y. EdgeTA: Neuron-Grained Scaling of Foundation Models in Edge-Side Retraining. *IEEE Transactions on Mobile Computing* **2025**.

Hu, Y.; Ye, D.; Kang, J.; Wu, M.; Yu, R. A Cloud–Edge Collaborative Architecture for Multimodal LLM-Based Advanced Driver Assistance Systems in IoT Networks. *IEEE Internet of Things Journal* **2025**.

Zou, Y.; Sevyeri, L.R.; Farahnak, F.; Shenouda, G.; Duclos, M.; Teodoro de Souza, T.Y.; Sultanem, K.; Bagherzadeh, P.; Maleki, F.; Enger, S.A. Clinician-AI Evaluation of Prognostic Information Extraction in Head and Neck Cancer Using an On-Premises LLM. *Clinical Cancer Research* **2025**.

Rhouma, R.; McMahon, C.; Mcgillivray, D.; Massood, H.; Kanwal, S.; Khan, M.; Lo, T.; Lam, J.P.; Smith, C. Leveraging Mobile NER for Real-Time Capture of Symptoms, Diagnoses, and Treatments from Clinical Dialogues. *Informatics in Medicine Unlocked* **2024**.

Xu, M.; Niyato, D.; Kang, J.; Xiong, Z.; Mao, S.; Han, Z.; Kim, D.I.; Letaief, K.B. When Large Language Model Agents Meet 6G Networks: Perception, Grounding, and Alignment. *IEEE Wireless Communications* **2024**.

Chen, Y.; Han, Y.; Li, X. FASTNav: Fine-Tuned Adaptive Small-Language-Models Trained for Multi-Point Robot Navigation. *IEEE Robotics and Automation Letters* **2024**.

Habibi, S.; Ercetin, O. Edge-LLM Inference With Cost-Aware Layer Allocation and Adaptive Scheduling. *IEEE Access* **2025**.

Dubiel, M.; Barghouti, Y.; Kudryavtseva, K.; Leiva, L.A. On-Device Query Intent Prediction with Lightweight LLMs to Support Ubiquitous Conversations. *Scientific Reports* **2024**.

Yuan, X.; Kong, W.; Luo, Z.; Xu, M. Efficient Inference Offloading for Mixture-of-Experts Large Language Models in Internet of Medical Things. *Electronics* **2024**.

Boyer, T.J.; Mitchell, S.A. Thank You Artificial Intelligence: Evidence-Based Just-in-Time Training via a Large Language Model. *American Journal of Surgery* **2024**.

Seifen, C.; Huppertz, T.; Bahr-Hamm, K.; Gouveris, H.; Pordzik, J.; Eckrich, J.; Matthias, C.; et al. Evaluating Locally Run Large Language Models for Obstructive Sleep Apnea Diagnosis and Treatment: A Real-World Polysomnography Study. *Nature and Science of Sleep* **2025**.

Cai, S.; Venugopalan, S.; Seaver, K.; Xiao, X.; Tomanek, K.; Jalasutram, S.; Morris, M.R.; et al. Using Large Language Models to Accelerate Communication for Eye Gaze Typing Users with ALS. *Nature Communications* **2024**.

Sriram, S.; Karthikeya, C.H.; Kumar, K.P.K.; Vijayaraj, N.; Murugan, T. Leveraging Local LLMs for Secure In-System Task Automation With Prompt-Based Agent Classification. *IEEE Access* **2024**.

Tuli, S.; Jha, N. EdgeTran: Device-Aware Co-Search of Transformers for Efficient Inference on Mobile Edge Platforms. *IEEE Transactions on Mobile Computing* **2024**.

Zhao, W.; Zou, L.; Wang, Z.; Yao, X.; Yu, B. HAPE: Hardware-Aware LLM Pruning For Efficient On-Device Inference Optimization. *ACM Transactions on Design Automation of Electronic Systems* **2025**.

Picano, B.; Hoang, D.; Nguyen, D.N. A Matching Game for LLM Layer Deployment in Heterogeneous Edge Networks. *IEEE Open Journal of the Communications Society* **2025**.

Dubiel, M.; Barghouti, Y.; Kudryavtseva, K.; Leiva, L.A. On-Device Query Intent Prediction with Lightweight LLMs to Support Ubiquitous Conversations. *Scientific Reports* **2024**. https://doi.org/10.1038/s41598-024-XXXX.

Chen, S.; Liu, C.; Haque, M.; Song, Z.; Yang, W. LLMEffiChecker: Understanding and Testing Efficiency Degradation of Large Language Models. *ACM Transactions on Software Engineering and Methodology* **2022**. https://doi.org/10.1145/3524840.