ISSN: P-2455-0515
E- 2394-8450

Educreator Research Journal

OPEN ACCESS

Volume–XIII, Special Issues–I(b) Jan – Feb 2026

Original Research Article

A CROSS-CLOUD STUDY OF COLD START DETECTION IN AZURE AND GCP SERVERLESS ENVIRONMENTS

** Ms. Asmi Jadhav, **Ms. Pratibha Dhamane & ***Ms. Vandana Maurya*

,Students, ***Assistant Professor, Department of Information Technology, B.K. Birla College, (Empowered Autonomous Status), Kalyan.*

Abstract:

Cold start latency is a well-known bottleneck in serverless computing, often causing noticeable delays when functions are invoked after periods of inactivity. In multi-cloud environments, differences in architecture, resource management, and workload behaviour introduce additional complexity. An experimental study on predicting cold start events and comparing performance across two widely used serverless platforms: Microsoft Azure Functions and Google Cloud Platform Cloud Functions.

Large-scale execution datasets are collected independently from both platforms and used to train and evaluate the models within their respective cloud environments to detect cold start events based on latency percentiles, request arrival patterns, memory usage, and temporal features to check cross cloud cold start detection the models were trained and tested separately on datasets from Azure-2019 and Google Cloud Functions-cold start dataset.

The results show that both the platform Azure and GCP have different features, so in the experiment, both datasets were trained and tested with their respective cloud environment, and the models used were Logistic Regression, XGBoost, and Random Forest classifiers. These models perform accurately in a cloud environment. But cross-cloud detection of cold start using machine learning performs poorly due to different architectures and features. The research suggests that cross-cloud cold start detection can be done by collecting real world dataset. In both the experiment, XGBoost and Random Forest out performs the Logistic Regression. Taken together, the findings emphasize the strength of tree-based models in capturing complex, non-linear performance dynamics that linear techniques are unable to represent effectively.

Keywords: *Serverless Computing, Cold Start Detection, Machine Learning, Performance Analysis, Azure Functions, Google Cloud Platform.*

Introduction:

Serverless computing has evolved into modern cloud applications, mainly because it makes development, deployment easy. The cloud handle servers, scaling, or infrastructure details, developers can just write code small functions run only when they're needed, but cold start remains major issue.

Cold start happens when function hasn't running for long time and needs to work when required. When request trigger the platform needs time to prepare the environment, load libraries, and allocate resources but applications that depend on fast responses, it can be a real problem. This highlight cold start is mostly affected by workload patterns, memory settings, runtime choices, and its internal systems design.

ISSN: P-2455-0515
E- 2394-8450

Educreator Research Journal

OPEN ACCESS

Volume–XIII, Special Issues–I(b) Jan – Feb 2026

Original Research Article

When organizations try to move applications to a multi cloud setup, improve reliability or avoid vendor locked, cold start is performance issue behaviour. Hence, comparing cold start behaviour across platforms like Microsoft Azure Functions and Google Cloud Platform Cloud Functions becomes important, especially for developers who want predictable performance.

Many research, execution data have tried to predict, when a cold start will occur and severe it might be. They applied different techniques ranging from simple statistical models to more complex ensemble and deep learning approaches have shown encouraging results. These leaves a gap of cross cloud cold start reduction where performance, consistency, reliability is essential. The paper presents control study of cross cloud comparison using ML models and are trained and tested using datasets collected separately from Azure and Google Cloud. The analysis focuses on latency behaviour, request arrival trends, memory usage, and time- based features. By comparing results across both platforms, the study highlights how cloud specific characteristics influence cold start prediction and show why platform aware modelling matters in real world, multi cloud serverless systems.

Literature Review:

1. **Cold Start**

   Serverless computing become a widely adopted cloud paradigm due to its scalability and cost-efficiency. Cloud provides allow developers to deploy applications without managing infrastructure. However, cold start latency remains one of the major performance challenges in serverless environments which affect the performance of applications.

   Cold start occurs when a function is invoked after a period of inactivity, requiring the initialization of a new execution environment. This process includes container provisioning, runtime initialization, and dependency loading, resulting in additional latency (Manner et al., 2018) several studies have analysed cold start behaviour and confirmed that it highly impacts latency-sensitive applications (Verma et al., 2024; Golec et al., 2024). Factors such as idle duration, memory allocation, runtime language, and traffic burst patterns strongly influence cold start latency (Chaudhary et al., 2024).

