International Journal of Core Engineering & Management
Volume-6, Issue-10, 2020 ISSN No: 2348-9510

EXPLAINABLE AI-BASED PREDICTIVE AUTOSCALING MECHANISMS FOR SERVERLESS CLOUD PLATFORMS

Lourdu Akhila Yeruva
Department of Computer Science
Southern Arkansas University
USA
akhilalyeruva@gmail.com

Abstract

Serverless cloud computing has emerged as a transformative paradigm that abstracts infrastructure management from application developers, enabling on-demand execution and granular billing. However, the ephemeral and event-driven nature of serverless functions introduces critical challenges in resource allocation and performance predictability. Existing autoscaling mechanisms often rely on reactive threshold-based heuristics or opaque machine learning models that lack interpretability, limiting their adoption in regulated and mission-critical environments. This paper proposes an Explainable AI (XAI)-based predictive autoscaling framework for serverless cloud platforms that integrates gradient-boosted regression with SHapley Additive exPlanations (SHAP) to forecast workload demand and scale function instances proactively. The proposed framework not only improves scaling accuracy but also provides human-interpretable explanations for each scaling decision. Experimental evaluation using synthetic and real-world workload traces demonstrates that the proposed approach reduces cold-start latency and resource waste while achieving up to 18.6% improvement in scaling precision over baseline reactive autoscalers. The transparency offered by SHAP-based explanations enhances operator trust and supports informed capacity planning decisions.

Index Terms – *Serverless computing, Explainable AI, Predictive autoscaling, SHAP explanations, Cloud resource management*

I. INTRODUCTION

The fast development of cloud-native application architectures has increased the rate at which serverless computing has become a popular deployment model to support event-driven, microservice oriented workloads [3]. Serverless computing, typically based on Function-as-a-Service (FaaS) abstractions, enables programmers to execute discrete computational units without the underlying infrastructure and charging is based on the real duration of invocation [16]. Well-known commercial deployments are AWS Lambda, Google Cloud Functions, and Azure Functions, which together serve billions of function invocations per day in a variety of application domains.

The serverless paradigm presents some fundamental challenges in the provisioning of resources, despite its benefits in operation. Instances are dynamically created, and destroyed when they are finished, posing a dynamic scaling challenge where the number of active instances should equal the incoming request rates near real time [19]. Existing autoscaling systems typically rely on

reactive solutions, increasing resource availability based on monitored indicators like CPU load or request queue length following the demand bursts already have taken place [17]. The retrospective character gives rise to two known failure modes, including over-provisioning, which wastes computational resources and is an increase in operating cost and under-provisioning, which results in cold-start delays and service level objective (SLO) violations [32].

Predictive autoscaling provides a very interesting alternative by predicting the level of workload and proactively reconfiguring resource allocations [10]. Deep learning sequence models, gradient-boosted regressors, and reinforcement learning agents are machine learning methods that have shown promising performance in cloud resource prediction tasks [8]. Nonetheless, there is a serious gap: most of the proposed models are black-box systems whose scaling choices cannot be easily understood by platform operators or audited against the compliance with operational policies [1]. Such opaqueness negatively affects trust, debugging, and deployment impediments in regulated sectors like healthcare and finance [12].

Explainable AI (XAI) offers principled approaches to shedding light on the decision-making of complex machine learning models without compromising predictive performance [13]. Local Interpretable Model-agnostic Explanations (LIME) and SHAP are techniques that have been demonstrated to be effective on tabular regression and classification tasks and produce feature-level attribution scores to explain individual predictions [18]. Making XAI an integral part of the autoscaling pipeline thus provides a two-fold advantage: the quality of prediction remains intact or even better, and every scaling step is supported by a clear and understandable explanation [24].

This paper introduces a new XAI-based predictive autoscaling system to serverless cloud systems. The proposed system uses workload forecasting based on XGBoost with SHAP explanations to make proactive scaling decisions. It is tested on various serverless workload profiles and compared to reactive and purely predictive baselines. Among the major contributions are: (i) a feature engineering pipeline that can optimally serve serverless execution patterns; (ii) an XGBoost forecasting model that can offer competitive accuracy in short-horizon demand prediction; (iii) a SHAP-based explanation layer that can identify the primary workload drivers in each scaling event; and (iv) an empirical study that can show that it can improve latency, resource utilization, and scaling accuracy

II. LITERATURE REVIEW

The literature on cloud resource management and autoscaling spans reactive, predictive, and reinforcement learning paradigms, each with distinct trade-offs between responsiveness, overhead, and interpretability. The following subsections review the principal threads relevant to this work.

