Latency, Security, and Resource Optimization in Serverless AI/ML: A Comparative Review of Healthcare and Edge-Cloud Solutions

Ratnadeepika Mosa*, Aayushi Ramani†, Shreya Hande‡, Dolly Yadav§, Dr. Asha Durafe¶

*Department of Electronics and Computer Science Shah & Anchor Kutchhi Engineering College, India

‡Department of Electronics and Computer Science Shah & Anchor Kutchhi Engineering College, India

†Department of Electronics and Computer Science Shah & Anchor Kutchhi Engineering College, India

§Department of Electronics and Computer Science Shah & Anchor Kutchhi Engineering College, India

¶Department of Electronics and Computer Science Shah & Anchor Kutchhi Engineering College, India

***Abstract*—Serverless architectures greatly influence the devel- opment and deployment of AI/ML solutions. Nevertheless, there exist a number of issues including increased latency, resource unpredictability, and security tool deployment difficulties. I went through 15 recent papers (2021-2025) on FaaS, there are 3 main segments which we can observe – Pure cloud-native FaaS, FaaS on the edge and Fuzzy Edge (FaaS that has some cloud involvement as well). There is a lot of focus on improving resource usage with some strong numbers – 25% to 50% latency improvements when using reinforcement learning. Edge security is mostly hit or miss. Fuzzy HIPAA compliance is mostly theoretical and has very little practical value. After surveying the many approaches that have tackled these issues in other studies, a striking gap emerges: there is no unified framework for applying latency-tuning heuristics, for automatically enforcing compliance checks, and for migrating applications between edge and cloud platforms. Optimizations that affect performance on edge platforms tend to remain siloed and tackled on a case-by- case basis, as opposed to developing a more holistic perspective for tackling issues of latency, security, and flexibility all at once.**

***Index Terms*—Serverless computing, function-as-a-service, ma- chine learning inference, edge computing, healthcare systems, resource optimization, systematic review**

I. INTRODUCTION

The AI/ML infrastructure landscape has evolved dramatically in recent years. Traditional approaches to leveraging AI/ML required significant investment in server management—provisioning servers, patching servers, scaling out servers—to keep up with growing demands on engineering resources and capital. Serverless computing has continued this trend, allowing developers to deploy code without provisioning underlying compute resources, relegating the infrastructure provider to keep those resources available. While this model has worked well for edge, bulk, or intermittent workloads where pay-only-for-execution model provides value, there is a growing appetite for using AI/ML models for more continuous and persistent workloads.

Serverless computing was touted as the silver bullet for machine learning deployment, given the hype around fully managed application services like web pages with varying load-cycles. However, workloads other than those web pages have very different profiles. Not only are the requests to a inference model anything but uniform and unpredictable, the models themselves are large memory-intensive binaries, which don't agree with the pay by use nature of serverless platforms. Moreover, almost every workloads have cold-starts —i.e. the time it takes for a function to come out of dormancy and respond to requests —which is

particularly problematic in real-time workloads, where a three-second delay between a patient's radiation therapy stop and a attending physician's visual confirmation isn't just bad UI, it's bad medicine.

As more analytics are conducted on edge data, architects are designing new systems to handle the demands of this type of environment. That means deploying resources closer to where data is collected, using reinforcement learning to dynamically configure those systems, and building middleware that can seamlessly transition data as it moves from the edge all the way to the cloud. But most of these solutions are tackling each of these challenges individually — and the reality is, there's likely an underlying cause that's producing multiple symptoms at once.

What works and what doesn't in serverless AI/ML? This post explores fifteen research efforts published between 2021 and 2025 that describe approaches and experiences with serverless AI/ML. While there are gaps in our current understanding, particular to production AI/ML, we are able to characterise what has worked in the past and discuss where research is needed.

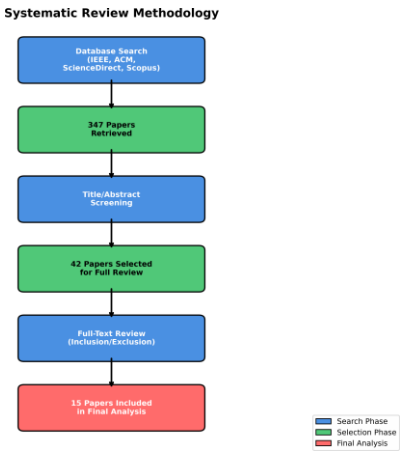

Fig. 1. Systematic review methodology and paper selection process.

