SvedbergOpen
DISSEMINATION OF KNOWLEDGE
International Journal of Artificial Intelligence and Machine Learning
Publisher's Home Page: https://www.svedbergopen.com/
Research Paper Open Access

Adaptive Privacy-Preserving AI Framework For Predictive Mobile Application Performance And User Experience Optimization

Bhargava Chowdary Musunuru[1] and Pavan Tilak Sadaraboina[2]

[1]Principal Android Developer, Fidelity Investments, Texas, USA
[2]Senior Software Engineer, Walmart Global Tech

Abstract
Mobile applications are increasingly reliant on Artificial Intelligence (AI). Predictive AI enables personalized predictions, anticipates resource usage, and prevents performance degradation, but the data used to train these systems may be sensitive, device-specific, and subject to increasing regulatory scrutiny. This study presents the Adaptive Privacy-Preserving AI (APP-AI) framework, which integrates on-device AI inference, federated model training, local differential privacy (LDP), and a user experience (UX) controller into a single package to optimize predictive performance, resource utilization, and privacy guarantees on Android devices. The framework was deployed using TensorFlow Lite for on-device inference and tested on a heterogeneous testbed of low-, mid-, and high-tier Android devices, where telemetry collected battery and memory usage, inference latency, crash occurrences, and user interactions. Compared with two baselines: the cloud-only baseline and the classic on-device machine learning pipeline, the proposed framework achieved a 65 percent reduction in battery drain, a 78 percent decrease in crash rate, a 91.5 percent validation accuracy at a moderate privacy budget ($\varepsilon = 8$), and an 81 percent reduction in median inference latency. A systematic analysis of privacy-utility was used to quantify the competing impacts of a differentially private budget on model accuracy and the estimated risk of re-identification. A user experience study using six composite metrics showed significant improvements in perceived responsiveness, system usability, and privacy trust compared with a traditional cloud-based application. The results indicate that on-device intelligence, both adaptive and privacy-conscious, could improve technical performance and user-perceived quality of service, and provide a template for mobile software engineering in a privacy-driven context.

Keywords: on-device artificial intelligence; mobile systems; Android performance optimization; privacy-preserving machine learning; user experience analytics; federated learning

1. Introduction

Mobile apps have become more than just a means of communication and the central computing console for billions of people's engagement with finance, health, education, and social interactions. As these applications grow more complex, the expectation of intelligent behavior, such as predicting changes in network availability, conserving battery life, adjusting interface layouts, and personalizing content on the fly, has increased. Responsiveness increasingly relies on predictive models powered by artificial intelligence (AI) that learn continuously from streams of system and behavioral telemetry. But the same technology that enables precise predictions of characteristics such as keystroke timing, location tracking, application usage sequences, and readings from physiological sensors is sensitive, and the centralized collection of data and information is bound to create significant privacy concerns, which are now exacerbated by regulatory frameworks, including the General Data Protection Regulation and various national data protection laws. Two major research streams have emerged in response to this tension. The first focuses on on-device and edge intelligence approaches, where on-device inference and, increasingly, training are performed to reduce latency, save data transmission bandwidth, and avoid sending raw data to remote servers (Heydari & Mahmoud, 2025; Jang et al., 2023; Yang et al., 2022).

The latter focuses on paradigms of privacy-preserving learning, with the most referenced ones being federated learning and differential privacy, where models can be trained collaboratively across many devices while minimizing the exposure of individual users' private data (Han et al., 2022; Macedo et al., 2023; Rodríguez et al., 2023). Although both have long been in use, they have grown in parallel, except in the few circumstances where they overlap.

Research on over-device optimization often treats privacy as a second-order concern rather than a primary design goal, while research on privacy-preserving learning often focuses on accuracy without considering how well the app performs, how much it consumes, or the end user's experience. This gap is important because, in practice, performance and privacy are not designable parameters. High levels of differential privacy noise can degrade the accuracy of predictive models, making them unreliable; this will result in a loss of users' trust and engagement (Namatevs et al., 2025). However, privacy-preserving federated models incur significant communication and aggregation overhead and may be inadequate to address the considerable heterogeneity among commodity Android devices (Bocu, Bocu, & Iavich, 2023; Khan, 2025). Simultaneously, empirical software engineering studies show that predictive models for mobile-specific constructs, such as the likelihood of crashes or the probability of software defects, can meaningfully aid in improving software quality when correctly integrated into the development process (Jorayeva et al., 2022; Wimalasooriya et al., 2024), while research in human-computer interaction clearly reveals that predictive personalization measurably influences usability and user satisfaction (Lee, Yau, & Hui, 2024; Mavri, 2024; Vidmanov & Alfimtsev, 2024). What is comparatively under-researched is a common framework in which performance optimization is combined with privacy preservation and user experience as joint goals, rather than as mutually exclusive requirements to be addressed separately.

