IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 10, May 2025
Impact Factor: 7.67

Enabling On-Device AI for Android: A Dual-Mode System for Offline and Online Large Language Model Inference

Dr. Shailesh Bendale[1], Disha Raskar[2], Akshada Kadam[3], Shlok Jagtap[4], Shivraj Kore[5]
HoD, Department of Computer Engineering[1]
Students, Department of Computer Engineering[2-5]
NBN Sinhgad Technical Institute Campus, Pune, India

Abstract: *As large language models (LLMs) become increasingly important to upcoming applications and use cases, their dependence on cloud infrastructure raises concerns around latency, privacy, and accessibility. This paper presents a inference system that has two modes- offline and online modes for LLM execution. The offline mode leverages a quantized variant of DeepSeek-R1-Distill-Qwen-1.5B using Llama-RN on mobile hardware, while the online mode utilizes the cloud-based Gemma API. Our system, implemented on a consumer-grade smartphone, demonstrates the feasibility of on-device LLM inference without compromising accessibility or efficiency. We discuss implementation strategies, memory considerations, and trade-offs, contributing to the growing field of edge-native LLM deployment on mobile devices.*

Keywords: On-Device AI, Large Language Models, Hybrid Inference, Offline AI, Edge Computing, Llama-RN, Quantized Models, Mobile Inference, React Native, Gemma, Android

I. INTRODUCTION

Large Language Models (LLMs) have become foundational in development of intelligent applications across domains such as education, healthcare, and productivity. However, a lot of these models rely on cloud-based inference, which introduces problems in terms of latency, privacy risks, and dependence on stable internet connectivity. For scenarios where uninterrupted AI access is critical—such as remote environments or privacy-sensitive tasks—there is a growing need to shift toward on-device AI inference. Recent advancements in model compression, quantization, and mobile-optimized runtimes have made it possible to deploy smaller LLMs directly on consumer devices. These models, although limited in capacity compared to their full-scale counterparts, offer the advantage of low-latency, offline operation while preserving the ability to understand the language.

This paper presents a dual-mode system that supports both offline and online LLM inference on mobile platforms. The offline mode is powered by a quantized version of DeepSeek-R1-Distill-Qwen-1.5B, deployed locally via Llama-RN, while the online mode utilizes the Gemma API for enhanced reasoning when internet access is available. The system is implemented using React Native, ensuring cross-platform compatibility and efficient resource usage.

II. LITERATURE REVIEW

Recent developments and innovations in Natural Language Processing have helped in the deployment of LLMs beyond traditional server-based environments. A key research focus has been the optimization of Transformer-based models for on-device inference. Kuchaiev et al. [1] presented optimization techniques like kernel fusion, quantization-aware model training to reduce latency and resource usage for on-device NLP tasks. Their work is foundational for enabling local inference in mobile applications. Federated learning has explored as a paradigm for decentralized model training. Bonawitz et al. [2] and Kairouz et al. [7] contributed significantly to this area by addressing privacy, scalability, and communication efficiency in training LLMs across edge devices. Although our proposed system does not employ federated learning, the privacy-preserving motivation for on-device AI is aligned with these works. Their frameworks

Copyright to IJARSCT
www.ijarsct.co.in

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 10, May 2025
Impact Factor: 7.67

establish the need for local computation in sensitive or offline scenarios, which our hybrid system directly addresses. Quantization techniques remain central to enabling efficient on-device inference. Lin et al. [3] introduced post-training quantization strategies specifically for LLMs, achieving reduced memory usage with minimal accuracy loss. Han et al. [6] further reviewed various techniques for compression of models like quantization, knowledge distillation, and pruning, forming a broad foundation for mobile-friendly AI deployment. These insights directly support our choice to deploy a quantized offline model in a GGUF format.

In terms of model compression and edge deployment, Choi et al. [8] surveyed a wide range of lightweight AI strategies suitable for edge hardware, including mobile phones. Their work highlights both the limitations and emerging solutions in edge-AI deployment, which our system extends by combining compression with a fallback to a more powerful cloud-based model. This dual-mode configuration is not widely explored in existing literature. Security and privacy remain concerning points in android NLP applications. Choquette-Choo et al. [4] emphasized the risks associated with cloud-based inference, including data leakage and inference attacks. This underscores the importance of secure, offline-first systems like ours, especially in privacy-sensitive use cases.

Finally, Huang et al. [5] provided a comprehensive survey on efficient inference techniques for LLMs, analysing trade-offs between computational overhead, memory usage, and latency. They concluded that no single solution fits all scenarios — further justifying our hybrid model that dynamically switches between offline and online modes depending on context. Despite the progress in these individual areas, current literature lacks practical implementations that unify quantized offline inference and cloud-based fallback within a single mobile framework. This work aims to fill that gap by demonstrating a dual-mode system that provides both privacy-conscious, low-latency offline inference and scalable online fallback, enabling real-world usability of LLMs on Android.