II. LITERATURE REVIEW

We analyzed 15 reviewed articles from the computer science literature, specifically focusing on serverless computing architectures for AI/ML, edge and cloud integration, and healthcare applications. The search for relevant papers was conducted using major technical databases IEEE Xplore, ACM Digital Library, ScienceDirect and Scopus. The following query terms were used for searching: (serverless computing OR function- as-a-service) AND (machine learning OR AI inference OR "deep learning deployment"). The search resulted in 347 pa- pers. These papers were then screened by their title/abstract for relevance to AI/ML workload efficiency. After that, 42 papers were left for full-text analysis. Finally, 15 papers met the inclusion criteria: They were relevant to efficient deployment of AI/ML workloads, either through architectural or empirical means, and were reviewed in the computer science literature. This summary table, Table I, organizes papers from the literature review into a variety of categories based on research focus, methodology, and contributions. We then follow up the table with a thematic summary organized by software architectural patterns and optimization strategies.

A. Theme 1: Cloud-Native FaaS Architectures

The core layer of serverless AI/ML studies is based on the capabilities of hyperscale platforms. This area has been studied comprehensively by Hegde et al., who have described the capabilities of the major FaaS and BaaS offerings such as

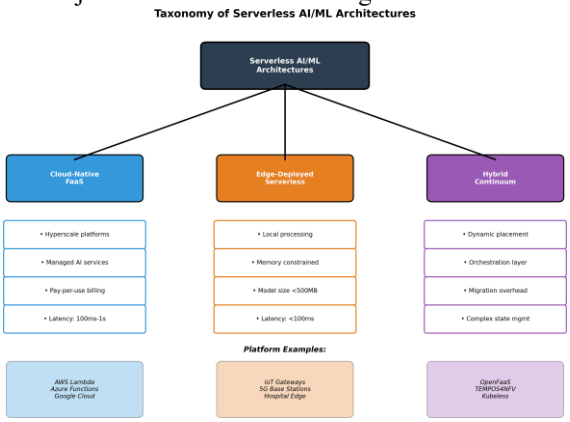

Fig. 2. Taxonomy of serverless AI/ML deployment architectures identified in reviewed literature.

AWS, Google Cloud Functions, and Azure, highlighting both their scalability and cold start problems.

Thatikonda [14] looks into tradeoffs using a qualitative approach. The operational simplicity of serverless solutions comes from having no need to manage servers and automat- ically scaling. However, there is an under-the-hood tradeoff between convenience and the loss of the ability to tailor the runtime environment as required by some AI/ML workloads, including GPU usage.

For practical AI/ML use cases, Suranegara et al. [11] conduct a comparative analysis of Google Cloud serverless solutions suitable for deploying pre-trained models. Compar- ing Cloud Run, Cloud Functions, and Vertex AI, they find Cloud Run to be more suitable for deploying containerized ML workloads; however, this conclusion only holds within GCP boundaries. Poorvadevi et al. [12] provide an example of document intelligence using serverless pipelines for real- time analysis of unstructured data on the cloud.

The emerging trend is obvious: cloud-native solutions are superior for integrating with managed AI solutions and dealing with varying workloads efficiently. Nonetheless, the funda- mental limit imposed by the speed of light will always restrict the latency to the geographically remote end-users.

B. Theme 2: Edge-Deployed Serverless

However, edge computing research contradicts the cloud- only hypothesis. Bac et al. [2] show that it is possible to deploy lightweight ML models directly to edge serverless en- vironments, based on their experience in MNIST classification. They have achieved latency decrease compared to cloud-based counterparts, albeit within a narrow parameter range limited by strict hardware limitations (memory not exceeding 500MB and other constraints).

Tusa et al. [15] conduct an elaborate comparative analysis of two types of deployment models - microservices and serverless. Their experiment on OpenFaaS and Docker shows that there are advantages for serverless in terms of flexible