2. **Cold Start Mitigation Techniques**

   Different strategies have been proposed to reduce cold start delays. Pre-warming techniques maintain active instances to avoid reinitialization overhead (Mohan et al., 2019). Lightweight virtualization approaches aim to reduce container startup time (Karam Zadeh & Shameli-Sendi, 2024). Application-aware optimization methods have been explored to improve function initialization efficiency (Bermbach et al., 2020).

   Other studies have proposed predictive and adaptive provisioning strategies to reduce cold start frequency (Agarwal et al., 2021; Khan, 2025). While these approaches improve latency, they may increase resource consumption and operational cost.

3. **Machine Learning for Cold Start Prediction**

   Research has shifted to machine learning-based cold start detection and prediction done using supervised learning models to classify cold and warm invocations using historical execution data and resource utilization metrics (Saravana Kumar & Selvakumar Samy, 2025). Advanced ensemble methods such as XGBoost have

ISSN: P-2455-0515
E- 2394-8450

Educreator Research Journal

OPEN ACCESS

Volume–XIII, Special Issues–I(b) Jan – Feb 2026

Original Research Article

shown high predictive performance by capturing nonlinear feature relationships (Liu et al., 2023). These predictive techniques enable proactive mitigation strategies, improving performance while maintaining cost efficiency. However, most existing studies evaluate prediction models within a single cloud environment, limiting understanding of cross-platform generalization.

4. **Cross-Cloud Challenges and Research Gap**

   Although cold start optimization has been extensively studied, cross-cloud evaluation remains limited. Differences in infrastructure design, container lifecycle management, and scaling policies across cloud providers can introduce domain shifts that affect model transferability (Golec et al., 2024).

   Additionally, some studies define cold start events using latency thresholds while also including latency-related features in prediction models. This may lead to feature dominance or potential data leakage (Manner et al., 2018).

   These limitations highlight the need for systematic cross-cloud comparative studies using consistent methodologies.

5. **Research Gap and Contribution**

   From the literature, it is evident that, cold start mitigation and prediction have been widely explored (Chaudhary et al., 2024; Verma et al., 2024). However, the research systematically compared cold start detection across different serverless platforms using the same experimental framework.

   This study fills the gap by conducting a comparative evaluation of machine learning-based cold start prediction across Azure and GCP serverless environments. By training and testing models independently on both platforms, this research analyses cross-cloud variability and evaluates model generalization capability in heterogeneous cloud infrastructures.

Methodology:

This study follows an experimental approach to analyse and compare cold start prediction performance across two cloud platforms: Microsoft Azure Functions and Google Cloud Platform Cloud Functions. The methodology consists of data collection, data cleaning, feature engineering, training of model, and performance evaluation.

1. **Data Collection**

   Execution trace datasets were collected independently from Azure and Google Cloud environments. The datasets contain large-scale records of serverless function invocations, including response latency, memory allocation, invocation timestamps, and request patterns. We used the ATOM dataset (Golec et al.,2023) to train and validate. Cold start events were identified using latency-based percentile thresholds, where unusually high initialization delays were labelled as cold starts.

2. **Data Cleaning**

   The datasets were cleaned to remove missing or inconsistent records. Feature normalization was implemented to ensure uniform scaling across numerical attributes. The data was separated in training and testing sets within each cloud environment to allow platform-specific evaluation.

ISSN: P-2455-0515
E- 2394-8450

Educreator Research Journal

OPEN ACCESS

Volume–XIII, Special Issues–I(b) Jan – Feb 2026

Original Research Article

3. **Feature Engineering**

Essential features were derived from execution logs to improve prediction performance. These include:

i. Request arrival frequency
ii. Inter-arrival time between invocations
iii. Memory allocation metrics
iv. Latency statistics (average, percentile values)
v. Temporal attributes (hour of day, workload trends)

These features were determined based on previous research work indicating their effect on cold start behaviour.

4. **Computational Models**

These supervised ML models were applied in this study:

i. **Logistic Regression** – For baseline linear classification
ii. **Random Forest Classifier** – For ensemble-based nonlinear prediction
iii. **XGBoost Classifier**- Gradient boosting algorithm optimized for performance and scalability.

Each model was trained separately on Azure and GCP datasets. Hyperparameters were tuned using validation data to improve generalization.

5. **Evaluation Strategy**

Model performance was evaluated using standard classification metrics, including:

i. Accuracy
ii. Precision
iii. Recall
iv. F1-Score

To assess platform behaviour, models trained and tested in single cloud environment. Comparative analysis was then conducted to examine differences in prediction performance between Azure and Google Cloud. Performance degradation and feature importance variations were analysed to understand cross-platform behaviour.

