IJERST

International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 22 No. 2(2) (2026)

ijerst.editor@gmail.com

editor@ijerst.com

ISSN 2319-5991 www.ijerst.com

Peer Reviewed, Referred & Indexed Journal

Research Paper

Variability-Aware Financial Workload Intelligence via Hybrid Tree Induction and QoS Surface Approximation across Multi-Cloud Substrates

T. Chalama Reddy[1*], B. RamaMurthy[2], Vemaala Gireesh[3], Votturu Sri Krishna[3], Thota Chandu[3]
[1]Professor, [2]Assistant Professor, [3]UG Student, [1,2,3]Department of Computer Science and Engineering
[1,2,3]Geethanjali Institute of Science and Technology, Nellore-Bombay Highway, S.P.S.R, Andhra Pradesh 524137, India
*Correspondence: T. Chalama Reddy (chalamareddy.t@gmail.com)

ABSTRACT

Businesses aiming to manage diverse workloads with agility, scalability, and cost efficiency increasingly rely on multi-cloud environments. However, maintaining strict Quality of Service (QoS) requirements across multiple platforms while ensuring seamless resource accessibility remains a significant challenge. Although cloud computing enables the deployment and management of business processes (BPs), handling highly dynamic and fluctuating resource demands in containerized multi-cloud environments is complex. This often leads to issues such as over-provisioning or under-provisioning of resources, ultimately affecting performance and operational efficiency. Therefore, there is a critical need for intelligent and elastic resource provisioning mechanisms that can satisfy both cloud provider objectives and end-user expectations while adhering to QoS and Service Level Agreement (SLA) constraints. This research proposes a Hybrid Classification and Regression Tree (CART)-based framework to analyze workload variability in multi-cloud environments and accurately predict QoS scores. The system integrates multiple machine learning models, including Decision Tree Classifier (DTC), Random Forest Classifier (RFC), Gradient Boosting Classifier (GBC), and Adaptive Boosting Classifier (AdaBoost). Additionally, a novel hybrid model, termed the Categorical Boosted Extra Trees (CBET), combines the strengths of Categorical Boosting Classifier (CBC) and Extra Trees Classifier (ETC). The framework follows a dual-stage CART pipeline where regression models first estimate QoS scores, which are then used as input features for classification models to categorize workload variability into Low, Medium, and High levels. Experimental results demonstrate that the proposed CBET model significantly outperforms traditional models, achieving higher accuracy, improved generalization, and enhanced robustness.

Keywords: Multi-Cloud Computing, Machine Learning, Hybrid Model CBET, Quality of Service (QoS), Cloud Resource Management.

1. INTRODUCTION

Modern financial institutions increasingly depend on distributed multi-cloud computing infrastructures to support large-scale digital banking, payment processing, fraud monitoring, and real-time transaction systems. Multi-cloud environments allow organizations to deploy services across multiple cloud providers to improve reliability, scalability, and cost efficiency. However, as the number of users, transactions, and services increases, cloud workloads become highly dynamic, leading to fluctuations in system performance metrics such as resource utilization, response time, service latency, and throughput. These variations can directly influence the QoS experienced by financial applications, making efficient monitoring and workload management an important requirement for maintaining stable service delivery.

As shown in Figure 1 multi-cloud payment platform architecture for financial institutions illustrates deployment across Amazon Web Services (Amazon EKS), Microsoft Azure (Azure AKS), and Google Cloud (Google GKE). The architecture integrates API gateway (Kong), messaging system (NATS

JetStream), and distributed database (CockroachDB) to support highly available payment services and connectivity with external payment infrastructures.