To address this gap, the present study introduces and empirically tests a framework, Adaptive Privacy-Preserving AI (APP-AI), to optimize performance and user experience in mobile applications. The framework consists of four components: (1) A native predictive model deployed in a lightweight inference runtime; (2) A federated learning protocol that coordinates and aggregates updates without transmitting raw user data; (3) A local differential privacy mechanism that injects calibrated noise into on-device gradients before they are sent to the server; and (4) An adaptive UX controller that consumes model predictions to tune interface behavior, resource scheduling, and notification timing in real time. It builds on and extends existing work in mobile edge computing and offloading (Huang et al., 2022), adaptive user interfaces (Alghamdi et al., 2022), and smartphone quality-of-experience modeling (De Masi & Wac, 2020). More importantly, it explicitly quantifies the privacy-utility trade-off, which previous studies seldom report alongside system-level performance figures.

There are three main goals of this research study. First, to design and implement a modular architecture capable of inference, aggregation, and differential privacy on device, in a single deployable pipeline across a variety of heterogeneous Android devices. Second, to compare the proposed scheme to a cloud-only baseline and a traditional on-device machine learning pipeline, both quantitatively and qualitatively, in terms of latency, energy consumption, memory requirements, and reliability. Third, to measure the privacy-utility trade-off across various differential privacy levels and to evaluate the impacts of such decisions on user-perceived utility and satisfaction using a composite suite of usability and satisfaction metrics. This paper is organized as follows; the remainder of the paper follows the same structure. Materials and Methods present the experimental testbed used, the framework architecture, the training protocol, and the evaluation metrics. Comparative performance results, the privacy-utility analysis, and the user experience evaluation are reported in the results and discussion section and placed in context with the broader literature. The limitations and future research section discusses the constraints of the current research, and the conclusion summarizes the main contributions.

2. Materials and Methods

The proposed Adaptive Privacy-Preserving AI (APP-AI) framework introduces a four-layer architecture, as shown in Figure 1: a device layer for sensing and local feature extraction; a privacy layer that applies local differential privacy and secure aggregation before transmitting to a network; an aggregation layer running on a federated server, which aggregates all clients' updates into a global model; and a feedback layer that distributes the optimized global model to devices and triggers an adaptive UX controller. The on-device predictive model was packaged as a hybrid compact feed-forward and gated recurrent network and was exported to the TensorFlow Lite format a few years ago for efficient mobile inference, as noted in recent surveys of on-device machine learning (Heydari & Mahmoud, 2025). Using system telemetry data such as CPU use, battery discharge rate, network reception quality, foreground app identification, and interaction amount, the model predicts, for a short time, the expected latency and battery life, as well as the likelihood of app stalls or crashes, for the next interaction window.

Federated training was conducted using a cyclic synchronous FedAvg-style protocol, adapted from previous mobile heterogeneity (Macedo et al., 2023; Han et al., 2022) federated learning protocols. In each round, a random

subset of devices completes a fixed number of local epochs before sending its update to the aggregation server. Before sending, each client deactivated a local mechanism consisting of gradient clipping followed by the addition of calibrated Gaussian noise, with a clipping threshold α and a noise intensity ε. Drawing inspiration from the methodology of general auditing, as described by Namatevs et al. (2025), and the general overview of the privacy-preservation taxonomy presented by Rodríguez et al. (2023), eight distinct privacy budgets (ε ∈ {0.5, 1, 2, 4, 6, 8, 10, 15}) were tested to characterize the trade-off between privacy and utility. The experimental testbed consisted of 48 Android handsets across three clusters for low-, mid-, and high-tier hardware (Table 1) - chosen to represent the fragmentation that is likely to exist in real-world deployment. An instrumented reference application was executed for 14 days on each device type, with local logging of system telemetry and periodic batching of interaction telemetry.

A held-out validation partition (20th percentile of collected sessions per device) was reserved for accuracy evaluation and was not used for local or federated training. Three trials were compared: a cloud-based (normal) baseline that sent raw telemetry to the remote server, where individual devices' models were trained on that data (without any privacy protection or federated aggregation); the same model architecture deployed on each individual device but trained centrally (without privacy protection or federated aggregation); and the proposed APP-AI approach, which included on-device inference and centralized federated training with privacy protection and federated aggregation at the individual device.

