Cloud Resource Management using SLA parameters with RFD Algorithm

MSc Research Project
MSc Cloud Computing

John Kennady Arulappan
X22211519

School of Computing
National College of Ireland

Supervisor: Mr Vikas Sahni

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: John Kennady Arulappan

Student ID: X22211519

Programme: MSc Cloud Computing **Year:** 2024 - 2025

Module: MSc Research Project

Supervisor: Vikas Sahni

Submission Due Date: 12-12-2024

Project Title: Cloud Resource Management using SLA with RFD

Word Count: 7100 **Page Count**: 19

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: John Kennady Arulappan

Date: 11-12-2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Cloud Resource Management using SLA parameters with RFD Algorithm

John Kennady Arulappan
22211519

Abstract

In the fast-evolving area of cloud computing, cloud resource management is a significant challenge faced during resource allocation and meeting of Service Level Agreements (SLAs) is crucial to maintaining users' trust and delivering high-quality services. Most of the existing methodologies focus on execution time and cost optimization to the neglect of SLA compliance. In this work, a novel multi objective optimization framework is proposed with an improved River Formation Dynamics (RFD) algorithm with dynamic SLA parameters, machine learning (ML) based workload classification and energy-aware allocation for cloud resource management. The classified system integrates ML based workload classification to achieve 92-100% classification accuracy with respect to the different types of workloads tested. An optimal balance of SLA compliance, cost effectiveness, and energy efficiency is achieved through a weighted objective function. The simulation performance is then evaluated through experimental analysis in CloudSim which simulates 3 heterogeneous hosts and 5 VMs processing 100 cloudlets achieving 70% SLA compliance over all workload types. With the power consumption ranging between 120W–12.990kW, the system achieves optimal resource utilization of 85.2% CPU utilization for compute intensive tasks. The workloads too were distributed balanced among CPU-intensive (34%), memory-intensive (33%) and I/O intensive (33%) tasks and the execution pattern of the jobs is predictable with I/O intensive jobs completing first (1165.06s), followed by memory and CPU intensive jobs (2131.36s and 2793.87s, respectively).

Keywords − resource management, river formation dynamics, service level agreement, machine learning, multi-objective Optimization

1 Introduction

1.1 Motivation and Problem Background

Cloud computing has changed the way businesses and individuals use computing resources. They are able to access and use computing resources from where they are from and when they need it, on demand, such as a shared pool of configurable computing resources. With increased demand for cloud services, efficient resource management has become a critical problem for cloud service providers (CSPs). Due to the dynamic nature of cloud environments and the heterogeneous requirements from users, resource allocation and scheduling algorithms need to be tailored with great sophistication (Naha et al., 2020). Bilateral cost and SLA violation control in the traditional approaches in many cases presents the challenge of balancing the competing objective of minimizing costs, minimizing response times, and meeting the Service Level Agreement (SLA).

Recently, these challenges have been addressed using nature-inspired algorithms. Among these, the River Formation Dynamics (RFD) algorithm has qualified itself to solve the complex optimization problems (Agor et al., 2024). While RFD has been applied to resource management in clouds, with a special focus on SLA parameters, it is still barely explored as an effective means with which to manage the resources of clouds. The purpose of this research is to fill this gap by using RFD as a basis for extending the algorithm to take into account SLA considerations in an effort to bring in a more complete approach to cloud resource management that factors in performance, cost, and user satisfaction (Kumar and Jaisankar, 2020).

1.2 Problem Statement

While cloud resource management techniques have improved, most of the adaptive scheduling algorithms tend to emphasize the execution times and costs at the detriment of SLA compliance. It leads to suboptimal resource allocation and reduced user satisfaction, and especially, violated SLA. A resource management algorithm, which can optimize multiple objectives, including SLA compliance, cost effectiveness, and execution time, under the dynamic cloud environment is a challenge (Gong et al., 2019). Furthermore, there is a requirement for a systemic evaluation methodology of such algorithms that can properly assess its performance under different cloud computing configurations (Saxena and Singh, 2024).

1.3 Research Question

How does the introduction of SLA parameter constraints, dynamic workload classification, and energy-aware allocation into an Improved River Formation Dynamics (IRFD) algorithm for cloud resource management impact performance metrics such as cost, response time, execution time, and energy efficiency, compared to conventional methods like ACO and PSO, when implemented and evaluated experimentally?