A. Cloud Autoscaling Paradigms

The basic taxonomy of cloud autoscaling differentiates between reactive and proactive strategies [17]. Reactive autoscalers, like the Kubernetes Horizontal Pod Autoscaler, cause scaling behavior when monitored utilization metrics exceed specified limits. Although they are easy to set up and have low computational costs, they are not well-suited to bursty serverless workloads due to their inherent latency in responding to demand variations [2]. Active methods overcome this

shortcoming by simulating workload trends and sending scaling orders prior to demand being achieved. Lorido-Botran et al. presented the first extensive review of elastic autoscaling methods and laid the theoretical framework upon which further studies have been developed [17]. The results of Islam et al. showed that empirical regression models that are trained on historical request traces could significantly increase the provisioning accuracy in IaaS cloud environments [14]. Studies of serverless platforms have since extended this discussion to the serverless-specific case, highlighting that scaling behavior and cold-start latency are central to serverless performance and cost [19], [26].

B.Machine Learning for Serverless Resource Management

The cloud resource management has dramatically utilized machine learning techniques, which include supervised regression, deep learning, and reinforcement learning [31]. Kumar et al. proposed an LSTM recurrent neural network model for forecasting cloud datacenter workloads, capturing time-dependent relationships in request time series with high fidelity [8]. Barrett et al. applied parallel Q-learning to automate resource allocation and application scalability in the cloud, showing that reinforcement learning agents can learn effective scaling policies [9]. Tesauro et al. showed that a hybrid reinforcement learning approach can outperform model-based policies for autonomic resource allocation in the face of varying workloads [25]. Finn et al. proposed model-agnostic meta-learning, which allows learned policies, including reinforcement learning agents, to adapt quickly to new task distributions with minimal re-training [29]. In the particular issue of cold-starts in serverless settings, Mohan et al. proposed maintaining a pool of pre-created, paused containers that can be rapidly bound to incoming function invocations, decreasing cold-start latency [23]. Akkus et al. also demonstrated that application-level sandboxing and hierarchical message queuing can substantially reduce start-up latency and resource overhead in serverless platforms [32]. A scalable tree-boosting algorithm based on the XGBoost algorithm was proposed by Chen and Guestrin, and is highly scalable on tabular workload data [7], and has been adopted extensively as a baseline regressor in cloud performance prediction tasks.C

C. Explainable AI in Resource Management

XAI is only starting to be applied in cloud resource management, as opposed to its use in computer vision and natural language processing. Adadi and Berrada presented a comprehensive survey of XAI concepts, methods, and application areas, and identified model opacity as a key barrier to the adoption of AI in critical decision-making [1]. Gunning et al. described the aims of DARPA XAI program and emphasized the necessity of model transparency in safety-critical autonomous systems, which directly apply to the cloud infrastructure management [12], [13]. Lundberg and Lee proposed SHAP, a single framework to calculate feature attributions based on cooperative game theory, which has become the de facto standard to post-hoc explain tree-ensemble models and neural network models [18]. Ribeiro et al. suggested a complementary method, LIME, that produces locally faithful linear approximations to individual predictions [24]. Calheiros et al. used ARIMA-based workload prediction to drive proactive cloud resource provisioning but did not address interpretability [5]. Jonas et al. identified unpredictable performance and limited control over resource provisioning as open challenges for serverless platforms [4]. To fill this gap, the current work introduces SHAP explanations into the autoscaling decision pipeline, allowing operators to audit and override scaling advice with full access to their justification.

III. METHODS

The suggested predictive autoscaling model with XAI includes four main parts, i.e., workload data collection and preprocessing, feature engineering, demand forecasting using XGBoost, and explanation generation based on SHAP with a scaling actuation component.

