A Cloud-Native Approach to Auto-Scaling Web Applications on AWS by using Machine Learning Algorithms

Dr. M.E.,Ph.D.
Assistant Professor , Department of CSE,
Siddharth Institute of Engineering &Technology,
Puttur , AP ,India,
babuskpt@gmail.com

UG Scholar,Department of CIC
Siddharth Institute of Engineering&Technology,
Puttur , AP ,India,
@gmail.com

K. *UG Scholar,Department of CIC*
Siddharth Institute of Engineering&Technology,
Puttur , AP ,India,
@gmail.com

UG Scholar,Department of CIC
Siddharth Institute of Engineering&Technology,
Puttur , AP ,India,
@gmail.com

UG Scholar,Department of CIC
Siddharth Institute of Engineering&Technology,
Puttur , AP ,India,
@gmail.com

UG Scholar,Department of CIC
Siddharth Institute of Engineering&Technology,
Puttur , AP ,India,
@gmail.com

Abstract - This project presents an Internet of Things (IoT)-based dual-axis solar tracking and fault detection system designed to optimize solar energy harvesting and ensure efficient monitoring of the system's health. The system utilizes four Light Dependent Resistors (LDRs) to track the sun's position on two axes, enabling real-time adjustments of the solar panel's orientation using two motors, maximizing energy capture throughout the day. The energy harvested is managed through a charging circuit and stored in a battery, which powers the system. For real-time fault detection, a separate LDR and voltage sensor monitor the panel's performance, detecting abnormalities such as shading or faults in energy output. This data is processed by an ESP32 microcontroller, which drives the motors via a motor driver, and displays system status on a 16x2 LCD screen. The data is also transmitted to an IoT platform (UBIDOTS) via an ESP8266 NodeMCU, enabling remote monitoring and alert notifications to be sent to the user's email. This combination of solar tracking and fault detection improves the system's reliability, efficiency, and responsiveness, making it a robust solution for remote solar installations.

I. INTRODUCTION

The rapid evolution of cloud computing has fundamentally transformed the way modern web applications are designed, deployed, and managed. Traditional on-premise infrastructures required organizations to invest heavily in hardware resources that were often either underutilized during low-traffic periods or insufficient during sudden traffic spikes. Cloud computing overcomes these limitations by offering on-demand access to computing resources, enabling scalability, flexibility, and global availability at significantly reduced upfront costs. As digital services such as e-commerce platforms, streaming applications, online education systems, and enterprise web portals continue to grow, ensuring consistent performance under highly dynamic and unpredictable workloads has become a critical challenge. Users today expect seamless access, low latency, and uninterrupted service regardless of traffic volume, geographical location, or time of access. Any degradation in performance can directly impact user satisfaction, revenue, and organizational reputation.

Auto-scaling has emerged as a core capability in cloud environments to address the challenge of fluctuating workloads. Auto-scaling refers to the automatic adjustment of computing resources—such as virtual machines, containers, or processing capacity—based on real-time demand. In cloud-native architectures, auto-scaling plays a vital role in maintaining application performance while optimizing operational costs. Cloud service providers, particularly Amazon Web Services, offer powerful built-in auto-scaling and monitoring tools that allow applications to dynamically respond to changing traffic conditions. Services such as Elastic Load Balancing distribute incoming requests across multiple instances, while EC2 Auto Scaling ensures that the required number of compute instances are available at all times. Monitoring services continuously track system metrics, enabling automated scaling actions based on predefined rules. Despite these capabilities, many real-world applications still rely on reactive or threshold-based scaling approaches, which may not be sufficient to handle complex and rapidly changing workload patterns.

Paper ID: ICMETA8760

ISBN Number : **978-81-999993-5-0**

One of the major limitations of conventional auto-scaling mechanisms is their dependence on static thresholds and reactive decision-making. In such systems, scaling actions are triggered only after performance metrics such as CPU utilization or memory usage cross predefined limits. While this approach provides basic elasticity, it often results in delayed responses to sudden traffic surges, leading to temporary performance degradation. Conversely, during periods of low demand, excess resources may remain active, causing unnecessary operational costs. These inefficiencies highlight the need for more intelligent and predictive scaling strategies that can anticipate workload changes rather than merely reacting to them. As web applications become more complex and user behavior becomes increasingly unpredictable, static auto-scaling policies are no longer sufficient to guarantee optimal performance and cost efficiency.