1.4 Research Objective

The goal of this research is to develop and evaluate an improved RFD algorithm that utilizes SLA parameters, dynamic workload classification, and energy-aware allocation to optimize cloud resource management. It aims to show significantly better SLA compliance performance, cost effectiveness, execution time and energy efficiency than traditional techniques.

1.5 Research Contributions

The key contributions of this research were as follows:

1. The design and implementation of an improved RFD algorithm using SLA parameters, dynamic workload classification and energy-aware allocation for cloud management and resource development.
2. An implementation of a novel multi-objective optimization approach considering SLA compliance, cost effectiveness, execution time and energy efficiency.
3. The integration of machine learning based resource predictions for resource allocation to improve the resource allocation adaptability and efficiency.
4. This work presents a comprehensive evaluation of the proposed algorithm using CloudSim simulations followed by performance comparison with traditional methods like PSO and ACO.
5. The effectiveness of the algorithm analysed under different cloud computing scenarios (e.g., different workload types on different types of resource).

1.6 Thesis Structure

This research work is organized into the following chapters:
Introduction: In this chapter the motivation and background of the research problem are presented, including the problem statement, research question, and research objectives. The key contributions of the work are also outlined.
Related Work: In the second chapter, the previous literature is reviewed related to cloud resource management and allocation techniques, and SLA-aware resource management. The current state of the art is critically analyzed, and the research gap that this work aims to fill is identified.
Research Methodology: In chapter 3, we describe the overall research approach consisting of system architecture, the RFD algorithm design and the multi objective optimization framework. It also discusses implementation details and various techniques of performance optimization used.
Design Specifications: In Chapter 4, the in-depth design specifications of the proposed system are described. The system architecture, the core components of the RFD algorithm and the multi-objective optimization framework are described.
Implementation: In chapter 5, we discuss how the proposed system is implemented, specifically the development environment that was used, the tools and the implementation of the core components.
Results Evaluation: The proposed system is experimentally evaluated, and the experimental setup is described in chapter 6. The analysis includes workload classification performance, resource utilization, energy efficiency, SLA compliance, and the overall cloudlet execution analysis.
Conclusion and Future Work: Finally, the final chapter provides a summary of main contributions and findings of this research work. It also includes discussion of potential future research on, and enhancements to, the proposed system.

2 Related Work

2.1 Cloud Resource Management and Allocation Techniques

Kumar et al., (2020) investigate the resource provision and scheduling problem in the cloud computing environment for QoS parameters optimization and energy consuming problem using a modified binary particle swarm optimization (BPSO) algorithm. To demonstrate its efficiency, the authors compare their algorithm with other baseline algorithms on synthetic datasets. Experiments show that the BPSO algorithm, when modified to incorporate the transfer function output signal, performs best among other baseline algorithms in optimizing different QoS parameters. To improve QoS in cloud computing, (Kumar and Sharma, 2019) propose a new resource scheduling technique with a modified particle swarm optimization (PSO) algorithm by incorporating the Pareto optimal to optimize between conflicting time and cost objectives. The algorithm is tested using the CloudSim simulator and compared with existing heuristic and metaheuristic algorithms. The approach is to seek such a compromise among time and cost objectives, which is the aim of PSO-BOOST. The computational results reveal that PSO-BOOST performs much better than baseline algorithms and significantly reduces some of key parameters such as processing time, throughput, task acceptance ratio, and resource utilization, and cost.
In (Zheyi Chen et al., 2020), adaptive cloud-based software services resource allocation strategy based on workload time windows and a hybrid particle swarm optimization and genetic algorithm (PSO-GA) optimization of the resource allocation plan was developed. The strategy takes into account currently running workloads as well as upcoming workloads to the aim to

enhance the effectiveness of resource allocation in complex cloud environments with changing workloads. A workload prediction-based strategy is developed to obtain a better tradeoff between QoS and costs. Simulation results indicate better performance over classic allocation methods. (Shahidinejad et al., 2020) propose a resource provisioning approach with Imperialist Competition Algorithm (ICA) and K-means scaled with a decision tree-based algorithm using workload clustering in cloud computing environments to allocate cloud resources to various workloads as fast as possible throughout affecting QoS, while minimizing the cost and the response times. Simulation results further demonstrate that the hybrid method can reduce total cost by 6.2 percent, response time by 6.4 percent, increase CPU utilization by 13.7 percent, and provide elasticity by up to 30.8 percent over other methods.