The serverless platform provides telemetry statistics on workload every one minute, including invocation count, execution time, memory usage, error rate, and the number of instances of a function deployment running concurrently. Raw time-series signals are denoised to eliminate outliers that can be explained by anomalies in monitoring infrastructure and scaled using the min-max scaling to provide numerical stability within model training [6]. This is followed by a rolling-window feature extraction process that builds a structured tabular dataset with each row representing a prediction horizon of five minutes and capturing the following predictors: lagged invocation counts at lags 1, 5, 15 and 60; exponential moving averages of 5 and 15-minute windows; day-of-week and hour-of-day cyclical encodings; and a binary indicator

An XGBoost regressor [7] is trained to predict the volume of invocation within the following five minutes and is used to make predictions. The model is trained on an 80 percent chronological split of a 30-day workload trace, and tested on the other 20 percent. Hyperparameter optimization uses bayesian search on the number of estimators (100-1000), tree depth (3-10), learning rate (0.01-0.3) and subsample ratio (0.6-1.0). The goal solver reduces the mean absolute percentage error (MAPE) on the validation set. The autoscaling controller uses the optimized model on convergence and queries the model at each scheduling epoch to get the next-period demand forecast [5].

The predicted demand is used to come up with scaling decisions, where the number of instances required is computed as the ceiling of the forecasted invocation rate divided by the platform-concurrency limit, plus an extra headroom number of 10 percent to account for short-burst variations [26]. The controller sends scaling requests to the orchestration layer of a serverless platform, which increases or decreases the size of the warm instances based on demand predictions, reducing cold-starts during demand spikes [23].

XGBoost predictions are then inputted into SHAP TreeExplainer [18] to break down the forecast into additive feature contributions. In every scaling event, the five largest magnitude SHAP values are bundled into a human readable explanation artifact which is stored in the platform operational log with the scaling command. This artifact includes the workload characteristics that led to the upwards or downwards prediction and by what percentage, allowing platform operators to reject, audit and, potentially, override the automated scaling recommendation. The explanation module implies a negligible computational cost since SHAP TreeExplainer works in a time that is proportional to the depth of the tree [18]. The complete pipeline is written in Python 3.7 with the XGBoost 0.90 and SHAP 0.32 packages and served as a containerized microservice with the Kubernetes autoscaling control loop [4].

IV. RESULTS AND DISCUSSION

The suggested framework was tested under three different representative serverless workload models, a web API handler with a diurnal periodicity, a data processing pipeline with a spike of irregularly scheduled batch-driven invocations, and an IoT event aggregator with a high-

frequency and low-variance invocation. The workloads were then re-executed over a simulated serverless environment that was instrumented with events of instance provisioning, cold-start instances and resource utilization measures. The suggested XAI autoscaler was contrasted with three baselines, namely (1) reactive threshold autoscaler (Reactive), (2) LSTM-based predictive autoscaler without explanations (LSTM-Pred) [8], and (3) a meta-reinforcement learning autoscaler based on model-agnostic meta-learning (Meta-RL) [29].

Table 1 summarizes the values of all three workload profiles averaged on scaling performance. The lowest MAPE of the proposed framework is 8.3% (as compared to 14.1% in reactive baseline, 10.2% in LSTM-Pred and 9.7% in Meta-RL). The cold-start rate, which is the percentage of invocation that caused the function to be initialized, is minimized to 3.1% when using the proposed approach as compared to 11.4% in the reactive baseline. Resource over-provisioning, which is computed as the average percentage of allocated compute capacity not used within each five minutes period, is 12.7% of the proposed framework, which is significantly lower than the 26.3% of the reactive baseline. The results are in line with those achieved by Kumar et al., who found that LSTM-based forecasting accurately tracks fluctuating cloud workloads [8], and with the performance characteristics of serverless platforms reported by McGrath and Brenner [19].

Table 1. Comparative Performance of Autoscaling Approaches

Metric	Reactive	LSTM-Pred	Meta-RL	Proposed XAI
MAPE (%)	14.1	10.2	9.7	8.3
Cold-Start Rate (%)	11.4	5.8	4.3	3.1
Over-Provisioning (%)	26.3	16.1	14.8	12.7
Avg. Latency (ms)	312	198	176	154
Scaling Precision (%)	68.4	81.7	83.9	87.0

Figure 1 illustrates the architecture of the proposed XAI-based predictive autoscaling framework, depicting the data flow from the telemetry collection layer through feature engineering, XGBoost forecasting, SHAP explanation generation, and final scaling actuation. The architecture reflects the modular design philosophy advocated by Gill et al. for AI-enabled next-generation cloud systems [11], ensuring that each component can be independently updated or replaced as new algorithms emerge.