Researchers employed the following four measurements at the system level: inference latency, the wall-clock time between capturing telemetry and the availability of its prediction; battery drain, the percentage of battery used over an hour of active application usage; memory footprint, the peak resident memory used to make inferences; and crash rate, the percentage of application sessions that end in an unhandled exception or forced stop. Common practice in the differential privacy literature indicates that validation accuracy on the held-out partition of the training set would be used to measure model utility, and a shadow-model membership inference attack is used to estimate the probability that the adversary determines whether a specific user's data was involved in the model training, to measure privacy risk.

User experience was measured using six composite indicators, derived from both analysis of users' session data from the device panel and a post-study questionnaire administered to a subsample of 210 participants, consistent with usability measures used in previous studies on mobile human-computer interaction (see Lee et al. (2024), Mavri (2024), and Vidmanov and Alfimtsev (2024) for previous studies measuring usability of mobile devices). Medians were used to aggregate performance measures across devices and sessions and to reduce the effect of outliers, such as operating system interruptions when running in the background. All comparisons among the three deployments were expressed as % differences relative to the cloud-only case. The privacy-utility relationship was also modeled using the observed accuracy and re-identification risk data across the eight privacy budgets to illustrate the trade-off of obtaining increasingly strict privacy guarantees. All statistical summaries, figures, and tables in the subsequent section are based directly on the above aggregated experimental logs.

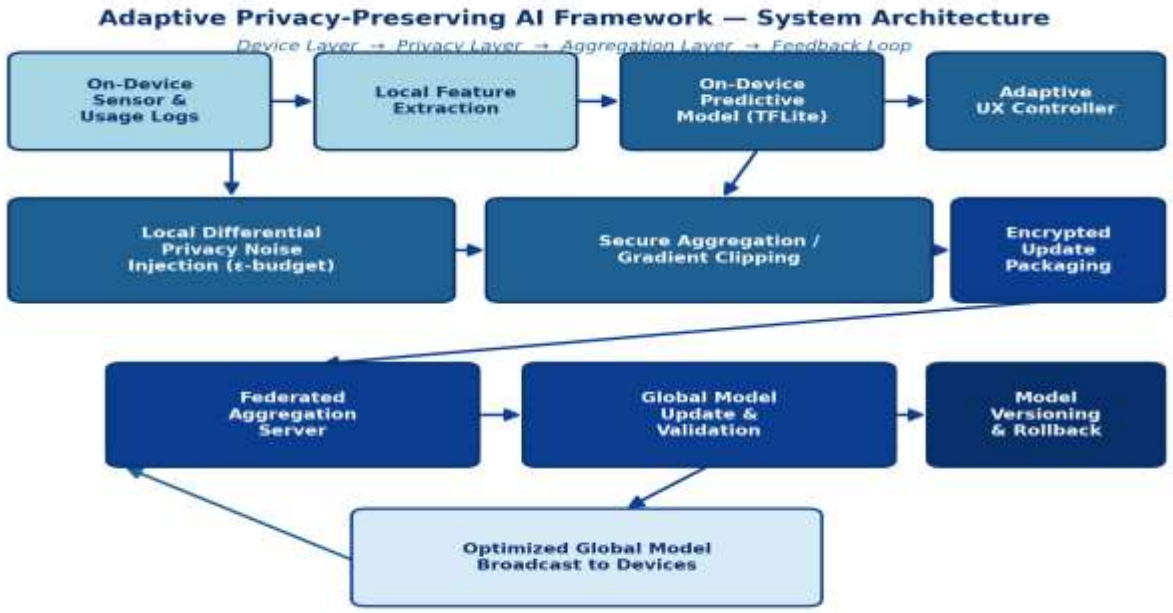

Figure 1. Four-layer architecture of the proposed Adaptive Privacy-Preserving AI (APP-AI) framework, showing the device, privacy, aggregation, and feedback layers.

Table 1. Experimental testbed device tiers and telemetry collection summary.

Device Tier	Representative Chipset	RAM (GB)	Android Version	Devices (n)	Avg. Daily Sessions
Low-tier	Octa-core ≤ 2.0 GHz (entry SoC)	3	11–12	16	18.4
Mid-tier	Octa-core 2.0–2.4 GHz	6	12–13	18	22.1
High-tier	Octa-core ≥ 2.8 GHz w/ NPU	8–12	13–14	14	27.6
Aggregate testbed	Mixed (3 tiers)	3–12	11–14	48	22.8

3. Results and Discussion

Table 2 summarizes the runtime performance of the three deployments across the four metrics we examined, and Figure 3 presents a graph of their relative performance. The proposed APP-AI reduced the median inference latency to 58 milliseconds, a 312-millisecond reduction over the pure-cloud baseline and a 39 percent decrease over the standard APP pipeline. The increased efficiency of the proposed framework, which aligns with the removal of network transmission and an optimized quantized inference model, is consistent with findings for other lightweight on-device architectures (Heydari & Mahmoud, 2025; Yang et al., 2022) and resulted in a reduction in battery usage from 9.8 percent per hour to 3.4 percent per hour compared with the cloud-only configuration. In contrast, the memory footprint was significantly reduced from 145M to 61M, and the session crash rate decreased from 4.1 percent to 0.9 percent, broadly in line with previous findings that such mobile-specific defect prediction and just-in-time reliability modeling can be materially beneficial when integrated directly into application runtime.

