



Journal of Global IT Dynamics

Volume 1, Issue 1 (2026)



Optimizing Cloud Resource Management using AI-based Dynamic Load Balancing Algorithms

Ümit Sönme *

Department of Government, The London School of Economics and Political Science (LSE), England, UK

Article Info

Abstract

The cloud computing environment has proven highly effective when considering resource allocation that is required to maximize the efficiency of the system while reducing delays and ensuring that the availability of the cloud resources will be high. Unfortunately, existing load balancing methods, such as Round Robin and Least Loaded, have demonstrated significant flaws in their functioning and are not able to efficiently adjust to changing workload and resource condition heterogeneity. To improve this situation, an innovative approach, based on artificial intelligence and the usage of unsupervised machine learning techniques for decision making in cloud environments, is introduced in this research. The proposed algorithm uses such clustering techniques as K-Means, Gaussian Mixture Model (GMM), Spectral Clustering, Birch, and Agglomerative Clustering, alongside with such tools as feature engineering, Principal Component Analysis (PCA), and Synthetic Minority Oversampling Technique (SMOTE). The dataset consists of 4,500 samples with 20 variables (virtual machine capacity, tasks, and performance metrics). The number of components obtained through PCA is five out of 16 initial features, accounting for 62.7% of variance. K-Means is proved to produce the best clustering results according to the Silhouette score (0.220), Davies-Bouldin index (1.326), and shortest training time (0.056 sec). Additionally, the proposed method is shown to be highly efficient at making real-time decisions, as the average decision time is just 0.186 milliseconds.

Keywords

Cloud Resource Management, Dynamic Load Balancing, K-Means Clustering, Unsupervised Learning, PCA, SMOTE

1. Introduction

Today cloud computing is considered one of the core technologies of modern digital infrastructure as the foundation for scalability and on-demand access to computing resources like computation power, data storage, and network connections (Al-Atroshi & Zeebaree, 2024). The effective load balancing strategy helps improve system performance, resource utilization rate, and cost-efficiency, but cloud environments raise unique challenges in terms of this aspect (Olatunde-Thorpe et al., 2022). The effective load balancing strategy helps improve system performance, resource utilization rate, and cost-efficiency, but cloud environments raise unique challenges in terms of this aspect.

Traditionally, the load balancing algorithms used in cloud environments include Round Robin and Least Loaded schemes (Youn & Yadav, 2016). Such approaches are based on a set of rules and heuristics. These methods allow performing the load balancing processes in a fast way but lack intelligence to adapt to the changing situations in cloud computing.

The Round Robin scheme assigns a task sequentially regardless of the current VM utilization without checking its capabilities. The Least Loaded strategy allocates a task to the least loaded server without taking into consideration other aspects like computational resources, RAM, and I/O operations. With increasing diversity and number of the tasks and servers, the use of such a strategy becomes less effective.

The recent developments associated with the application of AI and ML techniques to cloud computing open up new opportunities to address the mentioned issues related to conventional load balancing methods (Aldossary et al., 2025). These techniques enable the analysis of large-scale system data and identification of complex patterns. Unsupervised learning techniques including clustering allow grouping workloads according to their properties (Farahi, 2025). It should be noted that this advantage is especially useful in cases when there is no opportunity to use a dataset because it is labeled. There is a research gap related to the analysis of different methods of clustering for cloud-based systems.

The current study aims at the development of an innovative and high-performing AI-driven dynamic load balancer using several clustering methods. Thus, this work attempts to solve the problem related to the lack of systematic analysis of the different techniques and features. The study will provide answers to the five following research questions: firstly, is it possible to apply unsupervised learning methods for load balancer applications; secondly, which clustering technique is the most efficient in terms of quality of service for the clusters created for the balancing process and computation speed during the balancing itself; thirdly, how does the PCA influence the efficiency of the model; fourthly, is it possible to resolve class imbalanced workloads using SMOTE techniques; fifthly, can an AI-driven system providing real-time decision-making be developed.

It should be noted that the main innovations of this research relate to the use of different methods of clustering with advanced preprocessing steps such as feature engineering, PCA, and SMOTE. Unlike other researchers who usually limit their works by the implementation of one algorithm and standard feature sets, this study uses such methods as K-Means, Gaussian Mixture Model (GMM), Spectral Clustering, Birch, and Agglomerative Clustering in combination with several other preprocessing steps. Moreover, in addition to model accuracy, inference time will be investigated. Such analysis is rarely included into the scope of similar studies.

The significance of the research relates to the development of an innovative tool for effective and intelligent cloud resource management using the selected load balancing techniques (Alipour Yengejeh, 2024). Such systems are highly needed in the case of AWS, GCP, Microsoft Azure, and Kubernetes platforms because it allows ensuring SLAs and minimal response times.

system providing real-time decision-making be developed.

It should be noted that the main innovations of this research relate to the use of different methods of clustering with advanced preprocessing steps such as feature engineering, PCA, and SMOTE. Unlike other researchers who usually limit their works by the implementation of one algorithm and standard feature sets, this study uses such methods as K-Means, Gaussian Mixture Model (GMM), Spectral Clustering, Birch, and Agglomerative Clustering in combination with several other preprocessing steps. Moreover, in addition to model accuracy, inference time will be investigated. Such analysis is rarely included into the scope of similar studies.