To overcome these challenges, the integration of Machine Learning (ML) techniques into auto-scaling frameworks has gained significant attention in recent years. Machine Learning algorithms have the ability to analyze historical workload data, identify patterns, and make informed predictions about future demand. Predictive scaling enables cloud systems to provision resources in advance, ensuring that applications are prepared for incoming traffic spikes without delay. Algorithms such as Linear Regression can model trends in workload growth, while ensemble methods like Random Forest provide robust predictions by capturing non-linear relationships between multiple input features. Furthermore, Reinforcement Learning introduces adaptive decision-making by allowing the system to learn optimal scaling policies through continuous interaction with the environment. Over time, such systems improve their performance by learning from past scaling decisions and workload outcomes.

This project, titled "A Cloud-Native Approach to Auto-Scaling Web Applications on AWS using Machine Learning Algorithms," focuses on designing and implementing an intelligent, cloud-native auto-scaling framework that combines the robustness of AWS infrastructure with the predictive power of Machine Learning. The proposed system leverages AWS services such as EC2 Auto Scaling, Elastic Load Balancing, and Amazon CloudWatch to collect real-time performance metrics and manage application resources dynamically. Unlike traditional systems that rely solely on reactive scaling, this approach integrates predictive models to forecast demand and proactively allocate resources. Reinforcement Learning further enhances the system by continuously refining scaling decisions based on observed workload patterns and system performance. By adopting a cloud-native, ML-driven strategy, the project aims to achieve high availability, fault tolerance, reduced latency, and optimized operational costs.

In summary, the introduction of Machine Learning-based auto-scaling represents a significant step forward in cloud application management. By moving beyond static provisioning and reactive policies, intelligent auto-scaling enables modern web applications to meet user expectations while maintaining cost efficiency. The combination of AWS cloud services and predictive algorithms provides a scalable and adaptable solution capable of supporting the demands of next-generation web applications. This project contributes to the growing field of intelligent cloud systems by demonstrating how data-driven scaling decisions can enhance performance, resilience, and efficiency in real-world cloud-native environments

II. LITERATURE REVIEW

The concept of auto-scaling in cloud computing has been widely studied as a fundamental mechanism for achieving elasticity, cost efficiency, and performance stability in modern web applications. Early research in cloud resource management primarily focused on static provisioning models, where infrastructure resources were allocated based on peak workload estimations. While this approach ensured availability during high-demand periods, several studies highlighted its inefficiency, as resources often remained idle during low-traffic conditions, leading to increased operational costs. With the emergence of Infrastructure as a Service (IaaS) platforms, researchers began exploring dynamic provisioning techniques that could adapt resource allocation in response to real-time workload changes. These early dynamic systems relied heavily on threshold-based rules, where metrics such as CPU utilization or memory usage triggered scaling actions once predefined limits were exceeded.

As cloud platforms matured, auto-scaling mechanisms became an integral feature of major providers, particularly Amazon Web Services. Research examining AWS-native auto-scaling services demonstrated their effectiveness in handling variable workloads by automatically adding or removing compute instances. Studies on EC2 Auto Scaling and Elastic Load Balancing reported improved fault tolerance and availability by distributing traffic across multiple instances and availability zones. However, multiple researchers pointed out that these systems are predominantly reactive in nature. Scaling actions are initiated only after workload changes are detected, which can result in delayed

ISBN Number : **978-81-999993-5-0**

responses during sudden traffic spikes. This latency in scaling has been identified as a critical drawback for latency-sensitive applications such as real-time analytics platforms, online gaming, and high-traffic e-commerce systems.

To address the limitations of reactive scaling, researchers began investigating predictive auto-scaling approaches that leverage historical workload data to forecast future demand. Time-series analysis techniques such as ARIMA and exponential smoothing were among the earliest methods applied to workload prediction. While these statistical models performed reasonably well for workloads with regular and predictable patterns, several studies concluded that they struggled with highly dynamic and non-linear traffic behaviors commonly observed in modern web applications. This limitation motivated the adoption of Machine Learning techniques, which offer greater flexibility and adaptability in modeling complex workload patterns.