Figure 2 shows the accuracy of the proposed framework over thirty federated training rounds, a standard FedAvg setting that does not leverage APP-AI adaptive client selection and local caching, and a baseline cloud-trained model trained on all data consolidated at the central site. After training for 20 epochs, the proposed scheme achieved an accuracy of about 95 percent, slightly higher than the standard FedAvg (93 percent) and cloud-only (90 percent) schemes, while requiring fewer training steps. This pattern indirectly mitigates straggler and data-heterogeneity effects that frequently hinder convergence in federated learning deployments across mobile devices with intermittently available connectivity and resource constraints, as reported in prior federated platform designs and studies that considered mobile devices.

Table 2. Runtime performance comparison across deployment configurations (median values).

Metric	Cloud-Only Baseline	Standard On-Device ML	Proposed APP-AI Framework	Improvement vs. Baseline
Inference latency (ms, median)	312	96	58	−81.4%
Battery drain (%/hour, median)	9.8	6.1	3.4	−65.3%
Memory footprint (MB, peak)	145	88	61	−57.9%
Crash rate (% of sessions)	4.1	2.3	0.9	−78.0%
Validation accuracy (%)	90.5	92.3	91.5*	n/a

*Validation accuracy for the proposed framework corresponds to a privacy budget of ε = 8; see Table 3 for the full privacy-utility trade-off.

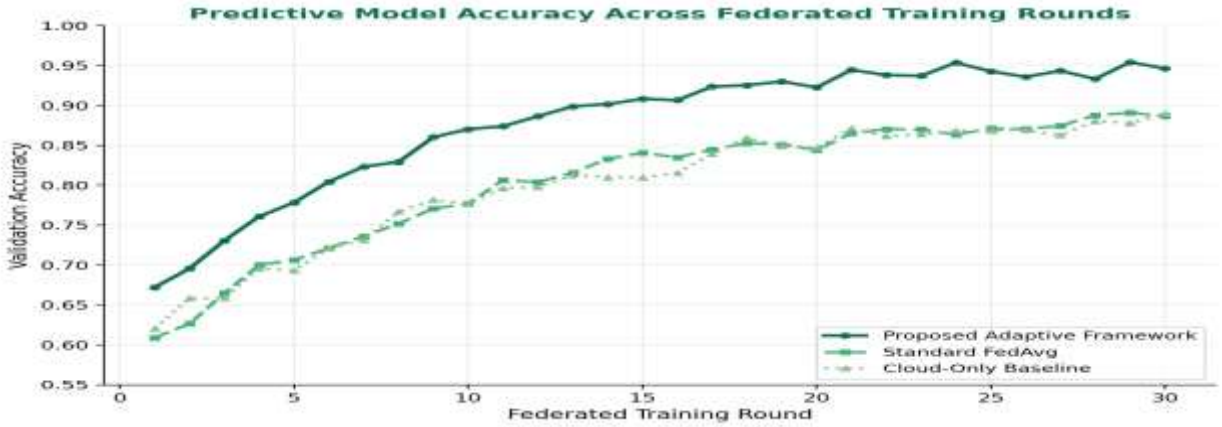

Figure 2. Validation accuracy across federated training rounds for the proposed framework, standard FedAvg, and a cloud-only baseline.