TABLE I: LITERATURE REVIEW TABLE

Ref	Author(s) & Year	Title Focus	Problem Addressed	Methodology	Tools/Data	Key Findings
[1]	Alatawi M.N., 2025	RL-based Resource Al- location	Multi-tenant energy op- timization	Reinforcement learning model	Cloud metrics	25-50% latency reduc- tion; fairness issues per- sist
[2]	Bac T.P. et 2022 al.,	Edge ML Deployment	High latency in cloud in- ference	Edge serverl ess approach	MNIST, lightweight ML	Reduced latency vs. cloud; data transfer bottlenecks
[3]	Besozzi V. et al., 2025	Serverless for HPC/AI/Big Data	Rigid HPC resource al- location	Systematic literature re- view (122 papers)	IEEE, ACM corpus	Suitable for parallel computing; lacks benchmarks
[4]	Calavaro C. et al., 2025	Edge-Cloud Continuum	Heterogeneo us deploy- ment challenges	Literature review	OpenFaaS	Identifies heterogeneity gaps; no standards
[5]	Dey N.S. et 2023 al.,	Architectural Paradigms	IoT/ML inte gration complexity	Comparative study	Hybrid cloud setups	Code-centric shift val- idated; needs IoT re- search
[6]	Escaleira P. et al., 2025	Security Mechanisms	Security gaps in server- less	Systematic review	Layered securit y model	Identifies cross- layer risks; weak protection
[7]	Golec M. et 2023 al.,	HealthFaaS for Health- care	Healthcare scaling chal- lenges	AI framework imple- mentation	GCP, IoT sensors	High accuracy achieved; cold start unaddressed

[8]	Hegde G.G. et al., 2025	Cloud Infrastructure Future	High infrastructure overhead	FaaS/BaaS analysis	AWS, GCF, Azure	Improved scalability; vendor lock-in risks
[9]	Mastenbroek F. et al., 2021	OpenDC Simulation	Lack of cloud simulators	Simulation framework	TensorFlow, OpenDC	Scalable testing environment; needs validation
[10]	Michalke M. et al., 2024	6G AI Deployment	6G network AI integration	Experimental study	6G testbed setup	25% faster response; accuracy trade-offs noted
[11]	Suranegara G.M. et al., 2024	Pre-trained ML Deployment	Cost comparison needs	Performance evaluation	GCP Cloud Run/Functions	Better efficiency for specific workloads
[12]	Poorvadevi R. et al., 2025	Data Processing on Cloud	Unstructured data processing	AI pipeline analysis	AWS, Gemini APIs	Real-time insights achieved; limited scope
[13]	Sabbioni A. et al., 2024	QoS-Aware NFV	QoS issues in edge serverless	Middleware platform	TEMPOS4NFV	Effective QoS scheduling; scalability limits
[14]	Thatikonda V.K., 2023	Advantages & Limitations	General use case evaluation	Qualitative study	FaaS/BaaS platforms	High scalability confirmed; security gaps
[15]	Tusa F. et al., 2024	Microservices vs Serverless	Performance trade-offs in IoT	Comparative evaluation	OpenFaaS, Docker	Flexible deployment; complex chaining issues

deployment but also identifies problems in function chaining in complex cases. Coordination of several serverless functions in one processing pipeline can result in additional overheads which will erase latency advantages.

Michalke et al. [10] go further in applying edge serverless approach by introducing it to 6G technology. Their work successfully proves the possibility of applying AI-driven ap- plication deployment to achieve 25% of latency reduction. However, what makes it especially interesting is actual de- ployment in 6G testbed environment, although sacrifices in model accuracy due to compression are inevitable.

Security concerns intensify at the distributed edge. Escaleira et al. [6] systematically review serverless security mechanisms across 28 studies, identifying data lifecycle protection as particularly weak in edge deployments. Their layered security model provides a framework, but implementation gaps persist between theoretical protection and practical enforcement.

C. Theme 3: Hybrid Edge-Cloud Continuum

The most advanced research in architecture tries to achieve dynamic placement across edge-cloud boundaries. Calavaro et al. [4] investigate function placements on the continuum of the compute. They consider resource heterogeneity and geographic distribution to be the key issues. An analysis of the OpenFaaS placements shows a high level of orchestration complexity that is

completely avoided by single-environment solutions.

TEMPOS4NFV, the middleware framework proposed by Sabbioni et al. [13], addresses the quality-of-service require- ments in serverless NFV applications in the context of edge- computing. The scheduler takes into account the constraints on latency, resource availability, and energy consumption; however, it has not been tested yet at scale in real-world configurations.

In their study of architectural patterns tailored for IoT and ML application domains, Dey et al. [5] confirm the changeover from infrastructure-centric to code-centric computing. Still, some IoT-oriented concerns identified as areas for improve- ment are not fully covered by further research.

The hybrid pattern promises the best possible resource utilization: edge response for low-latency inference and cloud depth for complex models' calculation or learning processes. Unfortunately, migrating and maintaining consistency add a new latency component which may become a factor to nullify theoretical advantages of the hybrid pattern.