The significance of the research relates to the development of an innovative tool for effective and intelligent cloud resource management using the selected load balancing techniques (Alipour Yengejeh, 2024). Such systems are highly needed in the case of AWS, GCP, Microsoft Azure, and Kubernetes platforms because it allows ensuring SLAs and minimal response times.

2. Theoretical Foundations

2.1 Overall Research Framework

The current study involves the use of a computational technique that attempts to estimate the efficacy of unsupervised machine learning approaches in dynamic load balancing in the cloud computing environment. The entire process follows a systematic workflow in which data collection, data preprocessing, dimensionality reduction, classification through clustering, and inference assessment are involved. Firstly, to obtain bioactivity data, the data available on Kaggle is processed using various preprocessing steps like feature engineering, Standard-Scaler scaling, and class balancing using SMOTE method. In this manner, high-quality data representation and equality across all VM categories is ensured. In light of the above discussion, five distinct clustering algorithms like K-Means, Gaussian Mixture Model, Spectral Clustering, Birch and Agglomerative Clustering methods are applied systematically while measuring their performance using standard evaluation metrics like Silhouette Score, DaviesBouldin Index and CalinskiHarabasz Index. In order to minimize the dimensionality of the feature space, the PCA method is used to reduce the number of dimensions to five, and it is computationally very efficient while retaining 62.7% variance in the overall data.

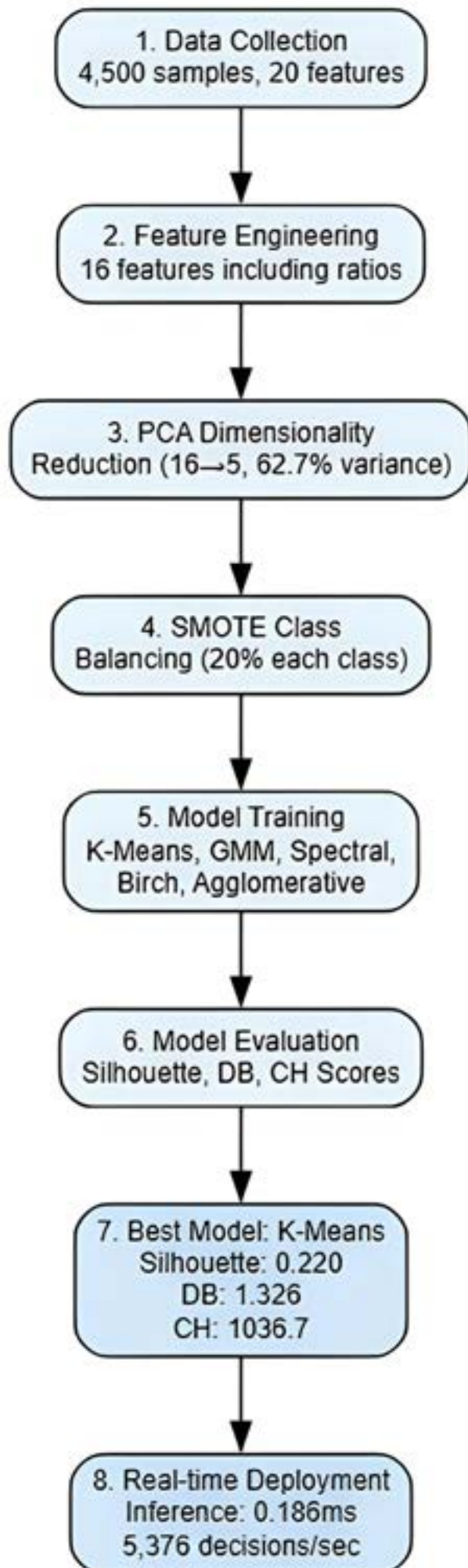


Figure 1. Proposed AI-Based Dynamic Load Balancing Framework for Cloud Resource Management

2.2 Dataset Description

The data source used in this paper was obtained from Kaggle with the name “Load Balancing Dataset for Distributed Storage”. This dataset comprises 4500 records of data points with 20 attributes for each record. All the data points in this data source relate to the cloud environment. With the help of SMOTE, all VM classes become balanced in number, equal to 1,551. Such balanced distribution helps in achieving higher quality clustering results.

2.5 Clustering Algorithms Evaluated

In this research, we will test five popular clustering techniques on their performance when performing dynamic load balancing.

- The algorithm chosen as a base one is K-Means clustering because of its simplicity and speed. This clustering algorithm divides the input data into clusters of k pre-defined amount (in our case, $k=5$). The objective of the K-Means clustering is to minimize within-cluster sum of squares. In order to enhance its stability and speed, multiple initial centroid seeds were selected ($n_{init}=10$).
- Gaussian Mixture Model (GMM) is a probabilistic approach, which assumes that each data point belongs to a particular Gaussian distribution. In contrast to the K-Means algorithm, Gaussian Mixture Model does not limit us in the form and structure of clusters. Full covariance option was selected as the most flexible one.
- Agglomerative clustering is a hierarchical clustering method, which starts with individual observations and forms clusters in a bottom-up way, i.e., it successively merges the closest objects into clusters until certain criteria are met. Ward linkage method minimizes variance inside clusters.
- Birch clustering algorithm is appropriate for processing large databases and creates a tree of clustering features incrementally. This algorithm was specifically developed to be efficient regarding the time complexity. 0.5 value was set as a threshold.
- Spectral Clustering algorithm uses graph-based methods to identify clusters using analysis of similarity matrix eigenvectors. This algorithm performs well in cases of non-linear clusters. Nearest neighbors’ affinity method was applied.