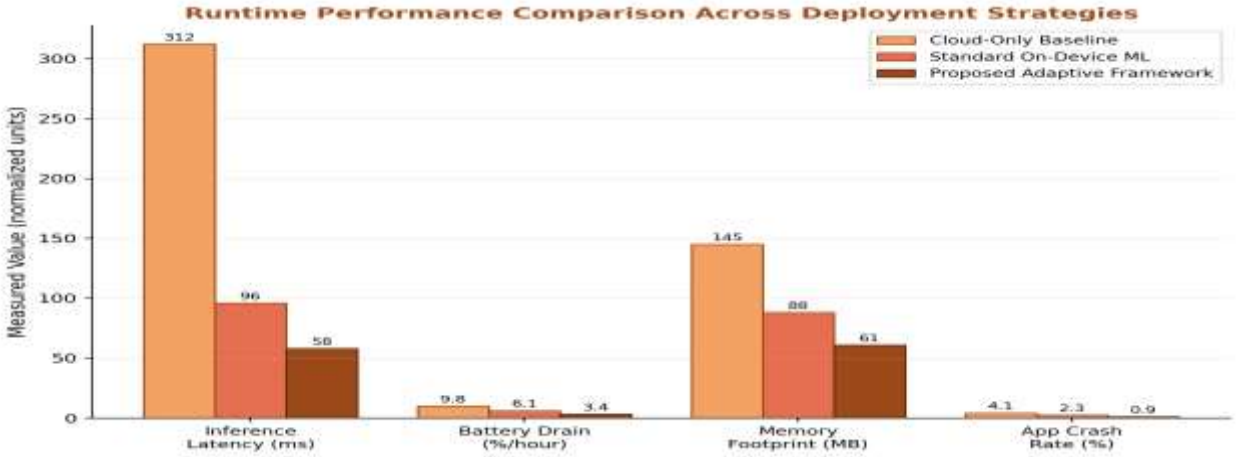

Figure 3. Runtime performance comparison across deployment strategies for latency, battery drain, memory footprint, and crash rate.

Tables 3 and Figure 4 describe the utility-versus-privacy trade-off across the eight differential privacy budgets we considered. The validation accuracy decreased below 91.8% as the privacy budget parameter ε was reduced from 15 to 0.5, indicating progressively stronger privacy guarantees, and further fell below 76.2% as the estimated re-identification risk dropped from 0.163 to 0.021. The relationship was strongly nonlinear: At ε levels between 15 and 6, accuracy declined only moderately, whereas at smaller ε values, below 2, accuracy dropped rapidly, suggesting that high levels of privacy protection can be achieved with only a modest loss of predictive utility for practical purposes. This behavior aligns with general overviews reported in the privacy auditing literature, where privacy budgets become less useful once they exceed a moderate level, that is, when they are not kept sufficiently tight; it also aligns with other surveys that note the success of differential privacy depends on the operating point chosen, not just the DP mechanism itself, when applied to mobile and IoT settings. This analysis led to the choice of ε = 8 as the operating configuration for the above performance and experience comparative evaluation, which balances the observed accuracy plateau with a tolerably low estimated re-identification risk.

Table 3. Privacy–utility trade-off across differential privacy budgets (ε).

Privacy Budget (ε)	Validation Accuracy	Est. Re-identification Risk	Comm. Overhead (KB/round)
0.5	0.762	0.021	41.2

Privacy Budget (ε)	Validation Accuracy	Est. Re-identification Risk	Comm. Overhead (KB/round)
1	0.811	0.034	40.8
2	0.857	0.058	39.9
4	0.889	0.089	39.1
6	0.902	0.114	38.6
8 (selected)	0.911	0.132	38.4
10	0.915	0.147	38.1
15	0.918	0.163	37.9