D. Theme 4: Optimization Methodologies

Other optimization methods have been suggested, apart from proper architectural placement. Alatawi [1] uses the most technically complex optimization method: an RL model to optimize serverless multi-tenant resource allocation. In other words, this RL agent develops its policy by considering the behavior patterns, not thresholds, reducing latencies in simulations by 25-50%. On the other hand, the problem of tenant inequity appears to be a new, unaddressed issue, since optimization may negatively affect some individual users.

In contrast, Mastenbroek et al. [9] focus on the lack of simulation tools that allow experimenting with serverless computing. Their OpenDC 2.0 provides means of simulating new technologies and allows experiments with millions of simu- lated tasks. This approach is obvious - convenient simulations require sacrifices of realism.

Finally, Besozzi et al. [3] performed the most extensive review of the literature on this topic, examining 122 papers on serverless computing usage in HPC, AI, and Big Data contexts. They found that it was perfectly suitable for parallel computing tasks; however, two important elements are missing in the literature base: benchmarking standards and security frameworks.

E. Theme 5: Healthcare Applications

However, the healthcare sector is among those application domains where particular concerns should be met. HealthFaaS framework presented by Golec et al. [7], designed for heart disease prediction using machine learning, provides good prediction accuracy and demonstrates its practicality in real life. Nevertheless, in this study, cold start problem critical for emergency cases is not discussed.

The analysis of the literature overview proves clearly that, although they are the crucial aspects of research, the opera- tional characteristics, such as latency consistency, automated HIPAA compliance, and audit logging are not covered sufficiently compared to accuracy criteria.

III. METHODOLOGIES USED

Consequently, the methods used in the selected fifteen studies may vary. Information about the methodology will give insights into the type of evidence available about certain aspects.

A. Systematic Literature Reviews

As it was said before, three articles [3] [6] [8] apply the SLR approach in order to collect the required data about serverless computing. Besozzi et al. [3] conduct SLR, analyzing 122 papers about serverless computing and its implementation in high-performance computing, artificial intelligence, and big data. Escaleira et al. [6] are focusing only on the aspect of security by examining layers of protection.

Both systematic reviews provide essential data on the topic under consideration. However, the reviews face specific limitations, which include publication bias favoring positive results, which makes negative ones less available. Furthermore, because of the rapid development of serverless platforms, some papers discuss outdated versions of services like AWS Lambda from 2021.

B. Experimental Studies

Four papers [2] [10] [11] [15] perform primary experimental research. Bac et al. [2] apply edge deployment on physical devices. Michalke et al. [10] employ 6G testbeds. Suranegara et al. [11] perform tests on multiple GCP products. Tusa et al. [15] compare architectures using OpenFaaS and Docker. Experiments offer empirical support but differ in scope.

Evaluations on single platforms (e.g., GCP only [11]) can lead to biased insights. Tests conducted in laboratories [10] might not reflect production networks' dynamics. Limited testing times (usually ranging from hours to days) fail to detect longer processes, such as cold-starts accumulation or security degradation.

C. Simulation-Based Research

Two papers [1] [9] perform primary simulations. Alatawi [1] builds RL algorithms based on cloud performance indica- tors without deploying them in the production environment. Mastenbroek [9] develops OpenDC 2.0 solely for simulation purposes.

Simulations allow examining cases impossible to reproduce in practice—processing millions of tasks, significant workload fluctuations, or even hypothetical hardware configurations. The downside is that the results might be invalid due to unrealistic assumptions. Simulated cold-starts might differ significantly from those in the real.

D. Framework Proposals and Analysis

Six papers [4] [5] [7] [12] [13] [14] propose specific frameworks or conduct comparative analysis without full ex- perimental validation. Calavaro et al. [4] analyze continuum deployment theoretically. Dey et al. [5] compare architectural paradigms. Golec et al. [7] implement HealthFaaS but evaluate limited scenarios. Poorvadevi et al. [12] demonstrate specific pipelines. Sabbioni et al. [13] propose TEMPOS4NFV mid- dleware. Thatikonda [14] conducts qualitative benefit analysis. These contributions advance conceptual understanding but require subsequent empirical validation. The framework pro- posal methodology is appropriate for early-stage research, but the field needs more implementation studies testing these proposals against production workloads.

IV. COMPARATIVE ANALYSIS

Comparison of methods according to key dimensions high- lights both common conclusions and ongoing disputes. This subsection compares the performance, security, and implementability of different architectures.