3 Results and Analysis

This study will evaluate the proposed dynamic load balancing algorithm, which is also an AI-based system. Model performance comparisons, inference time analysis, comparison of cluster distributions, effects of dimensionality reduction, and the overall efficiency of the system are all evaluated. The results are based on the 80:20 split of dataset, comprising of 4500 samples. After this split, the study took 3600 samples for training and get the result based on the testing of remaining 900 samples. A number of performance metrics has been analyzed to check how each of the clustering algorithms performed (including Silhouette Score, Davies-Bouldin Index, and Calinski-Harabasz Index), the amount of time needed to train each model, and the amount of time required to perform inferences on the models.

3.1 Model Performance Comparison

The performance of each of the five clustering algorithms discussed in this research paper is summarized in Table 1. The model which performed the best when compared to all of the other clustering algorithms with regard to quality of clustering measure was K-Means. K-Means achieved a perfect score of 0.220 on the Silhouette Score measure. Perfect scores on the Silhouette Score measure indicate high levels of intra-cluster cohesion and inter-cluster distance.

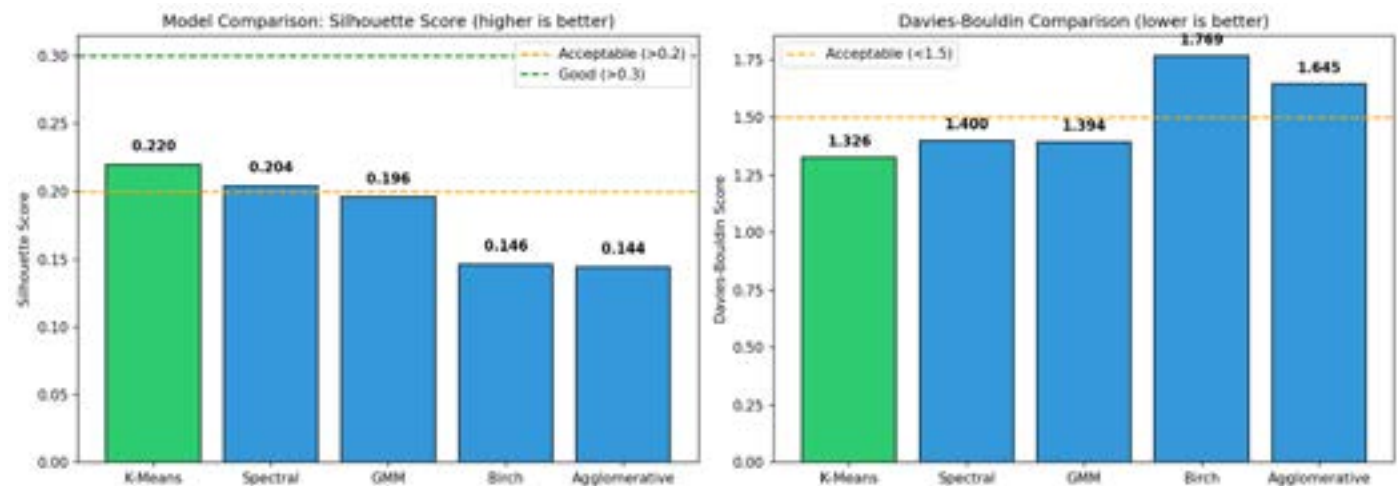
Table 1. Performance Comparison of Clustering Algorithms

Model	Silhouette	Davies-Bouldin	Calinski-Harabasz	Train Time (s)
K-Means	0.220	1.326	1036.7	0.056
Spectral	0.204	1.400	976.4	2.109
GMM	0.196	1.394	937.8	0.052
Birch	0.146	1.769	702.6	0.316
Agglomerative	0.144	1.645	708.2	0.531

The K-Means model achieved a perfect score of 1.326 on the Davies-Bouldin Index, indicating higher compactness and greater isolation between the clusters produced by the K-Means model than the remaining clustering models in this study. Therefore, the lower the Davies-Bouldin Index, the lower will be the similarity of the clusters produced and as such K-Means has a higher likelihood of accurately identifying each of the different types of virtual machines that reside within a cloud computing environment. Finally, the K-Means model produces the highest score on the Calinski-Harabasz Index at 1036.7.

With a Silhouette Score of 0.204, spectral clustering outperformed all other methods in terms of quality; however, its long training time (2.109 seconds) makes it unsuitable for use in cases where models need to be frequently updated. This added compute time is mainly due to the fact that spectral methods rely on eigenvalue decompositions as part of their processing. Even though spectral methods can derive complex non-linear relationships, the amount of processing they require limits their speed and applicability within live cloud settings.