Experimental Setup:

The experimental setup was designed to evaluate cold start prediction performance independently on two cloud platforms: Microsoft Azure Functions and Google Cloud Platform Cloud Functions. Both environments were configured to ensure consistent and fair comparison conditions.

1. **Cloud Environment Configuration**

Serverless functions were deployed on Azure and GCP using similar runtime configurations. Memory allocation settings were kept consistent across experiments to minimize variability caused by hardware differences. Default scaling policies provided by each cloud platform were used to reflect real-world deployment scenarios.

ISSN: P-2455-0515
E- 2394-8450

Educreator Research Journal

OPEN ACCESS

Volume–XIII, Special Issues–I(b) Jan – Feb 2026

Original Research Article

2. **Dataset Preparation**

   Execution logs were collected from each platform separately. The Azure dataset training and testing was completed, while the GCP dataset for training and validation GCP-specific models. Each dataset was divided into training (70%) and testing (30%) subsets to ensure proper model validation.

   Cold start events were label using latency percentile thresholds. High latency invocations exceeding the defined threshold were classified as cold starts, while normal invocations were label as warm starts.

3. **Model Training Environment**

   Model training and evaluation were performed using Python-based machine learning libraries. Logistic Regression, Random Forest classifiers, and XGBoost (Extreme Gradient Boosting) were applied using standard supervised learning workflows. Hyperparameter tuning was done with cross-validation to improve prediction accuracy.

4. **Performance Metrics**

   The models were evaluated using:

   i. Accuracy
   ii. Precision
   iii. Recall
   iv. F1-Score

   Feature importance analysis was also conducted for the Random Forest model to identify which execution parameters contributed most to cold start prediction.

5. **Comparative Analysis Strategy**

   After evaluating models independently on each cloud platform, results were compared to analyse differences in prediction performance, feature contribution, and workload behavior. This comparison helped identify platform-specific characteristics influencing cold start detection performance.

Results and Comparative Discussion:

This section highlights a comparative evaluation of Logistic regression, Random Forest, and XGBoost models on Azure and GCP datasets for cold start prediction.

1. Results on Azure Dataset

Model	Accuracy	Macro F1	Weighted F1	ROC-AUC
Logistic Regression	0.78	0.46	0.69	0.786
Random Forest	0.92	0.88	0.92	0.965
XGBoost	0.89	0.85	0.90	0.952

Table 1Performance on Azure Dataset

ISSN: P-2455-0515
E- 2394-8450

Educreator Research Journal

OPEN ACCESS

Volume–XIII, Special Issues–I(b)

Jan – Feb 2026

Original Research Article

XGBoost	Logistic Regression	Random Forest
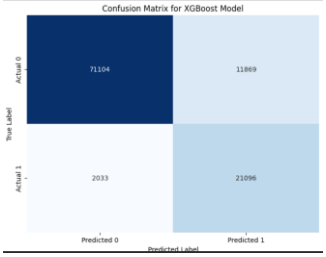	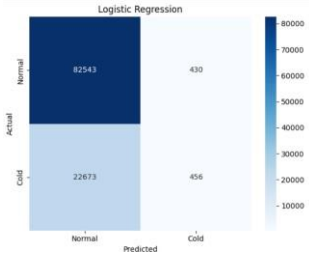	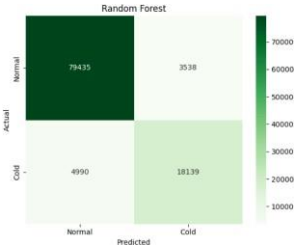
Figure 1XGBoost model	*Figure 2 Logistic Regression*	*Figure 3Random Forset*

Table 2 Confusion Matrix of Azure Models

Observations (Azure):

i. **Random forest** obtained the best accuracy (92%) and ROC-AUC (0.965).

ii. **XGBoost** showed strong cold start recall (0.84).

iii. **Logistic Regression** failed to detect cold starts effectively (recall = 0.02), indicating that linear models are insufficient for complex latency patterns.

Azure results show that ensemble tree-based models significantly outperform linear models in large-scale serverless workloads.