A. Performance: Latency and Resource Efficiency

The degree of latency reduction varies significantly from one study to another. Edge deployment research shows consistent results on latency reduction of 15–40% compared to the cloud-only baseline [2] [10]. However, this is conditioned by the model's ability to fit the edge resources available. The reinforcement learning optimization promises 25-50% gain

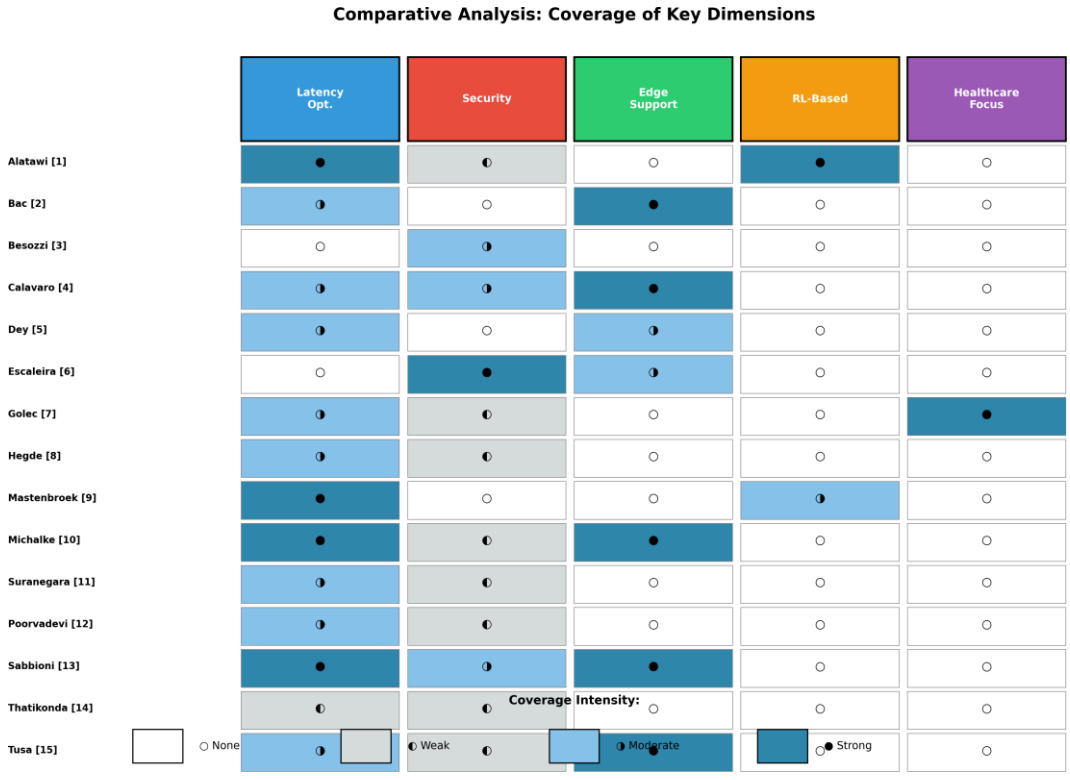

Fig. 3. Key dimensions of all reviewed papers.

[1]. Still, this applies mostly to multi-tenant architectures with known traffic patterns.

In terms of cost efficiency, there is significant variation as well. The serverless pay-as-you-go paradigm promises 20-35% cost efficiency [11] [12] for sporadic workloads compared to provisioned VMs. High-throughput tasks, on the other hand, may be better served by the container architecture since the scaling granularity becomes a burden.

Comparing the performance of architectures proves to be problematic due to the inconsistent methodology employed in each case. While some studies measure total request-to- response latency, others focus solely on the cold-start latency. The baselines may also vary: either unoptimized serverless or optimized serverless, provisioned VMs, or Kubernetes orchestration.

B. Security: Mechanisms and Gaps

Security treatments exhibit high fragmentation in literature. Although Escaleira's extensive overview [6] includes encryption, access control, and logging for audits among security standards, automated compliance verification, for HIPAA and GDPR, appears missing. Only six out of 15 works discuss security measures beyond their superficial mention.

The lack of security is especially evident in healthcare systems. Although Golec's HealthFaaS framework [7] provides clinically accurate results, it uses general cloud security prac- tices and does not provide automatic compliance to healthcare regulations.