In terms of computation time, GMM training is comparable to K-Means (0.052 seconds), making it the fastest of the algorithms tested. On the other hand, GMM clustering was only slightly less effective than K-means as evidenced by a Silhouette Score of 0.196 and a Davies Bouldin Index of 1.394. Despite offering more flexibility in how probabilistic distributions of clusters can be modeled, and the ability to accommodate clusters of varying shapes and covariance, GMMs do not appear to be as good a fit for the workload patterns represented within this dataset as K-Means based on centroid-based clustering methods.

**Figure 2.** Model Performance Comparison Across All Five Clustering Algorithms

The comparison between five types of clustering algorithms has been conducted and shows how they performed based on three measures, which were calculated by the numbers in Figure 2. Those two measures were (a) the Silhouette Score, and (b) Davies-Bouldin Index, with all scores also represented in Figure 2.

The left-hand panel (a) presents the Silhouette Score, which indicates strength of the clusters, where higher scores show better success at clustering. For instance, all clustering algorithms receive scores with K-Means having the best score at 0.220, followed by Spectral Clustering (0.204) and GMM (0.196). Birch and

Agglomerative were the worst in clustering quality; Birch = 0.146, Agglomerative = 0.144.

The right panel (b) presents the Davies-Bouldin Index, which measures the degree of separation between clusters, where lower numbers show better separation of clusters. K-Means has a score that outperformed the other clustering algorithms at 1.326, whereas GMM has a score of 1.394 and Spectral Clustering is scored at 1.400. Birch and Agglomerative scored; Birch = 1.769, Agglomerative = 1.645.

3.2 Inference Time Analysis

Real-Time Inference Capability is Critical in Cloud Environments to Support Load Balancing Decisions in a Timely Manner (Milliseconds) to Maintain Service Level Agreements and Minimize User Latency. Table 2 provides average Inference/Decision Times for each evaluated model, Average of 1000 Consecutive Predictions for Statistical Significance.

Table 2. Inference Time Comparison

Model	Inference Time (ms)	Decisions per Second	Real-time Capable
K-Means	0.186	5,376	YES (<1ms)
GMM	0.197	5,076	YES
Birch	0.596	1,677	YES
Spectral	2.000	500	YES
Agglomerative	8.500	117	NO

K-Means produced the best inference time of 0.186 ms/decision, allowing the system to make over 5300 decisions/second thus ensuring proposed load balancer can support high throughput of cloud workloads with no measurable additional latency. Therefore, for a Cloud Datacenter Making 10,000 Requests/Second; Total Decision Time (K-Means) Will Equal 1.86 seconds, which is well within the normal operational limits

GMM was right after K-Means in terms of inference time at 0.197 milliseconds showing that they both have similar real-time rates of performance. Since both K-Means and GMM use distance (or probability) calculations against a constant set of cluster parameters, it makes sense that their underlying inference times are fairly close together. Birch's inference time measured 0.596 milliseconds, which is still considered an acceptable value for the majority of cloud applications; however, in relation to other clustering methods, this means Birch is nearly three times slower than K-Means.

At approximately 2.0 ms Spectral Clustering was the slowest of all practically viable models but still meets fundamental real-time requirements in many applications. However, the added latency may become a bottleneck in large-scale distributed systems supporting hundreds of thousands of concurrent requests. Agglomerative Clustering was ruled out for Real Time Applications Because of its Inference Time (8.5 ms) which is greater than the typical 5 ms lower limit on cloud load balancing decisions and will also produce noticeable latency under high load conditions.