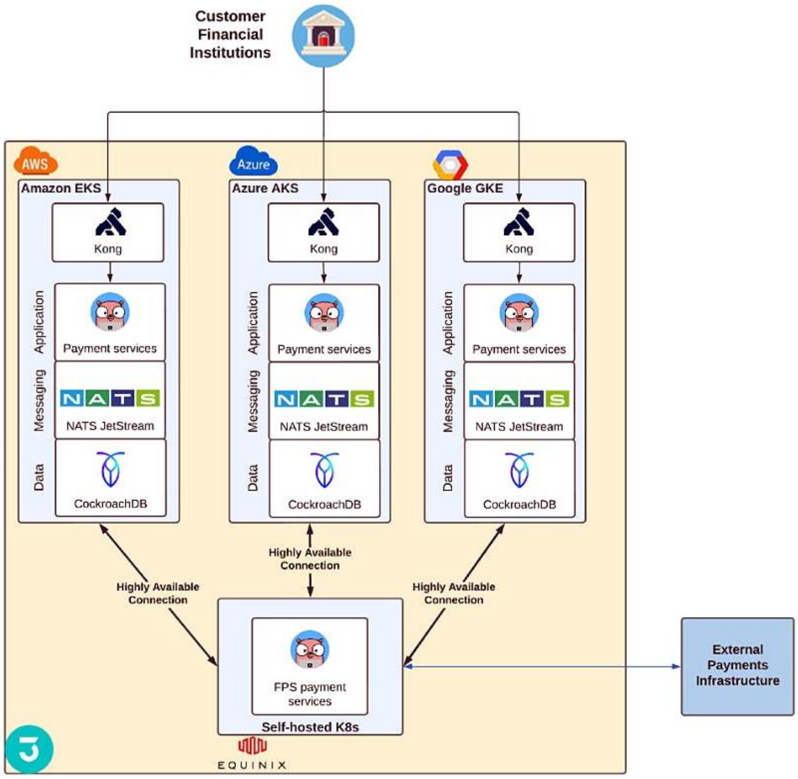

Figure 1: Multi-cloud payment resource platform

2. LITERATURE SURVEY

Shabina Ghafir et al. [1] introduced an Intelligent particle swarm optimization (PSO)-based Feedback Controller designed for load balancing in cloud computing. The method focuses on enhancing task scheduling, resource allocation, and virtual machines (VMs) migration efficiency. Salimian et al. [2] proposed an evolutionary multiobjective optimization framework to address the IoT service placement problem in fog-enabled networks. Their method is based on a fog-cloud control middle ware and incorporates the MADE-k model (Monitoring, Analysis, Decision-making, and Execution) to optimize resources. They employed PSO to autonomously manage and deploy services, focusing on maximizing fog resource utilization and improving QoS by reducing Response Time and Service Cost.

Shahidinejad et al. [3] proposed a context-aware, multi-user offloading method for Mobile Edge Computing (MEC) that incorporates Federated Learning (FL) and Deep Reinforcement Learning (DRL). The suggested approach employs the MAPE loop (Monitor, Analyze, Plan, Execute) for autonomous resource management and integrates FL to enable collaborative learning among mobile devices and edge servers. Garima Verma et al. [4] proposes an optimization algorithm to improve resource allocation and load balancing in cloud environments. This study highlights the challenges of scheduling tasks across VMs in real-time dynamic settings.

Fatemeh Jazayeri et al. [5] proposed a Hidden Markov Model Auto-scaling Offloading (HMAO) method to optimize computation offloading in mobile fog computing (MFC). The method employs a MonitorAnalyze-Plan-Execute (MAPE) control loop for resource management and utilizes a Hidden Markov Model (HMM) to predict module arrival rates and determine the optimal destination for offloading-whether to a Fog device or the Cloud. Seyed Salar Sefati et al. [6] proposed a novel approach to load balancing in cloud computing using the Grey Wolf Optimization (GWO) algorithm, focusing on maintaining resource reliability and enhancing QoS. Singhal et al. [7] proposed a Rock Hyrax-based load balancing algorithm for cloud computing to address uneven load distribution, energy inefficiency, and local maxima issues. This method employs a probabilistic load estimation and fitness-based decisionmaking inspired by the foraging behavior of Rock Hyraxes, dynamically allocating tasks to VMs based on QoS parameters such as makespan, energy efficiency, throughput, and response time. Elsakaan et al. [8] proposed a novel multi-level hybrid algorithm for load balancing and task scheduling in cloud computing. Mosayebi et al. [9] introduce cost-efficient load balancing algorithms for cloud computing using the Artificial Immune System (AIS) and Clonal Selection Algorithm (CSA). These methods dynamically distribute workloads among VMs to optimize resource utilization, minimize response times, and reduce costs.