III. METHODOLOGY

This study presents an Android-based application for executing both offline and online inference using large language models (LLMs). The system is built using React Native and leverages lightweight model integration for local inference.

A. Offline Inference

The offline inference component of our system is designed to bring large language model capabilities directly onto Android devices, without relying on continuous internet connectivity. For this, we integrated the DeepSeek-R1-Distill model in its quantized GGUF format, which significantly reduces the size and computational demands of the model, enabling it to run smoothly on consumer-grade mobile hardware. Model integration was accomplished using Llama-rn, a React Native wrapper around llama.cpp, which allows seamless communication between the JavaScript layer of the React Native app and the native inference engine. This bridge makes it possible to trigger inference operations directly from the app interface without needing to switch environments or use complex bindings.

To ensure that the model remains persistently available for offline usage, we used the expo-file-system module to store the GGUF model file locally on the device. This approach not only eliminates the need for redownloading the model across sessions but also enhances reliability in environments with intermittent or no internet access. The system was able to load the quantized model and generate coherent responses in real-time during testing, confirming the practicality of running an LLM directly on-device. This offline capability is particularly useful for latency-sensitive applications or use cases in remote areas where connectivity may be limited. This implementation demonstrated that even mid-range smartphones can handle transformer-based inference when appropriately quantized models are used. The Llama-rn integration allowed for a relatively smooth development workflow, bridging native execution with the JavaScript side of the app. Overall, the offline mode supports basic conversational tasks without external dependencies, showcasing the practicality of edge-native language models for lightweight NLP tasks.

B. Online Inference

While offline inference provides autonomy and privacy, some scenarios require access to more powerful models or longer context windows. For these cases, we implemented an online inference fallback using the Gemma API, a cloud-hosted LLM endpoint. This dual-mode design ensures reliability across different network conditions and hardware capabilities.

Copyright to IJARSCT
www.ijarsct.co.in

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 10, May 2025
Impact Factor: 7.67

When the user selects the online mode—or when offline inference exceeds resource limits—the system securely sends the prompt to the Gemma API and returns the generated response in near real-time. This online inference pipeline was designed to be lightweight, minimizing network latency and data transfer to preserve responsiveness. By offloading complex tasks to the cloud when necessary, the system ensures consistent quality of experience while conserving device resources.

C. User Interface

The user interface was designed using React Native components, offering a native-like feel across Android devices. The interface emulates a chat-based interaction model where users input natural language queries or prompts. Based on the selected mode—offline or online—the input is routed to the corresponding inference engine (Llama-rn or Gemma API). We also focused on the user interface design. The interface supports dynamic message rendering, input validation, error handling for failed inference, and mode toggling. It was also optimized for varying screen sizes and performance tiers, ensuring usability on entry-level to high-end smartphones. The intuitive layout makes it suitable for both technical users experimenting with LLMs and non-technical users engaging in daily AI-assisted tasks.

D. Testing and Device Evaluation

The system was tested on a OnePlus Nord CE2 5G device, equipped with 8 GB RAM and 128 GB storage, to evaluate its real-world viability on mid-range Android hardware. The focus was on assessing offline inference stability and usability during normal usage scenarios. The quantized GGUF model was loaded from local storage and executed through the Llama-rn interface. Across multiple sessions, the application maintained smooth interactions, with consistent model response generation and stable memory behaviour. The user interface remained responsive throughout, allowing for uninterrupted chat-based usage.

Overall, the system demonstrated that running a quantized language model on-device is feasible for daily use cases, especially in environments where network access is limited or intermittent. The results support the potential for deploying compact LLMs directly on mobile devices using lightweight integration methods. This also reduces reliance on cloud-based APIs, contributing to enhanced privacy and reduced latency. With further enhancements such as batching or lightweight scheduling, the system could be extended to handle more demanding tasks. Such edge-native solutions mark a meaningful step toward democratizing AI accessibility across diverse device ecosystems.