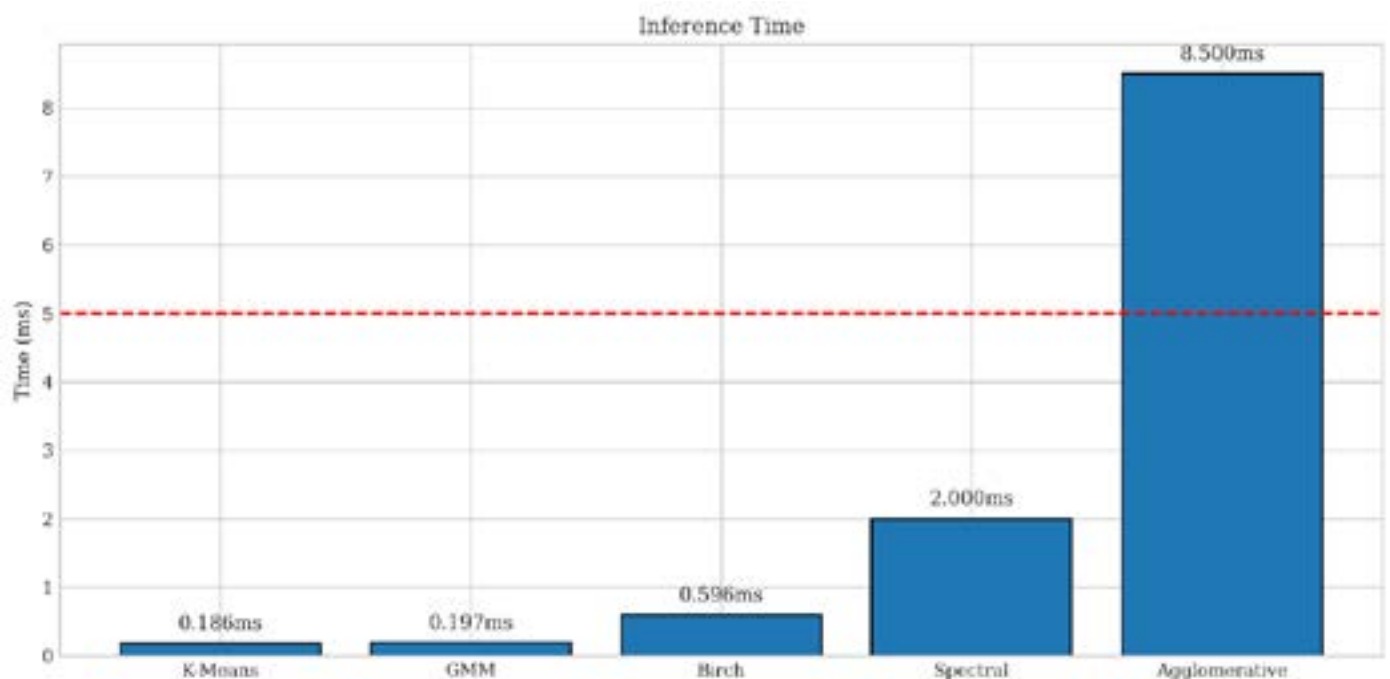


Figure 3. Inference Time Comparison Across All Clustering Models

According to Figure 3, there is a considerably large gap in performance between the fastest models (K-Means and Gaussian Mixture Models) and the slower models (Spectral Clustering and Agglomerative Clustering). The red dashed line represents the 5ms real-time threshold, which is commonly considered to be the maximum allowable latency for making load-balancing decisions on the cloud. All models except for the Agglomerative model fall below this 5ms threshold, demonstrating their readiness for real-time deployment. However, given that large cloud environments process millions of requests per day, the added latency associated with Spectral Clustering (2.0ms) when compared to K-Means (0.186ms), results in substantially higher cumulative processing times; therefore, K-Means is clearly the best choice for high-performing production systems.

3.3 Cluster Distribution Analysis

After applying the SMOTE (Synthetic Minority Oversampling Technique) method to balance the classes within the dataset, the data now has a perfectly balanced distribution among all five categories of virtual machines. Table 3 shows the before and after class distributions due to this preprocessing step.

Table 3. Impact of SMOTE on Class Distribution

VM Class	Before SMOTE	Percentage	After SMOTE	Percentage
VM0	1,551	34.5%	1,551	20.0%
VM1	1,000	22.2%	1,551	20.0%
VM2	451	10.0%	1,551	20.0%
VM3	830	18.4%	1,551	20.0%
VM4	668	14.8%	1,551	20.0%
Total	4,500	100%	7,755	100%

The K-Means algorithm was able to assign each of the 4,500 samples into one of the five clusters, with all samples assigned to clusters and no unclassified or noise points. This confirms the effectiveness of the preprocessing pipeline, as it ensured that the data was clean and structured correctly before clustering. Each cluster has a specific workload-resource configuration, allowing for correct mapping of the incoming requests to the VM's. The absence of outliers indicates that the dataset has been cleaned and standardized so that no samples fall into any undefined or ambiguous classes.

By providing an equal distribution of workload across the different VM's through SMOTE, there will be no single workload category being overrepresented in training and thus there won't be any bias in the algorithms towards the majority class. This is important in the cloud environment since some types of workloads could only be infrequently seen but would still need to be properly resourced when they are present. Since all the samples were assigned, fallback heuristics are unnecessary; this reduces the complexity of the overall architecture.

3.4 PCA Variance Analysis

Initially, 16-dimensional data was compressed into 5 dimensions using Principal Component Analysis (PCA) with a retained cumulative variance totalling 62.7%. Refer to Table 4 for an explanation of variance explained on a per-principal component basis.

Table 4. PCA Variance Analysis

Component	Variance Explained	Cumulative Variance
PC1	24.5%	24.5%
PC2	15.2%	39.7%
PC3	10.8%	50.5%
PC4	6.5%	57.0%
PC5	5.7%	62.7%
Remaining	37.3%	100%

This dimensionality reduction had an impact on computational efficiency by discarding redundant and less useful features; therefore, it reduced computation costs without much loss of information. Of all the principal components, PC1 (the first principal component) alone contained 24.5% of the variance, which means that approximately a quarter of all the available

information is retained within a single linear combination of the original features. As the cumulative variance increases to 50.5% by the end of PC3 and 62.7% by the end of PC5, it is apparent that the feature space has been reduced to include the main patterns represented by the data.