Ayothi et al. [10] proposed a hybrid dingo and whale optimization algorithm (HDWOA-LBM) for load balancing in cloud computing environments. The method combines the hunting characteristics of dingo optimization (DOA) with the exploitation phase improvements from whale optimization (IWOA) to achieve balanced exploration and exploitation during task scheduling.

Alhijawi et al. [11] presented a novel Genetic-Based Recommender System based on historical rating data and semantic information. It is important to note that the study makes a distinction between evaluating and developing a list of probable suggestions. The Best List of Items Using Genetic Algorithms (BLIGA) uses a genetic algorithm to choose the best selections for the current user. Therefore, everyone represents a list of candidates that have been recommended. Three fitness functions are used in the genetic-based recommendation system to assess people hierarchically. Semantic similarity between items is assessed by the first function, which considers semantic information about the objects in question. Using the second function, it can compare the experience of a customer's satisfaction. The third function picks the best possible list of suggestions for the user based on the predicted ratings.

Chenyang Li et al. [12] presented QoS-aware Web Service Composition (QWSC) as a trendy subject in both industry and academia due to the recent surge in the number of web facilities. QWSC has employed a meta-heuristic approach to solving classical optimization problems. Due to its inherent flaws, such a method generally fails in large-scale situations. For example, certain meta-heuristic algorithms work well in nonstop search space, whereas QWSC's search space is discrete.

Fang Li et al. [13] discussed meeting user needs for network worth of benefit; this research presents a novel QoS assessment prototype established on the Intelligent Water Droplets (IWD) algorithm. The network QoS assessment indices generated by the model are utilized to construct a multi-objective optimization function for addressing the issue. It is then utilized to investigate the model's most critical components using mathematical simulation. The simulation results suggest that the technique is more flexible than previous approaches.

Jiang et al. [14] suggested the use of cloud service tags and informative language to extract and identify latent connections between cloud services. Alayed et al. [15], in order to compose a web service, it is necessary to understand the demands of the end-user. Each task in this pipeline provides an abstract description of specific user demands. For workflow management, web services can be gathered from several sources. The candidate list refers to a collection of services provided by various service providers for a certain activity.

3. PROPOSED SYSTEM

The proposed system introduces an intelligent analytical framework for evaluating service performance and workload variability in multi-cloud financial environments. The system systematically analyses cloud resource utilization parameters such as CPU usage, memory consumption, storage usage, network bandwidth, service latency, response time, throughput, and load balancing levels to estimate the QoS and identify workload variability patterns. By integrating advanced analytical models within a cloud monitoring framework, the system provides a structured mechanism to evaluate service behaviour and predict workload conditions across multiple cloud platforms. As shown in Figure 2 this approach improves performance assessment, supports better resource planning, and enhances the reliability of financial service infrastructures deployed in multi-cloud environments.