2. Results on GCP Dataset

Model	Accuracy	Macro F1	Weighted F1	ROC-AUC
Logistic Regression	0.98	0.98	0.98	0.981
Random Forest	0.98	0.98	0.98	0.996
XGBoost	0.96	0.96	0.96	0.996

XGBoost	Logistic Regression	Random Forest
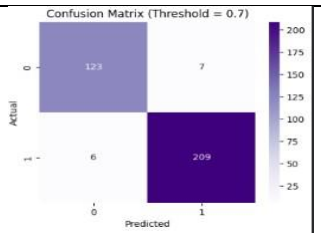	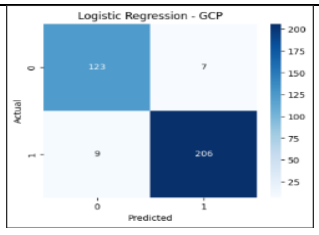	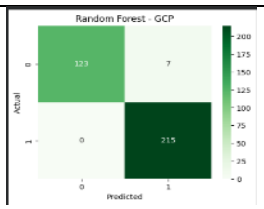
Figure 4 XGBoost -GCP	*Figure 5 Logistic Regression - GCP*	*Figure 6 Random Forset-GCP*

Table 3 Confusion Matrix of GCP Models

ISSN: P-2455-0515
E- 2394-8450

Educreator Research Journal

OPEN ACCESS

Volume–XIII, Special Issues–I(b) Jan – Feb 2026

Original Research Article

Observations (GCP):

i. All models performed significantly better compared to Azure.
ii. Logistic Regression achieved 98% accuracy, unlike in Azure.
iii. Random Forest and XGBoost both achieved ROC-AUC ≈ 0.996.
iv. The smaller dataset size and clearer feature separation likely contributed to higher performance.

3. Cross-Cloud Comparative Analysis

Model	Azure Accuracy	GCP Accuracy	Azure ROC-AUC	GCP ROC-AUC
Logistic Regression	0.78	0.98	0.786	0.981
Random Forest	0.92	0.98	0.965	0.996
XGBoost	0.89	0.96	0.952	0.996

Table 4 Azure vs GCP Comparison

Key Comparative Insights:

1. **Model Stability**
   - Random Forest consistently performs well across both platforms.
   - XGBoost shows strong generalization capability.
   - Logistic Regression is unstable across clouds.
2. Dataset Size Impact
   - Azure dataset (106k samples) is more complex and noisier.
   - GCP dataset (345 samples) is smaller and possibly more separable.
3. Cold Start Detection Capability
   - Ensemble models handle nonlinear latency relationships better.
   - Cross-cloud differences indicate that workload characteristics differ between Azure and GCP.
4. Multi-CloudImplication

   Models trained in one cloud environment may not behave identically in another due to:
   - Resource allocation policies
   - Container initialization mechanisms
   - Scheduling differences
   - Metric distribution variations

Cross-Cloud Generalization:

The cross-cloud evaluation reveals that model performance varies across cloud platforms due to differences in workload characteristics and infrastructure behaviour. Though random forest and XGBoost achieved better accuracy on both Azure and GCP datasets but there are differences in both clouds' accuracy and ROC-AUC indicate that, cold start patterns are platform specific. The Azure dataset, bigger and complex, which prediction increase greater challenges, whereas the smaller GCP dataset showed higher separability. This result explains that machine learning models for cold start detection require platform-aware tuning and feature alignment to

ISSN: P-2455-0515
E- 2394-8450

Educreator Research Journal

OPEN ACCESS

Volume–XIII, Special Issues–I(b) Jan – Feb 2026

Original Research Article

ensure reliable performance in multi-cloud serverless environments.

Conclusion:

This study compared machine learning methods for detecting cold starts in serverless environments on Microsoft Azure and Google Cloud Platform. Using huge Azure traces and a smaller GCP dataset, we trained logistic regression, random Forest, and xgboost models and results show that ensemble methods, especially random forest and xgboost, achieve better result than logistic regression.

On Azure, Random Forest reached best accuracy and ROC-AUC, catching nonlinear latency forms, whereas Logistic Regression faced difficulties with large datasets. On GCP, all models performed better, showing cross cloud differences in workload distribution and platform architecture so it must be customized for specific environments. Totally, collaborative learning proves very effective for detecting cold starts, though adapting approaches across platforms remains a challenge.

Future Work:

Future research can emphasis on collecting larger and more balanced multi-cloud datasets to improve cross-cloud generalization. Advanced models like as LSTM and other deep learning methods can be discovered to better capture temporal workload patterns and transfer learning techniques can help improve model portability between different cloud platforms. Integrating the prediction model into real-time serverless systems for proactive cold start mitigation is another important direction.