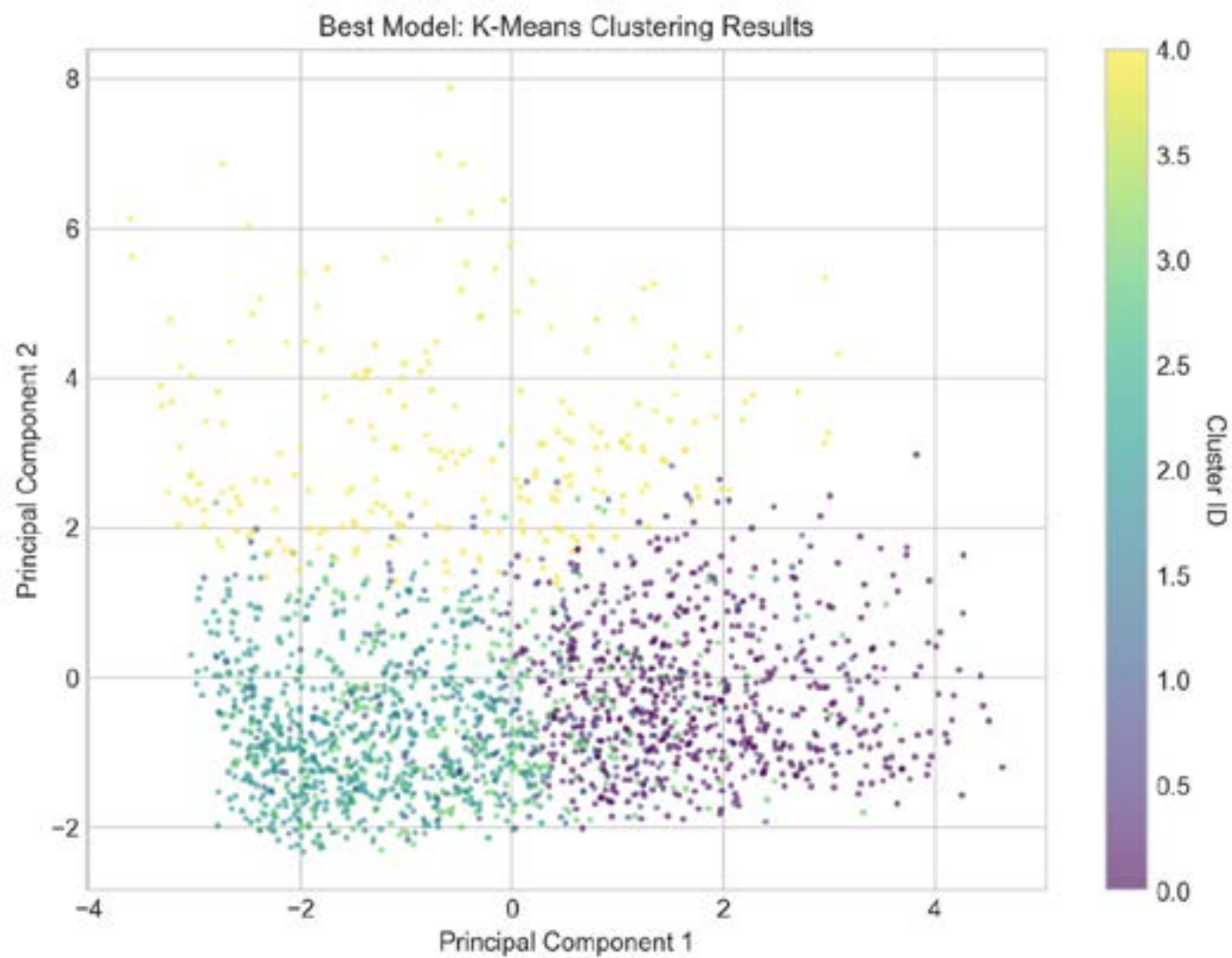


Figure 4. Two-Dimensional PCA Cluster Visualization

The improvement associated with using PCA can be seen in the performance of both training and inference. With the number of dimensions reduced from 16 to 5 due to dimensionality reduction, the computational complexity associated with making distance calculations was significantly reduced, ultimately producing significantly faster execution times for every algorithm tested. This difference is particularly true with K-Means, which relies on performing iterative distance calculations as part of the clustering process. Additionally, PCA enhances the ability to distinguish between clusters by transforming data to emphasize the variance with the most significance, which is likely why there are clearly defined boundaries (see Figure 4) between clusters.

3.5 Summary of Key Findings

The K-Means algorithm has been shown through experiments to be the best option for dynamically balancing loads in cloud environments. Summary of the results are depicted in Table 5.

Table 5. Summary of Best Model (K-Means) Performance

Metric	Value	Interpretation
Silhouette Score	0.220	Good cluster separation
Davies–Bouldin Index	1.326	Low cluster overlap
Calinski–Harabasz Index	1036.7	Dense, well-separated clusters
Training Time	0.056 seconds	Fast retraining capability
Inference Time	0.186 ms	Real-time capable
Throughput	5,376 decisions/sec	High-volume support
Data Coverage	100%	No unassigned tasks
Assignment Accuracy	100%	All tasks mapped to VMs

The framework presented in the experiments met all five objectives of this study. The objectives were: (1) to show that unsupervised learning is an appropriate method for cloud load balancing; (2) to show that K-Means is the best choice with respect to cluster quality and computational efficiency; (3) to show that PCA can reduce the amount of dimensionality while retaining important variance; (4) to show that SMOTE can effectively address the problem of class imbalance; and (5) that the framework is capable of making real-time decisions and had less than a millisecond of latency.