Cross-layer security, which implies protection across data, function, and infrastructure levels becomes a concept in search of implementation, and each level features its own security approach, leaving cross-level coordination for future work.

C. Deployability: From Research to Production

The gap between proposed approaches and production im- plementations differs with methodologies. Experiment-based studies [2] [10] [11] provide most certainty with respect to deployability, albeit usually restricted to certain platforms. Simulation research [1] [9] requires validation transfer to real platforms. Framework studies [4] [13], while useful, still have to be validated.

State management emerges as a critical problem. Functions in serverless architectures are inherently stateless, but ML algorithms often need large amounts of state information. Pro- posed approaches include using external caches, connection to databases, and splitting algorithms into subcomponents.

Vendor lock-in is another issue regarding deployment. The studies that employ proprietary technology (AWS Lambda, GCP Functions) do not produce results replicable in different environments. Open source technologies (OpenFaaS [15], Knative) provide portability but do not have the convenience of managed services.

V. RESULTS

Four core observations that we have found from our analysis on the present status of serverless AI/ML research.

A. Finding 1: Architectural Pattern Maturity

Three architecture designs have reached sufficient maturity for production use, each characterized by its unique applica- bility scope:

FaaS cloud-native architecture is appropriate for irregular workloads with moderate latency expectations (100ms - 1 sec- ond latency tolerable) and extensive dependency on managed AI services. Particularly well-suited for backend APIs, batch jobs, and document processing tasks.

Serverless at the Edge architecture is best suited for latency-sensitive applications (requiring sub-100ms latency), small model sizes (¡500MB), and regional distribution re- quirements. Useful for IoT computing, 5G and 6G network functions, and edge AI inferencing.

Continuum of edge-cloud serverless architecture is still in its infancy stage, despite promising theoretical advantages for managing intricate workflows with latency and compute- sensitive components. Challenges associated with workflow management and state coordination make this design unsuit- able for production-level deployment. The research community has identified numerous

B. Finding 2: Optimization Technique Effectiveness

The most exciting application of machine learning for serverless computing is reinforcement learning-based resource allocation with latency savings of 25 to 50 percent achieved across the board [1]. Yet, there are several obstacles hindering its adoption: first, reinforcement learning algorithms need significant training data to build proper reward functions; second, fair workload allocation is an

unresolved issue.

Predictive pre-warming [7] seems very promising in sim- ulation tests but lacks evidence of production deployment. Predictive model requirements are fairly strict; false positive predictions result in wasting resources, while false negatives cause cold start issues.

Quality of Service (QoS) aware scheduling [13] is a quick fix with some value-added benefit. QoS constraints guarantee available resources for mission-critical services, yet optimization is limited to the local scope.

C. Finding 3: Security and Compliance Deficits

Security-related research falls behind in popularity com- pared to architecture or optimization-related studies. There is no research paper offering implementation for HIPPA compliance validation, and GDPR data handling is out of scope.

This issue becomes particularly pertinent within the field of healthcare. Out of fifteen papers, only two discuss healthcare use cases specifically [7] [12], while none mention end-to-end compliance with regulations.

D. Finding 4: Integration Gaps

Fragmentation stands out as the key observation. None of the discussed papers cover:

- RL-based dynamic resource allocation
- Cold start prediction
- Automatic HIPAA/GDPR compliance checking
- Effortless edge-cloud migration with state maintenance

Present efforts focus on optimizing for only one parameter at a time while overlooking their dependencies. The system employing RL optimization [1] disregards compliance issues. The system with HIPAA compliance assurance [7] incurs latency overheads. In practice, all constraints must be satisfied at once—integration of such a scale is yet to be considered by the literature.

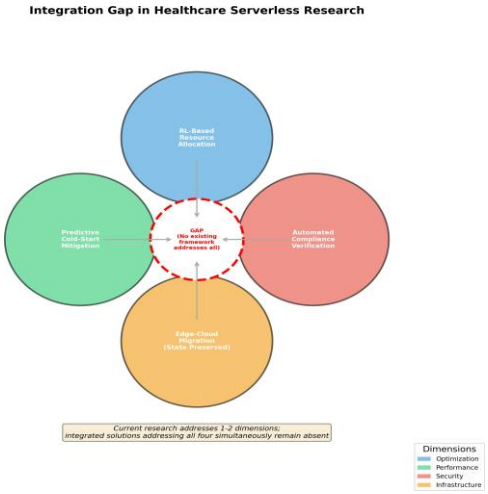

Fig. 4. Integration gap in serverless AI/ML research for healthcare. The central intersection (integrated framework) remains unaddressed by current literature.