E. System Architecture

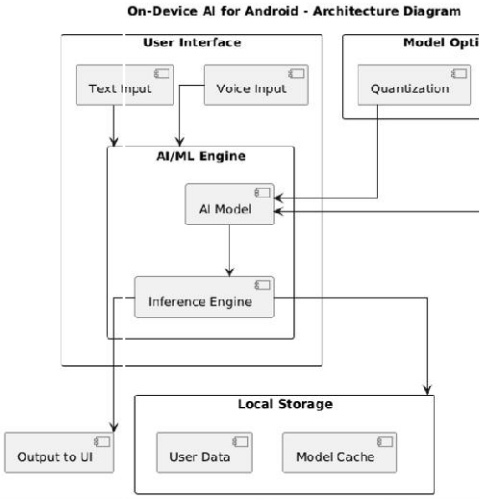

Fig. 1. System Architecture Diagram

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 10, May 2025
Impact Factor: 7.67

The architecture of the proposed application is designed to support both offline and online inference pathways for large language model interaction on Android devices. It comprises modular components that handle model loading, inference routing, and user interaction. The system prioritizes lightweight execution for local inference, while ensuring fallback to cloud-based APIs when necessary. The following diagram illustrates the overall flow and component structure of the system.

IV. RESULTS

To assess the system's practicality, we evaluated both offline and online inference modes on a OnePlus Nord CE2 5G (8 GB RAM, 128 GB storage). The comparison is summarized below:

TABLE I: PERFORMANCE AND RESOURCE COMPARISON: OFFLINE VS ONLINE LLM INFERENCE

Parameter	Offline Mode (DeepSeek-R1)	Online Mode (Gemma API)
Model Size	1.5 GB (INT4 GGUF)	N/A (runs on cloud)
RAM Usage	2.5 GB peak during inference	Minimal (handled by API server)
Internet Required	No	Yes
Test Device	OnePlus Nord CE2 5G	OnePlus Nord CE2 5G
Storage Method	Local storage via Expo File System	No storage needed
Performance	Smooth on-device response for short prompts	Dependent on network latency

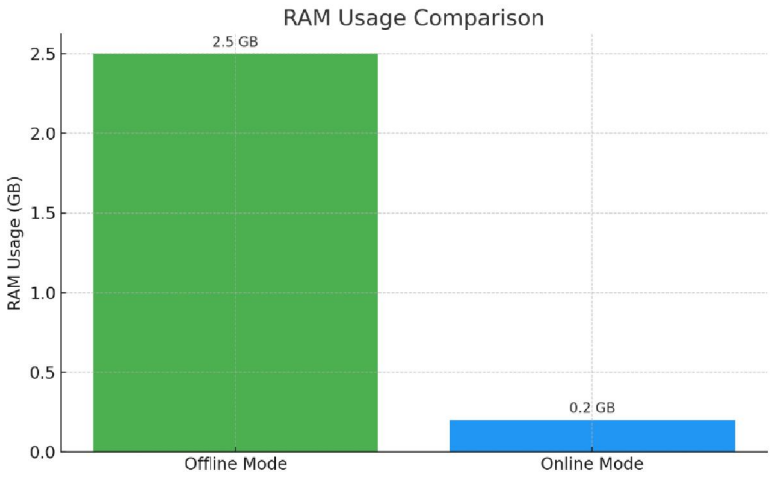

Fig. 2. RAM Usage Comparison Bar Chart

• Interpretation:

The results demonstrate a clear distinction in functionality and resource requirements between the offline and online inference modes. The offline mode, powered by the DeepSeek-R1 model, provides a fully self-contained experience on the Android device, allowing users to interact with a large language model without requiring any internet connectivity. This ensures uninterrupted access in low-connectivity environments and offers a higher level of user privacy, as all processing happens locally without transmitting user data externally. However, this comes with certain trade-offs. The offline setup demands approximately 1.5 GB of storage for the quantized model and peaks at around 2.5 GB of RAM during inference. While this was well-supported on our test device (OnePlus Nord CE2 5G), it may not be feasible for lower-end smartphones with limited memory or storage.

In contrast, the online mode using the Gemma API significantly reduces on-device computation and memory load, enabling broader device compatibility. It acts as a fallback when device constraints prevent local inference. The main drawback of the online mode is its dependency on stable internet connectivity and the variability in response time due

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 10, May 2025
Impact Factor: 7.67

to network latency. Additionally, since data is transmitted to external servers, privacy considerations may become a concern for sensitive use cases. Overall, the hybrid system architecture leverages the strengths of both modes. Offline inference guarantees accessibility, privacy, and independence from connectivity, while online inference ensures service continuity on resource-limited devices or during heavy load. This dual-mode strategy makes the solution adaptable across a wide range of real-world user scenarios.

V. LIMITATIONS

Despite the promising results, the system has certain limitations. The on-device inference is constrained by mobile hardware capabilities, limiting the size and complexity of deployable models. The application currently supports only single-turn, text-based interactions, and lacks support for features like multilingual understanding or multimodal input.
The absence of dynamic switching based on real-time device conditions such as battery level or network availability reduces adaptability. Performance may also vary across different Android devices due to hardware and OS-level differences.