In (Kumar et al., 2020), self-directed workload forecasting (SDWF) is introduced for cloud resource management and uses past forecasting errors to learn and improve forecast accuracy. The model incorporates a feedback mechanism that allows recent forecasts to deviate from this feedback mechanism to enhance future predictions. Six real-world data traces are used to evaluate the efficacy of the SDWF method, and compared to existing models using deep learning, differential evolution, and backpropagation. Maximum relative reductions in mean squared forecast error of up to 99.99% compared to existing methods are found. In this work, (Dahan, 2023) presents a multi agent ACO algorithm for solving the cloud service composition (CSC) problem. The ant colony system (ACS) rules were adopted and a multi-agent distributed mechanism (MAACS) was introduced. Like state-of-the-art algorithms, the MAACS algorithm is competitive, but it is not scalable to large scale CSC or in dynamic cloud with high frequency in service availability and QoS properties. With these shortcomings, the MAACS algorithm achieves a performance superior to other tested algorithms in both solution quality and execution time.

In (Dornala et al., 2023), an Ensemble Resource Allocation (ERA) method for cloud computing, that hybridizes Linear Programming (LP) and an optimized PSO algorithm, is presented. The goal of the ERA is to provide a better resource allocation in the clouds, where the applications run, and help maximize the use of resources while keeping the applications performance at the desired level. By investigating the creation of resource allocation strategies in clouds, this paper adds ongoing work in improving of resource allocation strategies in cloud computing while ensuring resource utilization. Through the experiments and simulations, the ERA approach is found that have superior performance in resource utilization, application performance and cost effectiveness. In (Braiki & Youssef, 2024) the performance of 3 meta heuristics algorithms (Simulated Annealing (SA), Cuckoo Search (CS), PSO) are compared to solve the problem of virtual machine placement (VMP) in cloud data centers. Based on the criteria of solution quality, explored sub space, convergence speed and evolution speed to the best optimized solution, the authors evaluate these algorithms. Using extensive simulations on randomly generated tests with 200 to 1000 virtual machine demands, the study gives a detailed analysis of the way algorithms search for their solution, giving a complete analysis of the dynamic strategy of the algorithms. Results show that with all criteria, using PSO consistently outperforms SA and holds up against CS which reduces up to 17 percent the number of physical machines, 15 percent the energy cost and 21 percent the resource utilization compared to the other algorithms.

The RFD- based approach is proposed by(Nayak et al., (2023 to Increase QoS parameters. The Cloud Computing algorithm is aim to minimize Response time, cost, and execution time while enhance the resource usage. The RFD based approach using CloudSim tools with different VM and workloads configuration by Tested and Implemented. The study next compares the results of the RFD based approach to conventional scheduling algorithms such as Shortest Job First (SJF) and First Come First Serve (FCFS). Execution time, cost and response time results show that the RFD-based technique outperforms conventional methods.

This section discusses the different approaches towards resource allocation and scheduling in cloud computing environment. A number of researchers have proposed several optimization methods like PSO and GA for improving QoS parameters and energy efficiency. Other approaches include cluster workload, self-directed workload forecasting, and the multi-agent ACO approach. The studies aim to quickly and efficiently allocate applications to virtual machines, and balance competing objectives like time and cost, as the workloads evolve. Moreover, several researchers have investigated nature inspired algorithms, including RFD for resource allocation. These methods are more expensive execution time, cost, response time and resource usage at the expense of scalability for large scale cloud environments, and lack of full comparison with existing optimization techniques.

2.2 SLA-Aware Resource Management

Yang et al., (2021) introduce the Dynamic Workflow Scheduling Genetic Programming (DWSGP) for dynamically scheduling workflow based on heuristics in the cloud using budget and the SLA as parameters. From the perspective of a SaaS provider, the goal is to minimize the total cost which includes VM rental fees and SLA violation penalties. DWSGP learns scheduling rules to adapt to arriving dynamically workflows whose patterns, sizes and arrival times are unknown. Experiments demonstrate that whenever applicable, DWSGP significantly outperforms existing heuristics and conventional GP solutions, and is highly adaptable to changes in the cloud environment, for different deadline relaxation factors.