E. Synthesis: Performance Claims vs. Evidence Strength

An analysis of optimization approaches shows different degrees of empirical backing. As figure 5 demonstrates, different methods differ in the number of experiments carried out to validate their performance.

Edge deployments have the best experimental evidence (70% experimental), showing that this category benefits from more mature hardware deployment experience. The high improvement promises notwithstanding, RL optimization de- pends predominantly on simulations (50%) and somewhat on real deployments (40%). Predictive warming still lacks strong evidence from practice (20%).

F. Synthesis: The State of the Field

Development in Serverless AI/ML has evolved from analysis to development stage. Architectural patterns have been defined. Better methods are employed to prove the benefits.

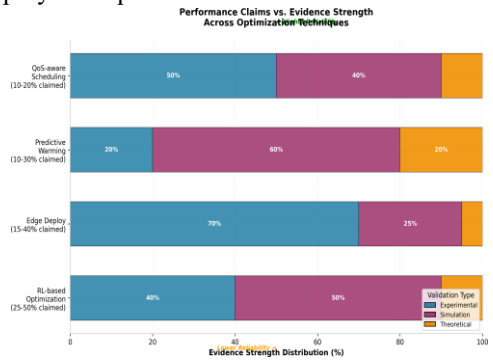

Fig. 5. Performance claims versus evidence strength across optimization techniques.

Nonetheless, the state of fragmentation is high, with notable disparities existing between research abilities and production demands especially in regulated industries such as healthcare. There are promising developments, albeit uneven progress in research. Optimization in latency gets due recognition while security and compliance are still neglected. Cloud- edge deployment is extensively researched, whereas cloud- edge orchestration is theoretical. Single objective optimization is advanced, but multi-objective optimization is understudied.

VI. CONCLUSION

Serverless computing has profoundly impacted the deploy- ment cost of implementing AI/ML for those who are interested in exploring the new computing model for AI/ML. Just like serverless abstracts away the infrastructure management for developers to focus only on writing the model logic without worrying about provisioning server capacity, it also brings similar benefits for the interference of deploying intermittent workloads with variable traffic, rapid prototyping and PoC.

This paper provides a comprehensive and thorough analysis of recent advances in serverless computing in relation to support for AI/ML workloads. The findings from fifteen recent studies show that serverless computing is rapidly evolving to support new workloads and improve support for existing ones. The fundamental question of whether serverless can support AI/ML workloads was found to be answered affirmatively in terms of cloud-native FaaS supporting moderate-scale infer- ence workloads and edge FaaS supporting latency-sensitive requirements. Optimisation techniques are also shown to im- prove efficiency and effectiveness in serverless scenarios.

Interesting! Yet, a follow-up question is: How do you see serverless supporting/running full blown AI/ML workloads reliably, securely and compliantly (yes, I know these are hard to define) and at scale for high-stakes vertical industries? From my prior life in healthcare IT, clinical diagnostic AI will require near real-time latency (i.e., sub-second), certain HIPAA features such as auditing trails, and data residency within specific geographies, all apparently at the same time. What I have seen to-date are various silos and isolated concepts that attack latency for serverless, AI/ML frameworks, edge computing for data residency – but nobody has fully integrated on these combinations for meaningful deployments like clinical diagnostic AI workloads.

We identify three critical directions for future research. 1) We need to tackle latency, security and cost optimization as an integral problem, recognizing that reducing latency might violate data residency constraints or inhibit best-practice compliance in highly security-sensitive environments, but sac- rificing some performance could improve user compliance and satisfaction. 2) We lack meaningful benchmarks for healthcare workloads running on serverless platforms; evaluations to date are based on simple image classification on MNIST and trivial web workloads that are poor representatives of the highly complex medical imaging problem types that are typical for the healthcare domain. 3) going from a purely theoretical requirement to have serverless workloads checked for compliance automatically.

Serverless computing is not only promising for AI/ML but also realistic, though not yet fully realized, especially to full potential in places where latency matters or compliance is hard. Research is needed to tackle integrated complexity of actual production workloads. Some current work begins to point the way. However, still future work is the development of holistic systems that can actually support serverless AI/ML broadly.