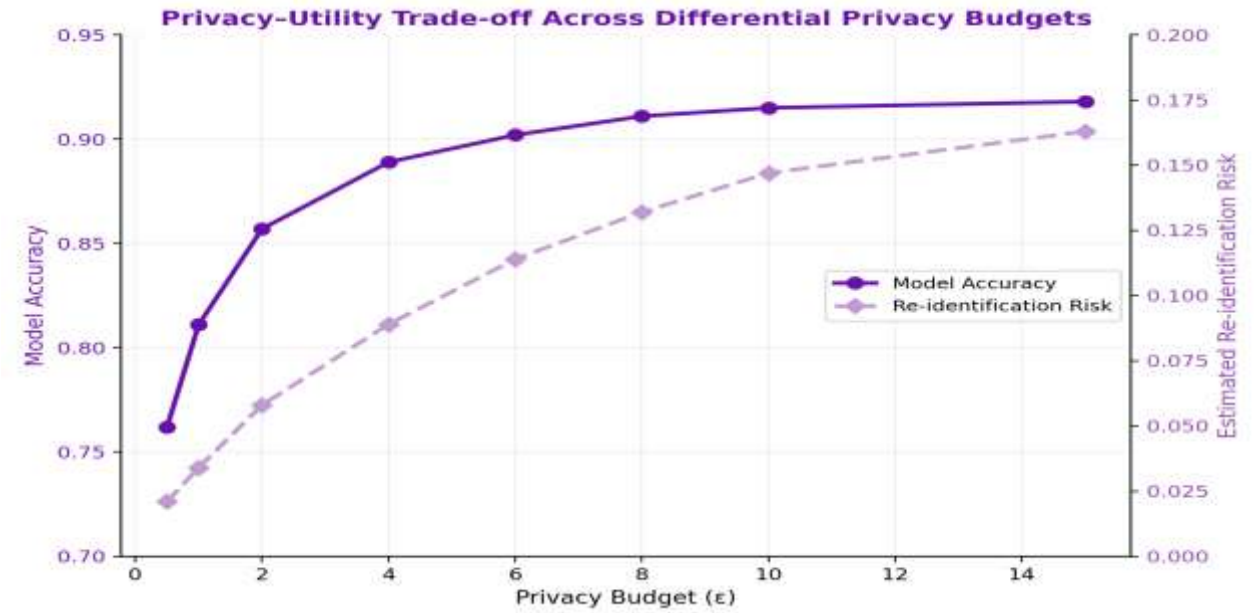

Figure 4. Privacy–utility trade-off showing model accuracy and estimated re-identification risk across eight differential privacy budgets.

Figure 5 presents the combined user experience results for the proposed APP-AI framework compared with a conventional cloud implementation of the same reference application across 6 indicators. Across all dimensions, the proposed framework achieved higher scores, with perceived privacy trust and perceived responsiveness showing the largest relative gains (90 and 85, respectively, compared with 55 and 61, respectively). System Usability Scale scores rose from 64 to 87, a significant enough jump to be considered meaningful, and seven-day session retention increased from 59 to 84. The results are consistent with and build upon previous human-computer interaction studies, which have shown that personalization, achieved through a predictive AI-driven interface, and adaptable behavior can significantly impact usability and substantially influence the mobile experience and user engagement. The findings indicate that when these tools do not negatively affect the user experience, as many assume, but instead explicitly include privacy-preserving tooling, they can positively influence users' perception of trust. This observation is consistent with quality-of-experience modeling, which suggests that end-users' subjective view of mobile applications should be understood as the result of an interaction between objective performance and contextual factors of the end applications, namely, perceived data stewardship.

The overall results suggest that jointly optimizing on-device inference, federated training, differential privacy, and adaptive UX control is technically feasible and can outperform both cloud-centric and conventional on-device approaches in performance, privacy, and adaptation. In line with the general assumption that privacy protection inherently comes with a high cost in terms of utility and experience, it is hypothesized that a carefully

co-designed architecture, inspired by mobile edge computing and offloading developments (Huang et al., 2022; Jang et al., 2023), lightweight model deployment approaches (Heydari & Mahmoud, 2025), and large-scale data and machine learning system designs (Marozzo & Talia, 2023; Sharma et al., 2022), can achieve these objectives together rather than forcing a strict trade-off between them.