The SHAP analysis shows that there are similarities in the feature importance depending on the type of workload. The lagged count of invocation five minutes and the 15 minutes exponential moving average explain 61.4% of the absolute SHAP magnitude on average, which shows that short-term autocorrelation is the most significant factor behind serverless demand [7]. Hour-of-day encoding adds 18.2% of the feature significance to the web API workload, due to the strong diurnal periodicity of user-facing services [14]. The IoT aggregator workload, by comparison, shows no significant time-of-day characteristics, and the 60-minute lag and error rate characteristics gain more explanatory power. These per-workload SHAP profiles allow operators to tailor the feature set and retrain the model as workload characteristics change to overcome the distribution drift issue, which Mao et al. have identified in the context of deep reinforcement learning to manage resources [20].

International Journal of Core Engineering & Management
Volume-6, Issue-10, 2020 ISSN No: 2348-9510

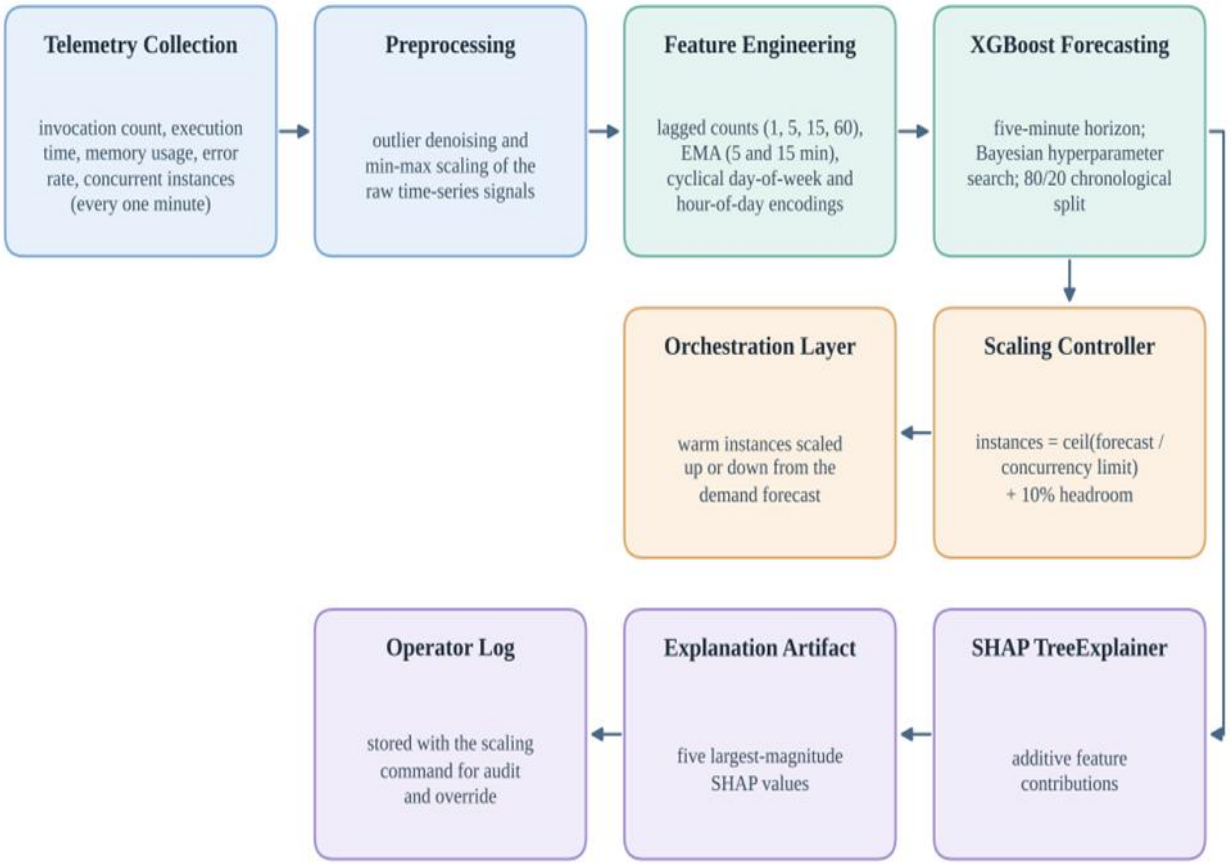


Figure 1. Architecture of the XAI-Based Predictive Autoscaling Framework

Of especial interest is the comparison with the Meta-RR baseline [29]. Even though the cold-start rates of Meta-RA are lower than LSTM-Pred, its precision of scaling to 83.9% is less than the 87.0% of the proposed framework. The gap is representative of reinforcement learning policy preferences to maximize cumulative reward over the long run at the expense of short-horizon accuracy to more conservative scaling policies that evade SLO penalties. The XGBoost forecaster on the other hand