VI. FUTURE SCOPE

The current implementation serves as a foundational step toward bringing LLM capabilities to mobile platforms, but several avenues exist for improvement. Future work could explore more advanced quantization and pruning techniques to deploy even larger models with improved performance. The integration of intelligent model-switching based on network quality and resource availability can enhance user experience further. Additionally, expanding the system to support multilingual capabilities, multimodal inputs, and fine-tuned task-specific behavior could significantly broaden its applicability. Incorporating local data caching, federated fine-tuning, and energy-aware optimization will also be key in scaling on-device AI for real-world adoption.

VII. CONCLUSION

This research demonstrates the feasibility of running large language models on Android devices through a hybrid inference approach. By integrating a quantized version of DeepSeek-R1 for offline inference and leveraging the Gemma API for online access, we provide a seamless and accessible user experience even in limited connectivity scenarios. Our implementation using React Native and Llama-rn highlights the potential for building lightweight, mobile-friendly AI applications without compromising usability.
While our system performs well for basic conversational tasks, it is currently limited by the size and capabilities of quantized models on mobile hardware. Despite this, the ability to run inference offline marks a significant step toward more autonomous, privacy-preserving, and reliable AI systems. This work lays the foundation for further research into optimized mobile deployments, improved model-switching mechanisms, and richer LLM experiences on smartphones.

REFERENCES

[1]. Kuchaiev, Oleksii, et al. "Optimizing Transformer Models for On-Device Natural Language Processing." Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, 2023.
[2]. Bonawitz, Keith, et al. "Federated Learning: Large-Scale Language Model Training on Mobile Devices." Proceedings of NeurIPS 2022: Advances in Neural Information Processing Systems, 2022. DOI:10.48550/arXiv.2211.02868.
[3]. Lin, Jie, et al. "Post-Training Quantization for Language Models." Proceedings of the IEEE Conference on Neural Information Processing Systems (NeurIPS), 2022, pp. 12200-12208. DOI:10.48550/arXiv.2204.09659.
[4]. Choquette-Choo, Christopher A., et al. "Secure Inference for Natural Language Processing Models." Proceedings of the 2022 USENIX Security Symposium, USENIX, 2022, pp. 953-970. DOI:10.48550/arXiv.2111.08400.
[5]. Huang, Minjia, et al. "Efficient Inference of Large Language Models: A Survey." ACM Computing Surveys, vol. 55, no. 3, 2023, pp. 1-36. DOI:10.1145/3517340.

Copyright to IJARSCT
www.ijarsct.co.in

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 10, May 2025
Impact Factor: 7.67

[6]. Han, Song, et al. "Deep Learning Model Compression: Techniques and Applications." IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 1, 2023, pp. 20-35. DOI:10.1109/TPAMI.2021.3089310.
[7]. Kairouz, Peter, et al. "Advances and Open Problems in Federated Learning." Foundations and Trends in Machine Learning, vol. 13, no. 1-2, 2023, pp. 1-210. DOI:10.1561/2200000083.
[8]. Choi, Sangwon, et al. "A Survey on Model Compression Techniques for Edge AI." IEEE Access, vol. 11, 2023, pp. 5981-6000. DOI:10.1109/ACCESS.2022.3209475.
[9]. Jiao, Xiaoqi, et al. "TinyBERT: Distilling BERT for Natural Language Understanding." Proceedings of the 2021 Annual Conference of the Association for Computational Linguistics (ACL), ACL, 2021, pp. 416-425. DOI:10.18653/v1/2021.acl-long.39.
[10]. Liu, Ji, et al. "Mobile Edge AI: Model Optimization and System Design." Proceedings of the IEEE, vol. 111, no. 3, 2023, pp. 444-467. DOI:10.1109/JPROC.2022.3203765.
[11]. Sun, Zhiqing, et al. "MobileBERT: A Compact Task-Agnostic BERT for Resource Limited Devices." Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 2158-2170. DOI:10.18653/v1/2020.acl-main.195.
[12]. Xu, Da, et al. "EdgeBERT: Optimizing BERT for Real-Time Inference on the Edge." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 3281-3290. DOI:10.1109/CVPRW53098.2022.00396
[13]. Wang, Yujun, et al. "Sparsity-Promoting Transformer for Mobile Devices." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023, pp. 5205-5213. DOI:10.1109/CVPRW57021.2023.00114.
[14]. Zhang, Haotian, et al. "LLM in a Flash: Efficient Large Language Model Inference with Limited Memory." arXiv preprint arXiv:2309.15567, 2023. DOI: 10.48550/arXiv.2309.15567