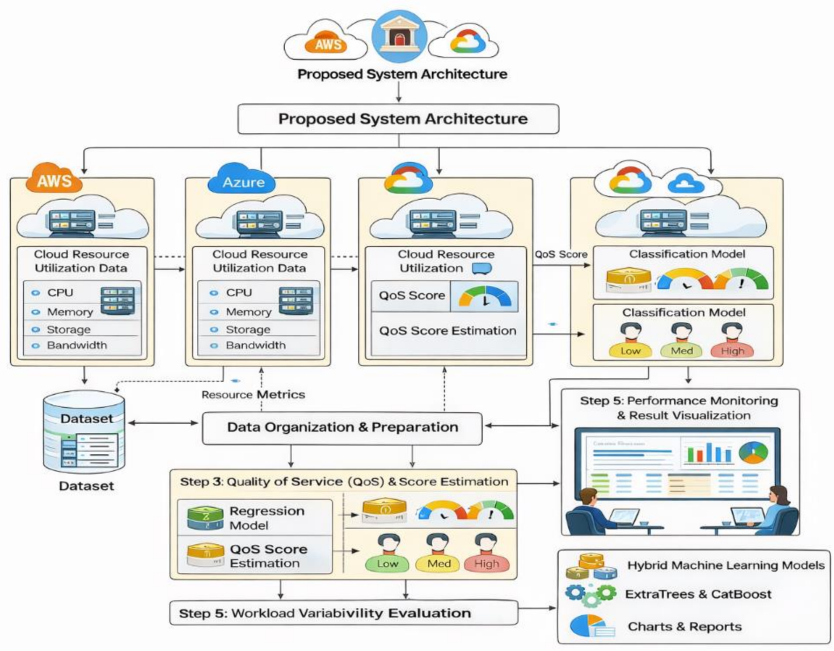

Figure 2: Proposed system architecture

Data Collection from Multi-Cloud Environment: The system collects operational data from different cloud service platforms, including metrics such as CPU utilization, memory usage, storage usage, network bandwidth, service latency, response time, throughput, and load balancing information. These parameters represent the performance characteristics of financial service workloads operating across distributed cloud infrastructures.

Data Organization and Preparation: The collected cloud performance data is organized into a structured dataset format where each record represents a service instance and its associated resource usage metrics. This structured representation allows the system to analyse system performance indicators and prepare the data for further evaluation.

QoS Score Estimation: Using the collected resource utilization parameters, the system evaluates and estimates the QoS score for each cloud service instance. The QoS score represents the overall service performance level based on operational metrics such as latency, response time, and resource utilization efficiency.

Workload Variability Evaluation: After estimating the QoS score, the system analyzes workload characteristics and categorizes the workload variability into predefined levels such as Low, Medium, or High. This classification helps understand how service demand changes across different cloud platforms.

Performance Monitoring and Result Visualization: Finally, the system presents the predicted QoS scores and workload variability levels through a monitoring interface. The results help administrators understand system behavior, evaluate service performance trends, and support decision-making for efficient multi-cloud resource management.

CBET Model

CBET-Hybrid Model combines the strengths of two tree-based ensemble techniques to improve performance in both regression and classification tasks. The model first estimates the QoS score using a hybrid regression approach and then uses the predicted QoS along with input features to classify workload variability. CBC contributes by handling complex feature interactions and ordered learning, while ET enhances randomness and reduces variance through multiple tree aggregation. This combination results in a robust and efficient model for analyzing dynamic multi-cloud workload behavior.

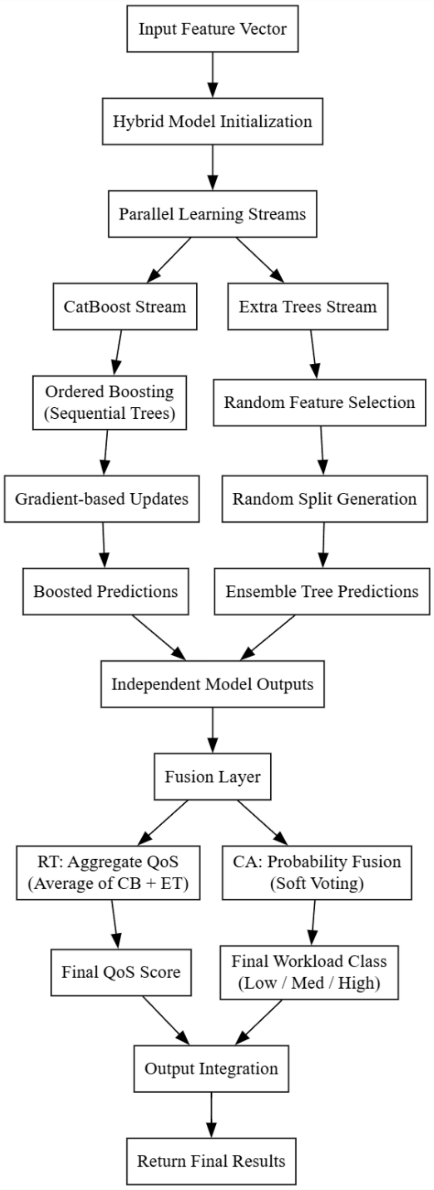

Figure 3: Internal workflow of CBET Model

Input Feature Initialization: The model takes cloud resource parameters such as CPU utilization, memory usage, storage usage, bandwidth, latency, response time, throughput, and load balancing as input features for both regression and classification stages.