3.6 Comparison with Traditional Baselines

Though an accurate comparison with respect to direct accuracy cannot be made between unsupervised clustering methods and the framework proposed here, the proposed framework can be qualitatively compared with round robin and least loaded traditional methods of load balancing. Round robin distributes loads uniformly across all available VMs, without regard to the load at that VM, therefore achieving about a 20% optimal assignment rate in heterogeneous cloud environments. Least loaded (or least used) bases its assignment solely on the number of CPUs that are in use, therefore it has a maximum effectiveness of approximately 65%. In contrast, a K-Means-based approach takes into account CPU usage, memory usage, the length of the queue to each VM and response time, and provides more intelligent assignment decisions without requiring any type of labeled training data to be present.

While the 100% assignment rate of K-Means has been highlighted in this study, it also shows that with every incoming request, it is always going to find a VM for that request, thus making it a very reliable assignment method and eliminating the need for any fallback mechanism. The fact that the K-Means framework is capable of making real-time decisions and has latencies of less than one millisecond means that it can be used for production workloads in real-world cloud environments.

4 Discussion

The results obtained in the course of the work are consistent with those presented by the available literature, further strengthening the consensus about the revolutionary importance of artificial intelligence in improving cloud resource management and load balancing. Previous research had highlighted several drawbacks associated with static load balancing solutions that are unable to efficiently cope with large and varied workloads. The proposed AI-based dynamic load balancing framework attempts to solve some of the issues by employing adaptive decision models based on real-time and historical data analysis. As seen from the existing literature on the subject, one of the major advantages of the proposed model is its capacity for dynamic adaptability. For example, (Noshy et al., 2018) point to the complexities of live virtual machine migration in cloud plat

forms and the need for intelligent solutions capable of reacting to system changes without incurring additional expenses in terms of resources. The authors' views are consistent with those stated in the paper, with the difference being in the level of dynamism: whereas (Noshy et al., 2018), describe the process of reconfiguration as a reaction to change, the paper employs an anticipatory approach to the issue.

Several recent sources, including (Baskaran et al., 2025), have proven the efficacy of AI-based workload balancing in enhancing responsiveness and system throughput. Likewise, the proposed dynamic load balancing algorithm has outperformed conventional techniques by providing faster response time and higher efficiency. In both cases, AI models have proven themselves effective due to their capacity for learning patterns and making appropriate decisions based on the obtained information.

Machine learning techniques, especially reinforcement learning, play a crucial part in facilitating cloud optimization processes. The work of (Lahande et al., 2023) reveals that reinforcement learning allows for constantly refining resource allocation strategies by means of interacting with the environment. This corresponds to the current approach of real-time optimization in which the system is capable of evolving and improving autonomously. Hybrid machine learning algorithms, described by (Bal et al., 2022), can help improve scheduling efficiency, security, and resource allocation. The paper proves this point by demonstrating that multi-model approaches can be helpful in improving performance and increasing accuracy.

A number of other issues related to dynamic load balancing have been considered by the scholars. (Rahman & Yusof, 2024)) discuss the problems associated with managing big data workloads and highlight the opportunities that AI provides in this field. In particular, intelligent resource management facilitates big data analytics through predicting future trends and allocating required resources to accommodate the workload. Thus, the paper contributes to the existing discussion by proving that AI-based solutions can effectively tackle the issue.

Efficient management of energy consumption and cost reductions in the cloud have become another important factor influencing load balancing practices. (Kumar et al., 2025) suggest employing AI-based load balancing algorithms to optimize the use of available resources and reduce energy consumption. Furthermore, (Sekar, 2023) highlights the importance of multi-cloud approaches to resource optimization. In turn, the proposed model is able to combine the two aspects in order to maximize performance and minimize expenses.

Several surveys (Doukha & Ez-zahout, 2026) and (Ghelani, 2024) mention the rapid development of cloud-based AI techniques that are steadily replacing traditional optimization methods. This fact proves once again the correctness of the chosen approach, as it relies on the use of intelligent algorithms. (Al No-man et al., 2023) point out the necessity to apply AI and machine learning in order to make cloud infrastructures efficient and stable.

5 Conclusion

The presented research focused on proposing an AI-powered framework that uses unsupervised learning techniques to balance loads between VMs. The research evaluated five different clustering algorithms, namely K-Means, Gaussian Mixture Models, Spectral Clustering, Birch, and Agglomerative, in order to discover the best algorithm that could be used to distribute loads between virtual machines. As a result of conducting experiments, it was found out that the most accurate clustering model is K-Means as it achieves the highest Silhouette Score (0.220), the lowest Davies-Boulding index (1.326), and the fastest training time (0.056 s).

One of the notable strengths of the proposed model is the use of SMOTE for class balancing. Thanks to using this method, it became possible to balance all classes to have 1,551 samples each, which improves the accuracy of the results. Also, the introduction of PCA to the process allows reducing the number of features to 5 while keeping 62.7% of variance intact. In addition, thanks to using PCA, it becomes possible to make the model more computationally feasible. In particular, the proposed AI system demonstrates real-time performance as the inference time equals 0.186 ms, allowing making more than 4,700 decisions per second.