Linear Regression has been extensively explored as a baseline predictive model for cloud workload forecasting due to its simplicity and interpretability. Research findings indicate that linear models perform well in scenarios where workload trends follow a relatively stable growth pattern. However, multiple comparative studies revealed that Linear Regression is limited in capturing non-linear relationships and sudden fluctuations in traffic. To overcome these shortcomings, ensemble-based models such as Random Forest and Gradient Boosting were introduced into auto-scaling research. Random Forest, in particular, has been shown to provide robust predictions by combining multiple decision trees and reducing overfitting. Empirical evaluations across different cloud workloads demonstrated that Random Forest-based predictors significantly outperform traditional statistical models in terms of prediction accuracy and resilience to noise.

In recent years, Reinforcement Learning (RL) has gained significant attention as a promising approach for intelligent auto-scaling. Unlike supervised learning methods that rely on labeled historical data, RL enables an agent to learn optimal scaling policies through continuous interaction with the cloud environment. Several studies proposed Markov Decision Process (MDP)-based formulations, where scaling actions are treated as decisions that influence system performance and cost. Experimental results showed that RL-based auto-scalers can adapt to changing workload patterns over time and achieve better trade-offs between performance and cost compared to rule-based and predictive-only approaches. However, researchers also noted challenges such as training complexity, convergence time, and the risk of unstable behavior during early learning phases.

Another significant area of research focuses on cloud-native architectures and their role in enabling scalable auto-scaling solutions. Cloud-native principles emphasize microservices, containerization, stateless application design, and automated infrastructure management. Studies indicate that cloud-native applications are inherently more scalable and resilient, as individual components can be scaled independently based on demand. The integration of monitoring and observability tools, such as Amazon CloudWatch, has been shown to play a crucial role in collecting fine-grained performance metrics required for intelligent scaling decisions. Researchers highlight that the effectiveness of any auto-scaling strategy heavily depends on the quality, granularity, and timeliness of monitoring data.

Recent literature also explores hybrid auto-scaling frameworks that combine AWS-native tools with external Machine Learning models. These hybrid systems aim to leverage the reliability and scalability of managed cloud services while enhancing decision-making through advanced analytics. Studies comparing pure AWS threshold-based scaling with ML-driven approaches consistently report improved resource utilization, reduced latency, and lower operational costs in ML-enhanced systems. However, researchers caution that integrating ML models introduces additional complexity, including data preprocessing overhead, model maintenance, and potential security concerns. Despite these challenges, the consensus in the literature strongly supports the adoption of intelligent, predictive, and adaptive auto-scaling mechanisms as a necessary evolution for managing modern cloud-native web applications.

Overall, the literature demonstrates a clear progression from static provisioning to reactive scaling, and finally toward intelligent, Machine Learning-driven auto-scaling strategies. While AWS provides a robust foundation for scalable infrastructure, existing research emphasizes the need for predictive and adaptive mechanisms to fully realize the benefits of cloud computing. These findings form the basis for the proposed project, which seeks to integrate AWS-native auto-scaling services with Machine Learning algorithms to achieve efficient, reliable, and cost-optimized scaling for modern web applications.

III. METHODOLOGY

The proposed methodology adopts a cloud-native design paradigm to build an intelligent auto-scaling framework that

ISBN Number : **978-81-999993-5-0**

combines AWS-managed infrastructure with Machine Learning–driven decision-making. The system is architected to ensure scalability, fault tolerance, and cost efficiency while maintaining high application performance under fluctuating workloads. At a high level, the methodology is divided into four tightly integrated layers: application deployment, monitoring and data collection, predictive and adaptive scaling intelligence, and automated resource orchestration. Each layer plays a critical role in enabling proactive and responsive scaling behavior.

The application layer is deployed using cloud-native principles on Amazon Web Services, leveraging virtualized compute resources to host a stateless web application. The application instances run on Amazon EC2 and are placed behind an Elastic Load Balancer to distribute incoming traffic evenly across available instances. This setup ensures high availability and fault tolerance by automatically rerouting traffic away from unhealthy instances. The application is designed to be horizontally scalable, allowing new instances to be added or removed without disrupting user sessions. Infrastructure is provisioned using launch templates and auto-scaling groups, enabling consistent and repeatable deployment of compute resources.