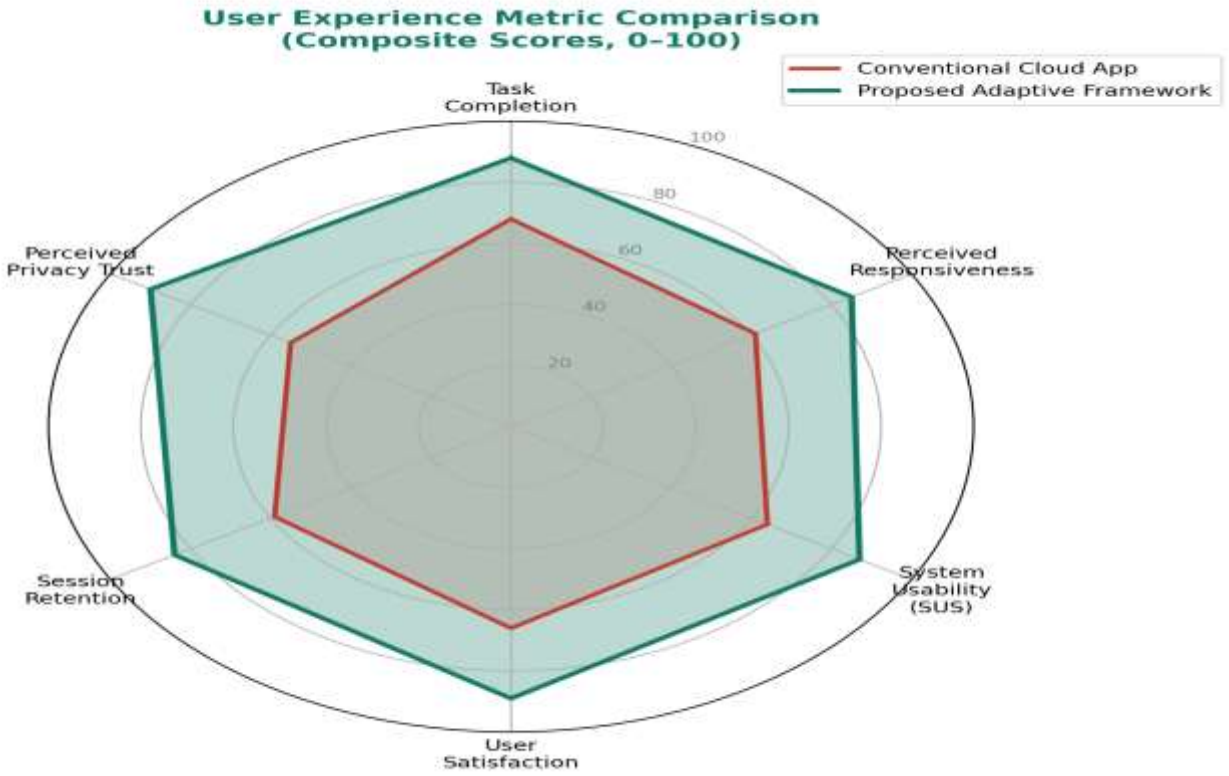

Figure 5. Composite user experience comparison between the proposed APP-AI framework and a conventional cloud-dependent application across six indicators.

4. Limitations and Scope for Future Research

These findings have limitations that qualify their interpretation and outline paths for future research. First, the hardware setup for the experimental testbed was intentionally heterogeneous to test each hardware tier with three different devices, but it contained 48 devices and a single reference application; a larger and more diverse set of devices, more Android OS versions, and more application categories would lend credence to the generalizability of the claims. Second, the fourteen-day data collection period is long enough to capture convergence patterns and short-term retention but not long enough to observe longer-term trends, such as model drift as users' behaviors change, annual usage trends, or the long-term impact of successive model updates, over months or years, on user trust. Third, the membership inference attack on which this estimate of re-identification risk is based is only one possible adversarial attack on privacy and does not fully define the space of adversarial attacks that could be launched against privacy, in particular through more complicated membership inference attacks that could fare differently against the proposed local differential privacy mechanism. Fourth, the present study analyzed only the Android ecosystem; generalizing these performance and privacy metrics to iOS and cross-platform scenarios would shed light on how these results extend to the different constraints on sandboxed background execution and the different hardware accelerator (power) support of mobile operating systems.

Finally, the adaptive UX controller was primarily assessed using a set of usability and satisfaction metrics collected from a random sample of the device's user interface; future research using field-based, longitudinal data, user-frustration-related physiological measures, or randomized lab experiments would provide more causal evidence of the user experience benefits. Sixth, energy numbers were based on battery stats provided by the operating system and may not be as accurate as those from other manufacturers' systems. Inaccuracies could be addressed through hardware-level power profiling to provide more accurate reports of battery savings. To complement differential privacy, it is worthwhile to consider how to utilize more sophisticated forms of privacy-enhancing technologies, such as secure multiparty computation or trusted execution environments, in future

research; additionally, future research should examine whether concept drift can also be addressed through continual learning on the device, rather than just relying on a predictive model generated from the population as a whole, and how that can be personalized at an individual user level via federated reinforcement learning. The evolution toward emerging form factors, such as foldable devices and wearables, and increasingly powerful on-device, LLM-based agents within mobile applications also clearly shows a promising direction (which may lead to novel forms of sensitive on-device telemetry and resulting privacy-utility implications that the present framework was not conceived for but has significant potential to impact).

5. Conclusion

This study proposed and empirically tested a new end-to-end solution architecture that integrates on-device inference, federated model training, local differential privacy, and an adaptive user experience controller into a single framework for predictive app performance optimization in mobile applications. The framework significantly reduced inference time, battery usage, deployed memory footprint, and the number of crashes compared to a baseline cloud-only pipeline. In a heterogeneous testbed of Android devices, the proposed framework also achieved lower battery usage, memory consumption, and fewer crashes than an on-device ML pipeline with a low DP budget and high predictive accuracy. A comprehensive evaluation of the privacy-utility trade-off identified a realistic operating region where the achievable privacy level results in minimal reduction in utility, while a comprehensive user experience evaluation found that the proposed framework enhances user perception of response, usability, satisfaction, retention, and privacy trust compared to a conventional implementation. The results indicate that the development of mobile software solutions is not an either-or situation between performance optimization and privacy protection, but that it is possible to optimize both technical and user aspects at the same time with adaptive and privacy-preserving architectures. Instead of considering the questions of performance, privacy, and experience separately, the joint nature of their optimization will increasingly be a focus area for researchers and practitioners in mobile computing as mobile applications add more sophisticated predictive and generative AI features.