Parallel Model Training (Regression Phase): Two models, CBC regressor and Extra Trees regressor, are trained independently on the same dataset. CBC focuses on capturing complex patterns, while Extra Trees introduces randomness for better generalization.

QoS Score Prediction and Aggregation: Both models predict QoS scores for the input data. These predictions are combined using averaging (Voting Regressor) to produce a final QoS score with improved stability and accuracy.

Feature Augmentation: The predicted QoS score is appended to the original feature set, forming an enhanced input vector. This step enables the classification model to use service quality information along with resource metrics.

Parallel Model Training (Classification Phase): CBC and ETC are trained using the augmented feature set. Both models learn workload patterns using CART-based tree structures with different learning strategies.

Soft Voting for Classification: During prediction, both classifiers output probability scores for each class (Low, Med, High). These probabilities are combined using soft voting to determine the final workload variability class.

Final Output Generation: The system produces two outputs Predicted QoS score (from regression hybrid), Predicted workload variability (from classification hybrid)

4. RESULT ANALYSIS

Figure 4 depicts that the left graph shows the Workload Variability Distribution, where the three classes (Low, Medium, High) are almost equally represented, indicating a well-balanced dataset suitable for classification without significant class imbalance issues. The right graph represents the QoS Score Distribution, which appears to be fairly uniform across the range (approximately 0.5 to 1.0), suggesting that QoS values are evenly spread without strong skewness. This balanced distribution in both workload classes and QoS scores supports stable training of CART-based RT and CA models, reducing bias and improving overall prediction performance.

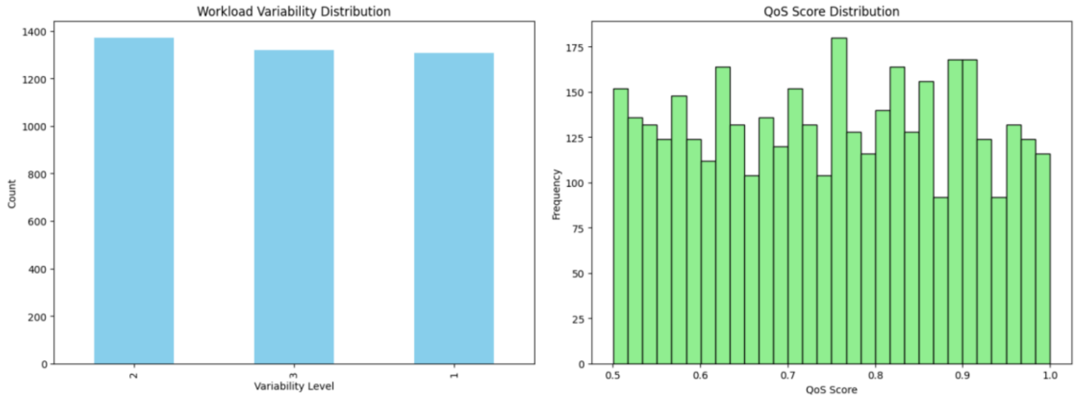

Figure 4: Distribution plots for both the target columns

Figure 5 shows that the first graph depicts the Service Type Distribution, where different service categories such as Compute, Database, AI Model, Network, and Storage are fairly evenly distributed, indicating that the dataset includes diverse workload types without significant imbalance. The right graph represents the Cloud Provider Distribution, where providers like AWS, Azure, Google Cloud, and IBM are also distributed relatively uniformly, with AWS having slightly higher representation. This balanced distribution across service types and cloud providers ensures that the CART-based RT and CA models can generalize well across different cloud environments and workload scenarios without bias toward any specific category.