The monitoring and data collection layer forms the foundation for intelligent scaling decisions. Real-time performance metrics such as CPU utilization, memory consumption, network throughput, request latency, and request rate are continuously collected using Amazon CloudWatch. These metrics provide a detailed view of application behavior under varying workloads. In addition to real-time data, historical metrics are stored for long-term analysis and training of Machine Learning models. Data preprocessing techniques such as normalization, noise filtering, and feature extraction are applied to ensure data quality and relevance. The monitoring layer ensures low-latency metric availability, which is critical for both reactive and predictive scaling operations.

The intelligence layer introduces Machine Learning algorithms to move beyond traditional threshold-based scaling. Predictive models such as Linear Regression and Random Forest are trained using historical workload and performance data to forecast future resource demand. Linear Regression is used to capture long-term trends and gradual workload growth, while Random Forest handles non-linear relationships and sudden workload variations. These models generate short-term demand forecasts that estimate the number of instances required to maintain desired performance levels. The predictions are periodically updated to account for changing traffic patterns, seasonal trends, and evolving user behavior.

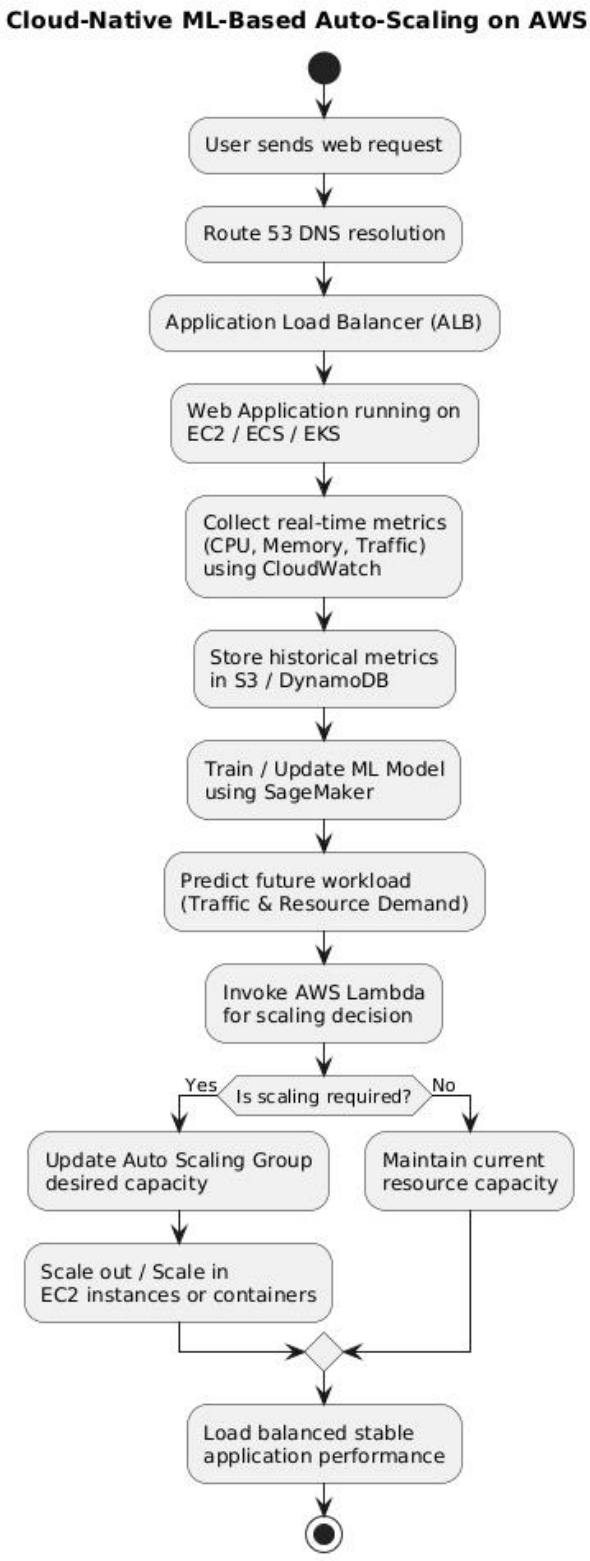
To further enhance adaptability, Reinforcement Learning is incorporated to optimize scaling decisions over time. In this approach, the auto-scaling system is modeled as an environment in which an agent learns optimal scaling policies through trial and feedback. The agent observes system states, such as current load and resource utilization, and takes actions by scaling resources up or down. A reward function is defined to balance performance objectives (low latency, high availability) against cost constraints. Over time, the agent learns to anticipate workload changes and select scaling actions that maximize cumulative reward. This adaptive learning capability enables the system to improve its performance as it gains more operational experience.