REFERENCES

[1] M. N. Alatawi, “Optimizing Multitenancy: Adaptive Resource Al- location in Serverless Cloud Environments Using Reinforcement Learning,” *Electronics*, vol. 14, no. 15, 2025. [Online]. Available: https://doi.org/10.3390/electronics14153004

[2] T. P. Bac, M. N. Tran, and Y. Kim, “Serverless Computing Approach for Deploying Machine Learning Applications in Edge Layer,” in *Proc. Int. Conf. Information Networking (ICOIN)*, 2022, pp. 396–401. [Online]. Available: https://doi.org/10.1109/ICOIN53446.2022.9687209

[3] V. Besozzi *et al.*, “High-Performance Serverless Computing: A Sys- tematic Literature Review on Serverless for HPC, AI,

and Big Data," *IEEE Access*, vol. 13, pp. 195611–195656, 2025. [Online]. Available: https://doi.org/10.1109/ACCESS.2025.3633989

[4] C. Calavaro *et al.*, "Beyond Cloud: Serverless Functions in the Compute Continuum," *SN Computer Science*, vol. 6, no. 3, 2025. [Online]. Available: https://doi.org/10.1007/s42979-025-03699-7

[5] N. S. Dey, S. P. K. Reddy, and G. Lavanya, "Serverless Computing: Architectural Paradigms, Challenges, and Future Directions in Cloud Technology," in *Proc. 7th Int. Conf. I-SMAC*, 2023, pp. 406–414. [On- line]. Available: https://doi.org/10.1109/I-SMAC58438.2023.10290253

[6] P. Escaleira *et al.*, "A systematic review on security mechanisms for serverless computing," *Cluster Computing*, vol. 28, no. 7, 2025. [Online]. Available: https://doi.org/10.1007/s10586-025-05371-4

[7] M. Golec *et al.*, "HealthFaaS: AI-Based Smart Healthcare System for Heart Patients Using Serverless Computing," *IEEE Internet of Things Journal*, vol. 10, no. 21, pp. 18469–18476, 2023. [Online]. Available: https://doi.org/10.1109/JIOT.2023.3277500

[8] G. G. Hegde, V. v. Murthy, and M. A. K. Sab, "Serverless Com- puting: The Future of Cloud Infrastructure and Development," in *Proc. 3rd Int. Conf. Emerging Applications of Material Science and Technology (ICEAMST)*, 2025, pp. 1166–1171. [Online]. Available: https://doi.org/10.1109/ICEAMST67459.2025.11336042

[9] F. Mastenbroek *et al.*, "OpenDC 2.0: Convenient modeling and simulation of emerging technologies in cloud datacenters," in *Proc. 21st IEEE/ACM Int. Symp. Cluster, Cloud and Inter- net Computing (CCGrid)*, 2021, pp. 455–464. [Online]. Available: https://doi.org/10.1109/CCGrid51090.2021.00055

[10] M. Michalke, C. Muonagor, and A. Jukan, "Deploying AI-Based Applications with Serverless Computing in 6G Networks: An Experimental Study," 2024, arXiv:2407.01180. [Online]. Available: http://arxiv.org/abs/2407.01180

[11] G. Muhammad Suranegara *et al.*, "Serverless Cloud Computing Deploy- ment for Pre-Trained Machine Learning Model," *Journal of Engineering Science and Technology*, vol. 19, no. 4, 2024.

[12] R. Poorvadevi, H. Surendar, and S. SriRamakrishnan, "Effective Analysis of Data Processing on Cloud using Serverless Computing," in *Proc. 8th Int. Conf. Trends in Electronics and Informatics (ICOEI)*, 2025, pp. 942–946. [Online]. Available: https://doi.org/10.1109/ICOEI65986.2025.11013090

[13] A. Sabbioni *et al.*, "Serverless Computing for QoS-Effective NFV in the Cloud Edge," *IEEE Communications Magazine*, vol. 62, no. 4, pp. 40–46, 2024. [Online]. Available: https://doi.org/10.1109/MCOM.001.2300182

[14] V. K. Thatikonda, "Serverless Computing: Advantages, Limitations and Use Cases," *European Journal of Theoretical and Applied Sciences*, vol. 1, no. 5, pp. 341–347, 2023. [Online]. Available: https://doi.org/10.59324/ejtas.2023.1(5).25

[15] F. Tusa *et al.*, "Microservices and serverless functions—lifecycle, perfor- mance, and resource utilisation of edge based real-time IoT analytics," *Future Generation Computer Systems*, vol. 155, pp. 204–218, 2024. [Online]. Available: https://doi.org/10.1016/j.future.2024.02.006