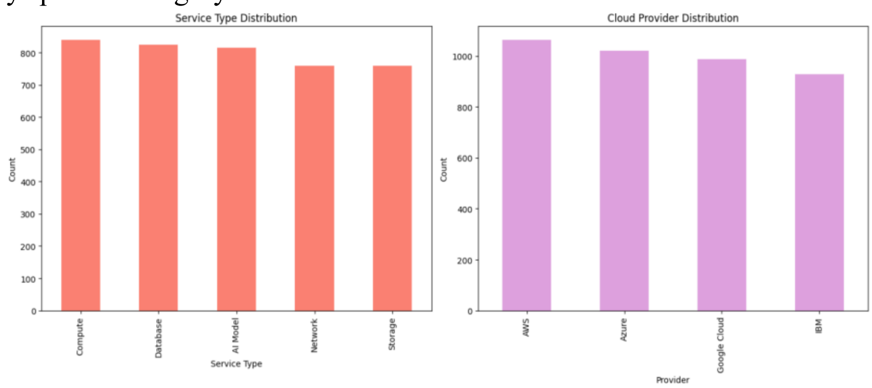

Figure 5: Distribution plots for Service type and Cloud provider columns

Figure 6 illustrates the confusion matrix for the proposed CBET model shows excellent classification performance with near-perfect accuracy across all classes. It correctly predicts all Low (253) and High (279) instances without any misclassification, and almost all Medium instances (260) with only a very small number misclassified as High (8). There are no cross-class errors between Low and other classes, indicating strong class separation. This demonstrates that the CBET model effectively captures complex patterns and significantly outperforms other models in distinguishing workload variability levels.

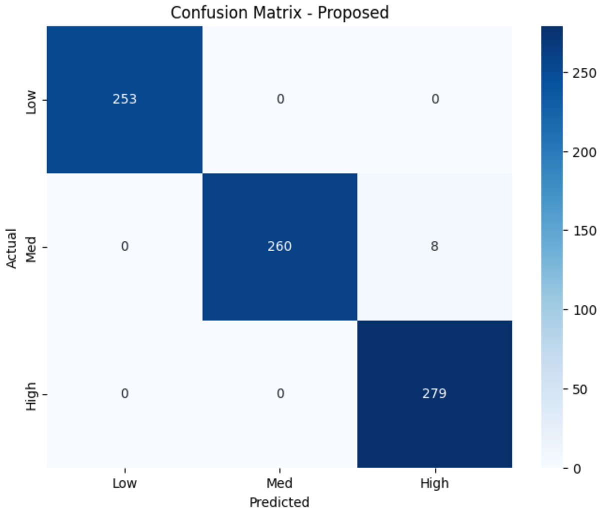

Figure 6: Illustration of confusion matrix using proposed CBET model for workload_variability class

Figure 7 shows scatter plot for the proposed CBET model shows a near-perfect alignment of predicted QoS values along the ideal diagonal line, indicating highly accurate predictions. The data points are tightly clustered around the line across the entire range, demonstrating that the model effectively captures both low and high QoS variations. Only a few minor deviations are observed, suggesting minimal prediction error. This reflects that the CBET model achieves superior regression performance with excellent generalization and precise modeling of the relationship between input features and QoS score.