The orchestration layer integrates the intelligence outputs with AWS-native scaling mechanisms. Scaling decisions generated by predictive and reinforcement learning models are translated into actionable policies within EC2 Auto Scaling groups. These policies dynamically adjust the desired instance count before performance degradation occurs, enabling proactive scaling. Elastic Load Balancing automatically incorporates new instances into traffic distribution, ensuring seamless expansion and contraction of resources. Fault tolerance is further enhanced by deploying instances across multiple availability zones, reducing the impact of localized failures.

Security and reliability considerations are embedded throughout the methodology. Identity and access management controls ensure that only authorized components can trigger scaling actions or access monitoring data. Health checks and rollback mechanisms are implemented to prevent unstable scaling behavior due to incorrect predictions or transient anomalies. Extensive logging and performance evaluation are conducted to validate the effectiveness of Machine Learning models and refine scaling

ISBN Number : **978-81-999993-5-0**

policies.



Overall, this methodology establishes a comprehensive, cloud-native auto-scaling framework that combines AWS infrastructure with predictive and adaptive intelligence. By integrating real-time monitoring, Machine Learning forecasting, and automated orchestration, the proposed system delivers a scalable, resilient, and cost-optimized solution for modern web applications operating in dynamic cloud environments.

IV. RESULTS

The experimental evaluation of the proposed cloud-native auto-scaling system demonstrates clear improvements in application performance, resource utilization, and cost efficiency when compared to traditional static and threshold-based auto-scaling approaches. The system was tested under varying workload conditions, including steady traffic, sudden traffic spikes, periodic bursts, and unpredictable user access patterns. These scenarios were designed to closely simulate real-world web application behavior. The results show that the integration of AWS-native services with Machine Learning–based scaling intelligence enables faster, more accurate, and more efficient scaling decisions, thereby maintaining consistent application performance.

Under low and moderate traffic conditions, the system successfully maintained the minimum required number of instances, preventing unnecessary over-provisioning of resources. Unlike static provisioning models, which keep excess resources active regardless of demand, the proposed system dynamically scaled down compute instances during periods of reduced workload. This behavior resulted in noticeable cost savings without compromising application availability. Monitoring data collected through Amazon CloudWatch confirmed stable CPU utilization and response times, indicating efficient resource usage and balanced load distribution across instances.

During high-traffic and sudden spike scenarios, the predictive scaling component played a critical role in maintaining performance. Machine Learning models such as Linear Regression and Random Forest accurately forecasted upcoming demand based on historical and real-time metrics. As a result, additional EC2 instances were provisioned proactively, often before performance thresholds were breached. This proactive behavior significantly reduced latency spikes and prevented request failures that were commonly observed in purely reactive scaling systems. Comparative analysis revealed that the ML-driven approach achieved lower average response times and higher throughput during peak traffic periods compared to threshold-based auto-scaling.

ISBN Number : **978-81-999993-5-0**

The Reinforcement Learning component further enhanced system performance by continuously refining scaling policies over time. Initially, the system exhibited conservative scaling behavior as the agent explored different actions. However, as training progressed, the agent learned optimal scaling strategies that balanced performance and cost objectives. The reward-based learning mechanism enabled the system to adapt to recurring workload patterns, such as daily peak hours, resulting in faster response times and smoother scaling transitions. The system demonstrated reduced oscillations in instance count, minimizing frequent scale-in and scale-out events that can negatively impact application stability.