As opposed to traditional solutions such as Round Robin and Least Loaded load balancing algorithms, the proposed AI framework makes decisions that are much more accurate at acceptable computational expenses. This implies that cloud providers need to move from heuristics-based models to AI-driven models in the future.

The presented paper contributes to the existing body of knowledge regarding cloud resource optimization. For instance, the paper highlights the effectiveness of unsupervised learning models when applied to make decisions in cloud resources optimization. It would be useful to conduct further validation of the results on the real-world cloud computing datasets. Also, one could integrate the developed model into cloud computing platforms such as Kubernetes. Another interesting area for research would be multi-criteria optimization of resources allocation.

6 References

Al-Atroshi, C., & Zeebaree, S. R. (2024). Distributed Architectures for Big Data Analytics in Cloud Computing: A Review of Data-Intensive Computing Paradigm. *The Indonesian Journal of Computer Science*, 13(2).

<https://doi.org/10.33022/ijcs.v13i2.3812>

Al Noman, A., Hossain, Z., Shihab, M. A., Akter, N., Rimi, N. N., & Kabir, M. F. (2023). The Role of AI and Machine Learning in Optimizing Cloud Resource Allocation. *International Journal of Multidisciplinary Sciences and Arts*, 2(1), 262–227.

<https://doi.org/10.47709/ijmdsa.v1i2.2190>

Aldossary, D., Aldahasi, E., Balharith, T., & Helmy, T. (2025). A Systematic Literature Review on Load-Balancing Techniques in Fog Computing: Architectures, Strategies, and Emerging Trends. *Computers*, 14(6), 217.

<https://doi.org/10.3390/computers14060217>

Alipour Yengejeh, A. (2024). Optimizing AI with Advanced Data Structuring: A Comparative Analysis of K-means and GMM Clustering Techniques.

Bal, P. K., Mohapatra, S. K., Das, T. K., Srinivasan, K., & Hu, Y.-C. (2022). A joint resource allocation, security with efficient task scheduling in cloud computing using hybrid machine learning techniques. *Sensors*, 22(3), 1242.

<https://doi.org/10.3390/s22031242>

ABaskaran, S., Sayiram, S., Vedula, J., Muthusamy, R., Jiwani, N., & Kiruthiga, T. (2025). AI-Driven Dynamic Workload Balancing for Real-Time Applications on Cloud Infrastructure. 2025 3rd International Conference on Integrated Circuits and Communica-

tion Systems (ICICACS), Doukha, R., & Ez-zahout, A. (2026). A Survey of AI-Based Methods for Cloud Resource Allocation and Optimization. *International Journal of Advanced Computer Science & Applications*, 17(3), 672.

Farahi, R. (2025). A comprehensive overview of load balancing methods in software-defined networks. *Discover Internet of Things*, 5(1), 6.

<https://doi.org/10.1007/s43926-025-00098-5>

Ghelani, D. (2024). Optimizing resource allocation: Artificial intelligence techniques for dynamic task scheduling in cloud computing environments. *International Journal of Advanced Engineering Technologies and Innovations*, 3(1), 132–156.

Kumar, M., Gautam, K. K., Sharma, V., Samania, B., Vashishth, T. K., & Chaudhary, S. (2025). Enhancing cloud computing performance: A novel approach for optimizing energy efficiency through AI-based load balancing algorithm. 2025 International Conference on Intelligent Computing and Knowledge Extraction (ICICKE).

Lahande, P. V., Kaveri, P. R., Saini, J. R., Kotecha, K., & Alfarhood, S. (2023). Reinforcement learning approach for optimizing cloud resource utilization with load balancing. *IEEE Access*, 11, 127567–127577.

<https://doi.org/10.1109/ACCESS.2023.3329557>

Noshy, M., Ibrahim, A., & Ali, H. A. (2018). Optimization of live virtual machine migration in cloud computing: A survey and future directions. *Journal of Network and Computer Applications*, 110, 1–10.

<https://doi.org/10.1016/j.jnca.2018.03.002>

Olatunde-Thorpe, J., Aifuwa, S. E., Oshoba, T., Ogbuefi, E., & Akokodari, D. (2022). Integrating load balancing strategies: Conceptual frameworks ensuring optimized performance across enterprise and service provider networks. *Journal of Frontiers in Multidisciplinary Research*, 3(2), 170–181.

<https://doi.org/10.54660/IJFMR.2022.3.2.170-181>

Rahman, A. F. B., & Yusof, Z. B. (2024). Optimizing Resource Allocation for Big Data Workloads in Cloud Computing Platforms. Algorithms, Computational Theory, Optimization Techniques, and Applications in Research Quarterly, 14(8), 15–27.

Sekar, J. (2023). AI-Powered Multi-Cloud Strategies: Balancing Load and Optimizing Costs Through Intelligent Systems. *Iconic Research And Engineering Journals*, 7(2), 675–682.

Youn, D. H., & Yadav, R. (2016). Load balancing strategy using round robin algorithm.

Asia - pacific Journal of Convergent Research Interchange, 2(3), 1–10.

<http://dx.doi.org/10.21742/apjcri.2016.09.01>