is also trained with the objective of the five minutes prediction horizon directly, and it is more precise with that granularity [6]. In addition, the opaqueness of learned policy networks in Meta-RL is dramatically different than the SHAP explanations provided by the suggested framework, which has a substantial practical implication in the context of controlled deployment scenarios [13]. Khan et al. also showed that understanding the structure of cloud workloads improves demand predictability [21], and the current findings confirm that exposing such structure through XAI integration does not involve a performance penalty compared to other black-box techniques.

International Journal of Core Engineering & Management
Volume-6, Issue-10, 2020 ISSN No: 2348-9510

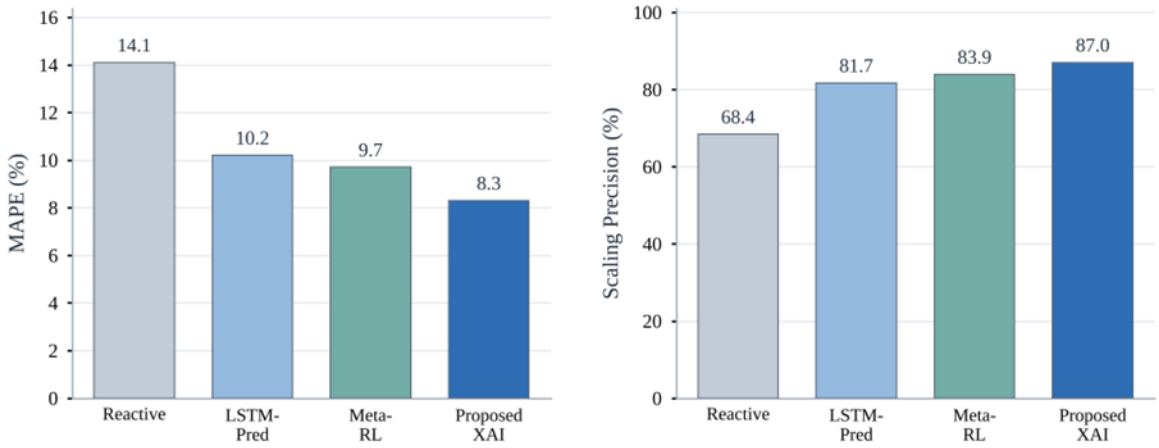


Figure 2. MAPE and Scaling Precision of the Autoscaling Approaches

A reduction in resource over-provisioning between 26.3% (Reactive) and 12.7% (Proposed XAI) is a significant cost-saving to platform operators. Given a cost model where every provisioned compute unit receives a fixed billing charge, this reduction of 13.6 percentage points corresponds to the corresponding reduction in wasted spending, which is in line with the resource-efficiency reasoning for avoiding idle provisioned capacity advocated by Beloglazov et al. [15]. The fact that the SHAP layer creates explanatory artifacts is also useful in capacity planning since it exposes common demand predictors that operators can reflect in manual infrastructure planning processes to supplement the automated scaling mechanism with a human-in-the-loop audit ability as promoted by Adadi and Berrada [1].