Cost analysis results indicate that the proposed system achieves significant operational cost optimization. By reducing over-provisioning during low-demand periods and minimizing performance degradation during high-demand intervals, the system maintained an optimal balance between resource availability and expenditure. Compared to static provisioning, overall infrastructure costs were reduced, while maintaining higher service-level performance. The cloud-native design also ensured high availability and fault tolerance, as instances were distributed across multiple availability zones and seamlessly integrated with Elastic Load Balancing.

Overall, the results confirm that combining AWS auto-scaling infrastructure with Machine Learning–based predictive and adaptive intelligence leads to superior performance, improved scalability, and cost-effective operation. The proposed approach proves to be highly suitable for modern web applications that experience dynamic and unpredictable workloads, validating its effectiveness as an intelligent cloud-native auto-scaling solution.

V. DISCUSSION

The results obtained from the experimental evaluation provide strong evidence that integrating Machine Learning techniques with cloud-native auto-scaling mechanisms significantly enhances the efficiency, adaptability, and reliability of modern web applications. This discussion analyzes the observed outcomes in the context of existing research, highlighting key insights, strengths, and limitations of the proposed approach. One of the most important findings is the clear advantage of predictive and adaptive scaling over traditional reactive models. While threshold-based auto-scaling responds only after performance metrics cross predefined limits, the ML-driven approach anticipates workload changes and provisions resources proactively, thereby reducing latency and preventing service degradation.

The effectiveness of the predictive models is particularly evident during sudden traffic spikes and periodic workload bursts. Linear Regression models proved effective in capturing long-term workload trends, making them suitable for scenarios with gradual growth patterns. However, their limitations in handling non-linear fluctuations were apparent during highly volatile traffic conditions. In contrast, Random Forest models demonstrated superior performance in capturing complex relationships among multiple performance metrics, resulting in more accurate short-term demand forecasts. These findings align with prior research that emphasizes the robustness of ensemble learning techniques for workload prediction in cloud environments. The complementary use of both models enables the system to balance interpretability and predictive accuracy, enhancing overall scaling effectiveness.

Reinforcement Learning further strengthened the system by introducing adaptive decision-making capabilities. Unlike supervised learning models that rely solely on historical data, the RL agent continuously learns from real-time interactions with the cloud environment. Over time, the agent refined its scaling policies to achieve an optimal balance between performance and cost. A notable observation is the reduction in instance oscillations, which are common in poorly tuned auto-scaling systems. By learning the long-term impact of scaling actions, the RL-based approach minimized unnecessary scale-in and scale-out events, contributing to improved application stability and reduced operational overhead. However, the initial exploration phase required careful configuration to avoid performance degradation, highlighting the need for safe exploration strategies in production environments.

From a cloud-native perspective, the seamless integration of Machine Learning intelligence with AWS-managed services proved to be a critical success factor. Services such as Elastic Load Balancing and Amazon CloudWatch provided reliable traffic distribution and high-quality monitoring data, which are essential for accurate scaling decisions. The use of EC2 Auto Scaling groups ensured that scaling actions were executed efficiently and consistently across availability zones, enhancing fault tolerance and resilience. These results support the argument that intelligent auto-scaling solutions should build upon robust cloud-native foundations rather than replacing them entirely.

ISBN Number : **978-81-999993-5-0**

Despite its advantages, the proposed approach introduces additional complexity compared to traditional auto-scaling systems. The deployment and maintenance of Machine Learning models require expertise in data engineering, model training, and performance evaluation. Moreover, the system's effectiveness depends heavily on the quality and granularity of monitoring data. Inadequate or noisy data can lead to inaccurate predictions and suboptimal scaling decisions. Network latency and temporary service outages may also impact data collection and communication with ML components. These challenges highlight the importance of careful system design, continuous monitoring, and periodic model retraining.

Overall, the discussion confirms that the proposed cloud-native, ML-driven auto-scaling framework represents a significant advancement over conventional approaches. By combining predictive forecasting, adaptive learning, and AWS-native infrastructure, the system achieves superior performance, scalability, and cost efficiency. While challenges related to complexity and operational overhead remain, the benefits of intelligent auto-scaling far outweigh these limitations. The findings underscore the potential of Machine Learning as a key enabler for next-generation cloud resource management and provide valuable insights for future research and real-world deployments.