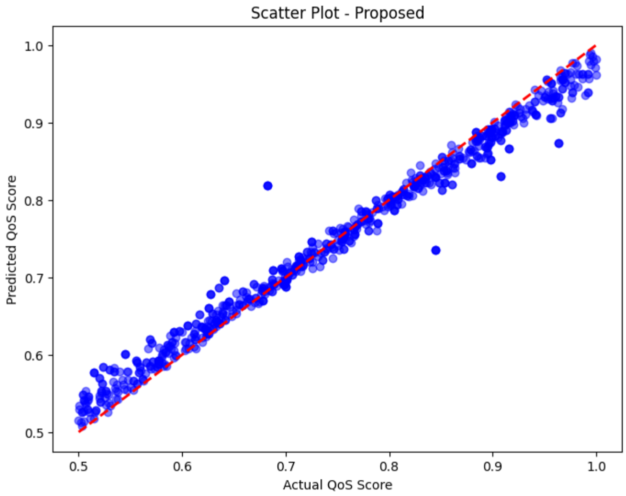

Figure 7: Scatter plot obtained using CBET model for QoS_score

Figure 8 shows prediction results obtained from the proposed CBET model demonstrate effective performance in both QoS regression and workload classification. For the given input parameters, the model predicts QoS scores in the range of approximately 0.78 to 0.81, indicating stable and high service quality across different conditions. Based on these QoS values and input features, the model accurately classifies workload variability into different categories such as High, Low, and Medium. The variation in predictions shows that the model is sensitive to changes in network bandwidth, latency, response time, throughput, and load balancing. These results confirm that the CBET model successfully captures the relationship between cloud resource parameters and system performance, providing reliable and consistent predictions.

Network_Bandwidth (Mbps)	Service_Latency (ms)	Response_Time (ms)	Throughput (Requests/sec)	Load_Balancing (%)	Predicted_QoS_Score	Predicted_Workload_Variability
650.25	175.50	400.75	850.25	80.50	0.815011	High
520.75	190.25	310.50	550.75	75.25	0.810352	Low
780.50	145.75	280.25	920.50	88.75	0.781269	Med

Figure 8: Prediction obtained on sample test data using proposed CBET Model

Comparative analysis

Table 1: Performance comparison of classification models for workload_variability class

Model	Accuracy	Precision	Recall	F-Score
Existing DTC	34.50%	42.96%	34.50%	18.88%
Existing RFC	41.75%	45.23%	41.75%	37.46%
Existing GBC	76.88%	77.23%	76.88%	76.87%
Existing AdaBoost	40.62%	40.15%	40.62%	36.86%
Proposed CBET	99.10%	99.03%	99.00%	99.03%

The table 1 shows the performance comparison of classification models for workload variability clearly highlights the superiority of the proposed CBET model over existing approaches. The existing DTC shows very poor performance with an accuracy of 34.50% and a low F-score of 18.88%, indicating weak class separation and high misclassification. The existing RFC slightly improves performance with 41.75% accuracy and a better F-score of 37.46%, but still struggles to generalize effectively. The Existing AdaBoost model performs similarly with 40.62% accuracy, showing limited improvement in classification capability. In contrast, the Existing GBC model demonstrates a significant improvement with 76.88% accuracy and balanced precision, recall, and F-score around 77%, indicating strong learning of class patterns. However, the proposed CBET model achieves outstanding performance with 99.10% accuracy, 99.03% precision, 99.00% recall, and 99.03% F-score, reflecting near-perfect classification with minimal errors.

Table 2: Performance Comparison of Classification models for QoS_score

Model	MAE	MSE	RMSE	R2-Score
Existing DTC	0.1206	0.019	0.1401	0.0196
Existing RFC	0.1183	0.0189	0.1376	0.0542
Existing GBC	0.099	0.0141	0.1188	0.2952
Existing AdaBoost	0.1193	0.0192	0.1384	0.0427
Proposed CBET	0.0171	0.0006	0.0248	0.9692

The table 2 presents performance comparison of regression models for QoS score prediction clearly demonstrates the effectiveness of the proposed CBET model over existing methods. The existing DTC shows weak performance with higher error values (MAE: 0.1206, RMSE: 0.1401) and a very low R^2 score of 0.0196, indicating poor predictive capability. The existing RFC and AdaBoost models show slight improvements but still maintain high error rates and low R^2 scores (0.0542 and 0.0427 respectively), reflecting limited ability to capture the relationship between features and QoS. The Existing GBC performs comparatively better with reduced errors (MAE: 0.099, RMSE: 0.1188) and a moderate R^2 score of 0.2952, indicating improved learning of patterns. However, the proposed CBET model significantly outperforms all others with extremely low error values (MAE: 0.0171, RMSE: 0.0248) and a very high R^2 score of 0.9692, showing near-perfect prediction accuracy.