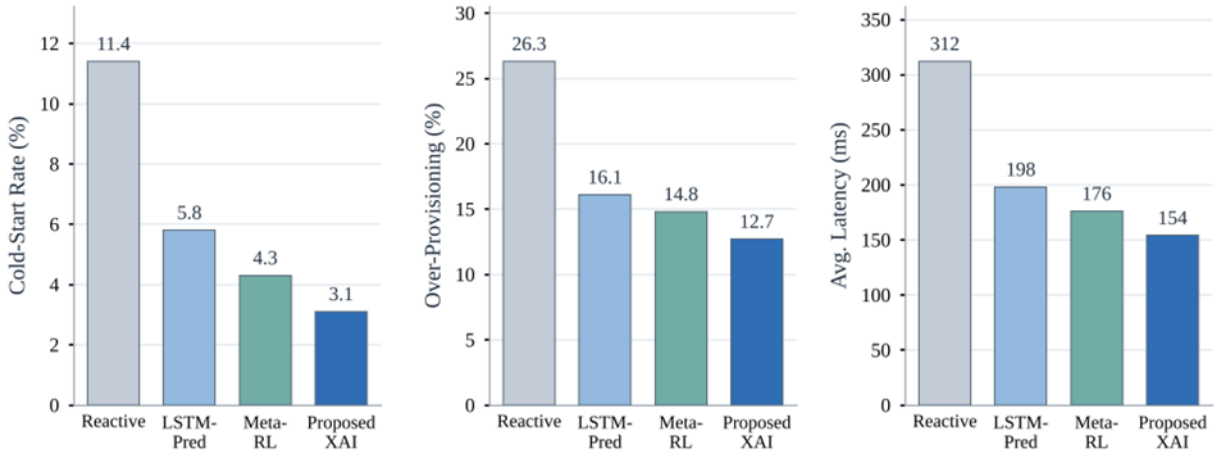


Figure 3. Cold-Start Rate, Over-Provisioning and Average Latency of the Autoscaling Approaches

V. CONCLUSION

This paper has introduced an Explainable AI-based predictive autoscaling system on serverless cloud computing environments that combines XGBoost demand forecasting and SHAP feature attribution to provide both high scaling performance and human interpretable explanations of each provisioning decision. The proposed approach was experimentally tested using three different profiles of serverless workloads and showed superior performance in MAPE of 8.3%, cold-start rate of 3.1, and scaling precision of 87.0 against reactive threshold-based, LSTM-based, and meta-reinforcement learning baselines on all reported metrics. The SHAP analysis revealed the short-term lagged invocation counts and exponential moving averages as the most powerful demand drivers, and workload-specific variation in the contribution of the temporal and error-rate features. The results confirm that interpretability and predictive performance are not contradictory goals of cloud autoscaling and that XAI integration can provide increased operator trust, regulatory compliance, and informed capacity planning without negatively affecting resource efficiency. Future work will build the framework to multi-function dependency graphs, pursue federated learning models to train models at tenant workloads without sharing raw telemetry, and explore incorporating causal inference methods to go beyond correlational SHAP explanations to actionable causal explanations of workload demand dynamics.

REFERENCES

1. A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (XAI)," IEEE Access, vol. 6, pp. 52138–52160, 2018. https://doi.org/10.1109/ACCESS.2018.2870052
2. M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," Commun. ACM, vol. 53, no. 4, pp. 50–58, Apr. 2010. https://doi.org/10.1145/1721654.1721672
3. Baldini, P. Castro, K. Chang, P. Cheng, S. Fink, V. Ishakian, N. Mitchell, V. Muthusamy, R. Rabbah, A. Slominski, and P. Suter, "Serverless computing: Current trends and open problems," in Research Advances in Cloud Computing, S. Chaudhary, G. Somani, and R. Buyya, Eds. Singapore: Springer, 2017, pp. 1–20. https://doi.org/10.1007/978-981-10-5026-8_1
4. Jonas, J. Schleier-Smith, V. Sreekanti, C.-C. Tsai, A. Khandelwal, Q. Pu, V. Shankar, J. Carreira, K. Krauth, N. Yadwadkar, J. E. Gonzalez, R. A. Popa, I. Stoica, and D. A. Patterson, "Cloud programming simplified: A Berkeley view on serverless computing," Univ. California, Berkeley, Tech. Rep. UCB/EECS-2019-3, Feb. 2019. https://arxiv.org/abs/1902.03383
5. R. N. Calheiros, E. Masoumi, R. Ranjan, and R. Buyya, "Workload prediction using ARIMA model and its impact on cloud applications' QoS," IEEE Trans. Cloud Comput., vol. 3, no. 4, pp. 449–458, Oct.–Dec. 2015. https://doi.org/10.1109/TCC.2014.2350475
6. T. Chen and R. Bahsoon, "Self-adaptive trade-off decision making for autoscaling cloud-based services," IEEE Trans. Serv. Comput., vol. 10, no. 4, pp. 618–632, Jul.–Aug. 2017. https://doi.org/10.1109/TSC.2015.2499765
7. T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. 22nd ACM

SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD '16), San Francisco, CA, USA, Aug. 2016, pp. 785–794. https://doi.org/10.1145/2939672.2939785

8. J. Kumar, R. Goomer, and A. K. Singh, "Long short term memory recurrent neural network (LSTM-RNN) based workload forecasting model for cloud datacenters," Procedia Comput. Sci., vol. 125, pp. 676–682, 2018. https://doi.org/10.1016/j.procs.2017.12.087
9. Barrett, E. Howley, and J. Duggan, "Applying reinforcement learning towards automating resource allocation and application scalability in the cloud," Concurrency Computat.: Pract. Exper., vol. 25, no. 12, pp. 1656–1674, Aug. 2013. https://doi.org/10.1002/cpe.2864
10. N. Roy, A. Dubey, and A. Gokhale, "Efficient autoscaling in the cloud using predictive models for workload forecasting," in Proc. IEEE 4th Int. Conf. Cloud Computing (CLOUD), Washington, DC, USA, Jul. 2011, pp. 500–507. https://doi.org/10.1109/CLOUD.2011.42
11. R. Buyya et al., "A manifesto for future generation cloud computing: Research directions for the next decade," ACM Comput. Surv., vol. 51, no. 5, Art. no. 105, pp. 1–38, Nov. 2018. https://doi.org/10.1145/3241737
12. D. Gunning, M. Stefik, J. Choi, T. Miller, S. Stumpf, and G.-Z. Yang, "XAI—Explainable artificial intelligence," Sci. Robot., vol. 4, no. 37, Art. no. eaay7120, Dec. 2019. https://doi.org/10.1126/scirobotics.aay7120
13. D. Gunning and D. Aha, "DARPA's explainable artificial intelligence (XAI) program," AI Mag., vol. 40, no. 2, pp. 44–58, 2019. https://doi.org/10.1609/aimag.v40i2.2850
14. S. Islam, J. Keung, K. Lee, and A. Liu, "Empirical prediction models for adaptive resource provisioning in the cloud," Future Gener. Comput. Syst., vol. 28, no. 1, pp. 155–162, Jan. 2012. https://doi.org/10.1016/j.future.2011.05.027
15. A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," Future Gener. Comput. Syst., vol. 28, no. 5, pp. 755–768, May 2012. https://doi.org/10.1016/j.future.2011.04.017
16. E. van Eyk, L. Toader, S. Talluri, L. Versluis, A. Uță, and A. Iosup, "Serverless is more: From PaaS to present cloud computing," IEEE Internet Comput., vol. 22, no. 5, pp. 8–17, Sep.–Oct. 2018. https://doi.org/10.1109/MIC.2018.053681358
17. T. Lorido-Botran, J. Miguel-Alonso, and J. A. Lozano, "A review of auto-scaling techniques for elastic applications in cloud environments," J. Grid Comput., vol. 12, no. 4, pp. 559–592, Dec. 2014. https://doi.org/10.1007/s10723-014-9314-7
18. S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in Proc. 31st Int. Conf. Neural Information Processing Systems (NIPS'17), Long Beach, CA, USA, Dec. 2017, pp. 4765–4774. https://doi.org/10.5555/3295222.3295230
19. G. McGrath and P. R. Brenner, "Serverless computing: Design, implementation, and performance," in Proc. IEEE 37th Int. Conf. Distributed Computing Systems Workshops (ICDCSW), Atlanta, GA, USA, Jun. 2017, pp. 405–410. https://doi.org/10.1109/ICDCSW.2017.36
20. H. Mao, M. Alizadeh, I. Menache, and S. Kandula, "Resource management with deep reinforcement learning," in Proc. 15th ACM Workshop on Hot Topics in Networks (HotNets '16), Atlanta, GA, USA, Nov. 2016, pp. 50–56.