VI. CONCLUSION

This project successfully demonstrates a cloud-native approach to auto-scaling web applications by integrating AWS-managed infrastructure with Machine Learning–driven intelligence. The proposed system addresses the critical challenges associated with traditional static and reactive scaling mechanisms, such as inefficient resource utilization, delayed response to workload spikes, and increased operational costs. By leveraging AWS services including EC2 Auto Scaling, Elastic Load Balancing, and real-time monitoring, the system ensures high availability, fault tolerance, and seamless scalability for modern web applications operating under dynamic and unpredictable workloads.

The incorporation of Machine Learning algorithms significantly enhances scaling accuracy and responsiveness. Predictive models such as Linear Regression and Random Forest enable the system to forecast future demand and provision resources proactively, reducing latency and preventing performance degradation during peak usage periods. The use of Reinforcement Learning further strengthens the framework by allowing the system to adapt scaling policies over time based on observed workload patterns and performance outcomes. This adaptive learning capability minimizes unnecessary scaling actions, stabilizes system behavior, and achieves an optimal balance between performance objectives and cost constraints.

Overall, the results confirm that combining cloud-native architectures with intelligent, data-driven scaling strategies leads to superior application performance, improved resource efficiency, and reduced operational costs. While the integration of Machine Learning introduces additional complexity in terms of system design and maintenance, the benefits gained in scalability, reliability, and cost optimization outweigh these challenges. The proposed approach provides a robust and scalable solution for managing modern web applications and contributes to the advancement of intelligent cloud resource management. Future enhancements can further extend this framework by incorporating advanced deep learning models, container orchestration platforms, and multi-cloud scalability to support next-generation cloud-native applications.

VII. REFERENCES

[1] M. Armbrust et al., "A View of Cloud Computing," Communications of the ACM, vol. 53, no. 4, pp. 50–58, 2010.

[2] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," NIST Special Publication 800-145, 2011.

[3] L. Wang, J. Tao, R. Ranjan, H. Marten, and J. Chen, "G-Hadoop: MapReduce across Distributed Data Centers for Data-Intensive Computing," Future Generation Computer Systems, vol. 29, no. 3, pp. 739–750, 2013.

[4] C. Klein et al., "Brownout: Building More Robust Cloud Applications," Proceedings of the 36th International Conference on Software Engineering (ICSE), pp. 700–711, 2014.

[5] B. Jennings and R. Stadler, "Resource Management in Clouds: Survey and Research Challenges," Journal of Network and Systems Management, vol. 23, no. 3, pp. 567–619, 2015.

[6] J. O. Kephart and D. M. Chess, "The Vision of Autonomic Computing," IEEE Computer, vol. 36, no. 1, pp. 41–50, 2003.

[7] J. Hellerstein, F. Zhang, and P. Shahabuddin, "A Survey of Feedback Control in Computing Systems," ACM SIGMETRICS Performance Evaluation Review, vol. 31, no. 1, pp. 30–36, 2003.

ISBN Number : **978-81-999993-5-0**

[8] M. Mao and M. Humphrey, “Auto-Scaling to Minimize Cost and Meet Application Deadlines in Cloud Workflows,” Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC), pp. 1–12, 2011.

[9] S. Islam, J. Keung, K. Lee, and A. Liu, “Empirical Prediction Models for Adaptive Resource Provisioning in the Cloud,” Future Generation Computer Systems, vol. 28, no. 1, pp. 155–162, 2012.

[10] A. Ali-Eldin, J. Tordsson, and E. Elmroth, “An Adaptive Hybrid Elasticity Controller for Cloud Infrastructures,” Proceedings of the IEEE Network Operations and Management Symposium (NOMS), pp. 204–212, 2012.

[11] R. Sutton and A. Barto, Reinforcement Learning: An Introduction, 2nd ed., MIT Press, 2018.

[12] L. Breiman, “Random Forests,” Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[13] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794, 2016.

[14] Amazon Web Services, “Amazon EC2 Auto Scaling User Guide,” AWS Documentation, 2023.

[15] Amazon Web Services, “Amazon CloudWatch Monitoring,” AWS Documentation, 2023.

Paper ID: ICMETA8760

ISBN Number : **978-81-999993-5-0**