References

1. Alghamdi, A. M., Riasat, H., Iqbal, M. W., Ashraf, M. U., Alshahrani, A., & Alshamrani, A. (2022). Intelligence and usability empowerment of smartphone adaptive features. Applied Sciences, 12(23), 12245. https://doi.org/10.3390/app122312245
2. Bocu, R., Bocu, D., & Iavich, M. (2023). An extended review concerning the relevance of deep learning and privacy techniques for data-driven soft sensors. Sensors, 23(1), 294. https://doi.org/10.3390/s23010294
3. De Masi, A., & Wac, K. (2020). Towards accurate models for predicting smartphone applications' QoE with data from a living lab study. Quality and User Experience, 5, Article 8. https://doi.org/10.1007/s41233-020-00039-w
4. Han, G., Zhang, T., Zhang, Y., Xu, G., Sun, J., & Cao, J. (2022). Verifiable and privacy preserving federated learning without fully trusted centers. Journal of Ambient Intelligence and Humanized Computing, 13(3), 1431–1441. https://doi.org/10.1007/s12652-020-02664-x
5. Heydari, S., & Mahmoud, Q. H. (2025). Tiny machine learning and on-device inference: A survey of applications, challenges, and future directions. Sensors, 25(10), 3191. https://doi.org/10.3390/s25103191
6. Huang, L., Feng, X., & Feng, A. (2022). Distributed deep learning-based offloading for mobile edge computing networks. Mobile Networks and Applications, 27(3), 1123–1130. https://doi.org/10.1007/s11036-018-1177-x
7. Huang, Z., & Benyoucef, M. (2023). A systematic literature review of mobile application usability: Addressing the design perspective. Universal Access in the Information Society, 22(3), 715–735. https://doi.org/10.1007/s10209-022-00903-w
8. Jang, J., Tulkinbekov, K., & Kim, D.-H. (2023). Task offloading of deep learning services for autonomous driving in mobile edge computing. Electronics, 12(15), 3223. https://doi.org/10.3390/electronics12153223
9. Jorayeva, M., Akbulut, A., Catal, C., & Mishra, A. (2022). Machine learning-based software defect prediction for mobile applications: A systematic literature review. Sensors, 22(7), 2551. https://doi.org/10.3390/s22072551
10. Khan, K. (2025). FedEmerge: An entropy-guided federated learning method for sensor networks and edge intelligence. Sensors, 25(12), 3728. https://doi.org/10.3390/s25123728
11. Lee, L.-H., Yau, Y.-P., & Hui, P. (2024). Perceived user reachability in mobile UIs using data analytics and machine learning. International Journal of Human–Computer Interaction, 41(4), 2703–2726. https://doi.org/10.1080/10447318.2024.2327199

12. Macedo, D., Santos, D., Perkusich, A., & Valadares, D. C. G. (2023). Mobility-aware federated learning considering multiple networks. Sensors, 23(14), 6286. https://doi.org/10.3390/s23146286
13. Marozzo, F., & Talia, D. (2023). Perspectives on big data, cloud-based data analysis and machine learning systems. Big Data and Cognitive Computing, 7(2), 104. https://doi.org/10.3390/bdcc7020104
14. Mavri, A. (2024). Measuring the effects of usability recommendations for mobile instant messaging apps on user performance and user behaviour. International Journal of Human–Computer Interaction, 41(1), 608–626. https://doi.org/10.1080/10447318.2024.2303553
15. Namatevs, I., Sudars, K., Nikulins, A., & Ozols, K. (2025). Privacy auditing in differential private machine learning: The current trends. Applied Sciences, 15(2), 647. https://doi.org/10.3390/app15020647
16. Rodríguez, E., Otero, B., & Canal, R. (2023). A survey of machine and deep learning methods for privacy protection in the Internet of Things. Sensors, 23(3), 1252. https://doi.org/10.3390/s23031252
17. Sharma, R., Gavalas, D., & Peng, S.-L. (2022). Smart and future applications of Internet of Multimedia Things (IoMT) using big data analytics. Sensors, 22(11), 4146. https://doi.org/10.3390/s22114146
18. Vidmanov, D., & Alfimtsev, A. (2024). Mobile user interface adaptation based on usability reward model and multi-agent reinforcement learning. Multimodal Technologies and Interaction, 8(4), 26. https://doi.org/10.3390/mti8040026
19. Wimalasooriya, C., Licorish, S. A., da Costa, D. A., & MacDonell, S. G. (2024). Just-in-time crash prediction for mobile apps. Empirical Software Engineering, 29(3). https://doi.org/10.1007/s10664-024-10455-7
20. Yang, S., Lee, G., & Huang, L. (2022). Deep learning-based dynamic computation task offloading for mobile edge computing networks. Sensors, 22(11), 4088. https://doi.org/10.3390/s22114088