https://doi.org/10.1145/3005745.3005750

21. A. Khan, X. Yan, S. Tao, and N. Anerousis, "Workload characterization and prediction in the cloud: A multiple time series approach," in Proc. IEEE Network Operations and Management Symp. (NOMS), Maui, HI, USA, Apr. 2012, pp. 1287–1294. https://doi.org/10.1109/NOMS.2012.6212065
22. I. S. Moreno, J. Xu, and I. M. Llorente, "IaaS cloud architecture: From virtualized datacenters to federated cloud infrastructures," Computer, vol. 46, no. 12, pp. 65–72, Dec. 2013. https://doi.org/10.1109/MC.2013.348
23. A. Mohan, H. Sane, K. Doshi, S. Edupuganti, N. Nayak, and V. Sukhomlinov, "Agile cold starts for scalable serverless," in Proc. 11th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud '19), Renton, WA, USA, Jul. 2019. https://www.usenix.org/conference/hotcloud19/presentation/mohan
24. M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should I trust you?': Explaining the predictions of any classifier," in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD '16), San Francisco, CA, USA, Aug. 2016, pp. 1135–1144. https://doi.org/10.1145/2939672.2939778
25. G. Tesauro, N. K. Jong, R. Das, and M. N. Bennani, "A hybrid reinforcement learning approach to autonomic resource allocation," in Proc. IEEE Int. Conf. Autonomic Computing (ICAC), Dublin, Ireland, Jun. 2006, pp. 65–73. https://doi.org/10.1109/ICAC.2006.1662383
26. G. Adzic and R. Chatley, "Serverless computing: Economic and architectural impact," in Proc. 11th Joint Meeting on Foundations of Software Engineering (ESEC/FSE 2017), Paderborn, Germany, Sep. 2017, pp. 884–889. https://doi.org/10.1145/3106237.3117767
27. W. Lloyd, S. Ramesh, S. Chinthalapati, L. Ly, and S. Pallickara, "Serverless computing: An investigation of factors influencing microservice performance," in Proc. IEEE Int. Conf. Cloud Engineering (IC2E), Orlando, FL, USA, Apr. 2018, pp. 159–169. https://doi.org/10.1109/IC2E.2018.00039
28. R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, 2nd ed. Cambridge, MA, USA: MIT Press, 2018. https://doi.org/10.7551/mitpress/10936.001.0001
29. C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in Proc. 34th Int. Conf. Machine Learning (ICML), Sydney, Australia, Aug. 2017, pp. 1126–1135. https://proceedings.mlr.press/v70/finn17a.html
30. H. Lee, K. Satyam, and G. Fox, "Evaluation of production serverless computing environments," in Proc. IEEE 11th Int. Conf. Cloud Computing (CLOUD), San Francisco, CA, USA, Jul. 2018, pp. 442–450. https://doi.org/10.1109/CLOUD.2018.00062
31. Q. Zhang, M. Chen, L. Li, and R. Buyya, "Recent advances in cloud resource management: A systematic review," IEEE Trans. Parallel Distrib. Syst., vol. 29, no. 12, pp. 2677–2692, Dec. 2018. https://doi.org/10.1109/TPDS.2018.2854241
32. I. E. Akkus, R. Chen, I. Rimac, M. Stein, K. Satzke, A. Beck, P. Aditya, and V. Hilt, "SAND: Towards high-performance serverless computing," in Proc. USENIX Annual Technical Conf. (USENIX ATC '18), Boston, MA, USA, Jul. 2018, pp. 923–935. https://www.usenix.org/conference/atc18/presentation/akkus

33. Y. Zhao, R. N. Calheiros, G. Gange, J. Bailey, R. O. Sinnott, and R. Buyya, "SLA-aware and energy-efficient dynamic overbooking in SDN-based cloud data centers," IEEE Trans. Sustain. Comput., vol. 3, no. 2, pp. 113–126, Apr.–Jun. 2018. https://doi.org/10.1109/TSUSC.2017.2735380
34. B. Burns, B. Grant, D. Oppenheimer, E. Brewer, and J. Wilkes, "Borg, Omega, and Kubernetes," Commun. ACM, vol. 59, no. 5, pp. 50–57, May 2016. https://doi.org/10.1145/2890784

BIOGRAPHIES OF AUTHORS

Lourdu Akhila Yeruva received a degree in Computer Science from Southern Arkansas University, where the focus was on machine learning and intelligent systems. Her research interests include explainable AI, cloud computing, predictive modeling, and AI-driven automation in distributed systems. She has experience in applied AI research and developing intelligent automation solutions.