References:

1. *Chaudhary, S., Somwanshi, D. K., Rajawat, V. S., Sharma, M., & Verma, P. (2024, December). Optimizing Cold Start Latency in Serverless Computing: A Comprehensive Review of Techniques and Emerging Solutions. In Proceedings of the 6th International Conference on Information Management & Machine Intelligence (pp. 1-8).*
2. *S. Agarwal, M. A. Rodriguez and R. Buyya, "A Reinforcement Learning Approach to Reduce Serverless Function Cold Start Frequency," 2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid), Melbourne, Australia, 2021, pp. 797-803, doi: 10.1109/CCGrid51090.2021.00097.*
3. *keywords: {Runtime environment;Cloud computing;Time-frequency analysis;Fault tolerance;Memory management;Fault tolerant systems;Reinforcement learning;Serverless Computing;Faas;Reinforcement Learning;Q-Learning;Cold Start;Kubeless},*
4. *Saravana Kumar, N., & Selvakumara Samy, S. (2025). Cold start prediction and provisioning optimization in serverless computing using deep learning. Concurrency and Computation: Practice and Experience, 37(4-5), e8392.*
5. *Karamzadeh, A., & Shameli-Sendi, A. (2024). Reducing cold start delay in serverless computing using lightweight virtual machines. Journal of Network and Computer Applications, 232, 104030.*

ISSN: P-2455-0515
E- 2394-8450

Educreator Research Journal

OPEN ACCESS

Volume–XIII, Special Issues–I(b) Jan – Feb 2026

Original Research Article

6. *Khan, M. A. N. H. (2025). Minimizing Cold Starts in Serverless Environments with Predictive Optimization Approach Using Bi-LSTM and Genetic Algorithms (Doctoral dissertation, Dublin, National College of Ireland).*
7. *Arora, A. S., Yachamaneni, T., & Kotadiya, U. (2024). Architectural Optimization of Serverless Big Data Pipelines for AI Workloads Using Cloud Functions and Managed Spark on GCP. International Journal of Emerging Trends in Computer Science and Information Technology, 5(1), 61-68.*
8. *Bermbach, D., Karakaya, A. S., & Buchholz, S. (2020, March). Using application knowledge to reduce cold starts in FaaS services. In Proceedings of the 35th annual ACM symposium on applied computing (pp. 134-143).*
9. *Liu, X., Wen, J., Chen, Z., Li, D., Chen, J., Liu, Y., ... & Jin, X. (2023). Faaslight: General application- level cold-start latency optimization for function-as-a-service in serverless computing. ACM Transactions on Software Engineering and Methodology, 32(5), 1-29.*
10. *Verma, P., Goel, P., & Rani, N. (2024, April). A review: Cold start latency in serverless computing. In 2024 Sixth International Conference on Computational Intelligence and Communication Technologies (CCICT) (pp. 141-148). IEEE.*
11. *Golec, M., Walia, G. K., Kumar, M., Cuadrado, F., Gill, S. S., & Uhlig, S. (2024). Cold start latency in serverless computing: A systematic review, taxonomy, and future directions. ACM Computing Surveys, 57(3), 1-36.*
12. *Manner, J., Endreß, M., Heckel, T., & Wirtz, G. (2018, December). Cold start influencing factors in function as a service. In 2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion) (pp. 181-188). IEEE.*
13. *Mohan, A., Sane, H., Doshi, K., Edupuganti, S., Nayak, N., & Sukhomlinov, V. (2019). Agile cold starts for scalable serverless. In 11th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 19).*
14. *Ristov, S., Hollaus, C., & Hautz, M. (2022, July). Colder than the warm start and warmer than the cold start! experience the spawn start in faas providers. In Proceedings of the 2022 Workshop on Advanced tools, programming languages, and Platform's for Implementing and Evaluating algorithms for Distributed systems (pp. 35-39).*
15. *Golec, M., et al. (2023). ATOM: AI-Powered Sustainable Resource Management for Serverless Edge Computing Environments. IEEE Transactions on Sustainable Computing. https://doi.org/10.1109/TSUSC.2023.3348157 (doi.org in Bing)*

Cite This Article:

Ms. Jadhav A., Ms. Dhamane P. & Ms. Maurya V. (2026). *A Cross-Cloud Study of Cold Start Detection in Azure and GCP Serverless Environments.* **In Educreator Research Journal: Vol. XIII (Issue I),** pp. 37–45.
Doi: https://doi.org/10.5281/zenodo.19916041