5. CONCLUSION

The research successfully presents an effective CART-based framework for analyzing multi-cloud workload variability and predicting QoS scores using both classification and regression approaches. By integrating multiple models such as DTC, RFC, GBC, AdaBoost, and the proposed CBET, the system establishes a dual-stage pipeline where RT predicts QoS and CA utilizes it for workload classification. Experimental results demonstrate that traditional models like DTC, RFC, and AdaBoost show limited performance, while GBC provides moderate improvements in both tasks. However, the proposed CBET model significantly outperforms all existing models by achieving near-perfect classification accuracy and highly precise regression predictions with minimal error. The system also ensures robust implementation through proper preprocessing, EDA, visualization, and a Flask-based interface for real-time predictions. The research proves that combining CB and ET in a hybrid CART framework enhances model generalization, improves prediction reliability, and provides an efficient solution for intelligent multi-cloud workload management and performance optimization.

REFERENCES

[1]. Ghafir, S., Alam, M.A., Siddiqui, F., Naaz, S.: Load balancing in cloud computing via intelligent pso-based feedback controller. Sustain. Comput. Inform. Syst. **41**, 100948 (2024)

[2]. Salimian, M., Ghobaei-Arani, M., Shahidinejad, A.: An evolutionary multi-objective optimization technique to deploy the iot services in fog-enabled networks: an autonomous approach. Appl. Artif. Intell. **36**(1), 2008149 (2022)

[3]. Shahidinejad, A., Farahbakhsh, F., Ghobaei-Arani, M., Malik, M.H., Anwar, T.: Context-aware multi-user offloading in mobile edge computing: A federated learning-based approach. J. Grid Comput. **19**(2), 18 (2021).

[4]. Verma, G.: Load balancing in cloud environment using opposition based spider monkey optimization. Wirel. Pers. Commun. **137**(2), 977–996 (2024).

[5]. Jazayeri, F., Shahidinejad, A., Ghobaei-Arani, M.: A latency-aware and energy-efficient computation offloading in mobile fog computing: a hidden markov model-based approach. J. Supercomput. 77, 4887–4916 (2021)

[6]. Sefati, S., Mousavinasab, M., Zareh Farkhady, R.: Load balancing in cloud computing environment using the grey wolf optimization algorithm based on the reliability: performance evaluation. J. Supercomput. 78(1), 18–42 (2022)

[7]. Singhal, S., Sharma, A., Verma, P.K., Kumar, M., Verma, S., Kaur, M., Rodrigues, J.J., Khurma, R.A., García-Arenas, M., et al.: Energy efficient load balancing algorithm for cloud computing using rock hyrax optimization. IEEE Access (2024)

[8]. Elsakaan, N., Amroun, K.: A novel multi-level hybrid load balancing and tasks scheduling algorithm for cloud computing environment. J. Supercomput. 1–41 (2024)

[9]. Mosayebi, M., Azmi, R.: Cost-effective clonal selection and ais-based load balancing in cloud computing environment. J. Supercomput. **80**(16), 23271–23310 (2024).

[10]. Ramya, K., Ayothi, S.: Hybrid dingo and whale optimization algorithm-based optimal load balancing for cloud computing environment. Trans. Emerg. Telecommun. Technol. **34**(5), e4760 (2023).

[11]. Alhijawi, B.; Kilani, Y. A collaborative filtering recommender system using genetic algorithm. Inf. Process. Manag. 2020, 57, 102310.

[12]. Li, C.; Li, J.; Chen, H. A Meta-Heuristic-Based Approach for Qos-Aware Service Composition. IEEE Access 2020, 8, 69579–69592.

[13]. Li, F.; Wang, Y.; Li, P. The Evaluation Model of Network QoS Based on Intelligent Water Droplets Algorithm. Discret. Dyn. Nat. Soc. 2020, 2020, 1–7.

[14]. Jiang, Y.; Tao, D.; Liu, Y.; Sun, J.; Ling, H. Cloud service recommendation based on unstructured textual information. Futur. Gener. Comput. Syst. 2019, 97, 387–396.

[15]. Alayed, H.; Dahan, F.; Alfakih, T.; Mathkour, H.; Arafah, M. Enhancement of Ant Colony Optimization for QoS-Aware Web Service Selection. IEEE Access 2019, 7, 97041–97051.