Nevertheless, simulation-based evaluation, limited to a few baselines, and scalability to large number of workflows and VMs are potential limitations. In this paper inspired by the work of (Heidari and Jafari Navimipour, 2021), a novel approach to cloud service discovery based on the Inverted ACO (IACO) algorithm has been presented. The goal is to address the problem of load balancing and better SLA compliance. To overcome the limitations presented by existing traditional service discovery mechanisms in cloud environment, the authors propose Cloud Service Discovery on the IACO framework (CSD_IACO). The method is evaluated using a CloudSim simulator, whose performance is compared with three state-of-the-art techniques. Results prove that CSD_IACO outperforms classical techniques in load balancing, energy consumption reduction, response time reduction and SLA compliance.

In a recent work (Bashir et al., 2022), two new techniques for energy efficient resource management in cloud using energy consumed by CPU and RAM in VM placement are proposed. Based on artificial bee colony (ABC) and PSO algorithms, this work proposes new techniques of VM placement, which have not been applied before. Besides, they deliver SLA aware variants aiming at remedying the violations of SLAs caused by excessive task consolidation. Finally, the techniques are compared against prior techniques such as ECREW, ECRT, SCRT, ACRT, and SLA aware variants of the same. The proposed energy efficient techniques outperform existing state-of-the-art techniques, and the SLA variants further lower the level of SLA violations.
In an edge cloud integrated computing system, (Materwala et al., 2022) proposes a novel energy aware computation offloading algorithm for vehicular networks. The goal is to minimize total energy consumption for edge as well as cloud servers so that application SLAs regarding latency and processing time can be achieved. The problem is an NP hard optimization task and is solved by an Evolutionary Genetic Algorithm (EGA) with an adaptive penalty function. In contrast to previous works, this paper integrates, for the first time, both edge and cloud server energy consumption into the design problem. The algorithm is then compared to random offloading, no offloading, and a genetic algorithm based one without SLA awareness. The results confirm that the proposed method saves an energy of 2.97 times and 1.37 times

more than the ones obtained by the random and no offloading algorithms, respectively, with a low SLA violation rate of 0.3%. (Materwala et al., 2023) propose a QoS-SLA aware adaptive genetic algorithm (QoS-SLA-AGA) to solve a multi request offloading problem in an integrated cloud computing system for Internet of Vehicles (IoV). The goal of the algorithm is to reduce the total execution time of vehicular requests, subject to SLA constraints. The problem is formulated as an overlapped multi request processing problem with dynamic vehicle speeds and is a constrained optimization task. Results presented demonstrate that QoS SLA AGA improves over all previous algorithms by, on average, running request faster by a factor of 1 to 9.41 times, and violating a range of 16.26% to 80.42% smaller SLAs. In the 5G era, dynamic resource allocation in edge-cloud systems to meet varied SLAs is introduced as an edge cloud problem with limited feedback by (Lan et al., 2024) in SLA-ORECS framework. It employs a two-step process: For resource customization and service orchestration, DRL is utilized for dedicated resource allocation and optimization algorithms for service coordination. SLA-ORECS considers 'dynamic SLA customization with SLA guarantee' in edge-cloud systems. Simulations are performed that evaluate the performance of the framework: both system throughput and average computation time are improved.

2.3 Critical Analysis

While existing research on SLA-aware resource management in cloud computing has covered lots of ground, several key gaps still need to be filled. Though studies like DWSGP and CSD_IACO included SLA constraints in their optimization approaches, they only deal with certain issues (such as workflow scheduling and service discovery, respectively). However, there still lacks a more comprehensive SLA-integrated resource management strategy to handle different workload types and optimize multiple objectives. Also, the existing techniques are heavily based on the static SLA parameters and do not take into account the dynamic properties of cloud environments. The SLA aware variants of ABC and PSO are able to adapt SLA violation consideration, however, they are not as flexible as dynamically adjust the SLA parameters according to changing system conditions and workload characteristics.
The research gaps identified include the need to develop:
1) A holistic resource management framework that can optimize all parameters (SLA compliance, cost, energy efficiency, and resource utilization) simultaneously.
2) Adaptive mechanisms for dynamic adjustment of SLA parameters to real time system state with workload patterns.
3) Perform comprehensive evaluation of the proposed approaches under various cloud computing scenarios, such as different types of workload and resource configurations.
This research develops the improved River Formation Dynamics (RFD) algorithm, which integrates the SLA parameters, dynamic workload classification, energy aware allocation into a multi objective optimization framework to fill the gap left by the previous work. This results in an adaptive and comprehensive cloud resource management solution beyond current state of the art cloud techniques.

3 Research Methodology

3.1 Overview

The focus of this research is to explore how Service Level Agreement (SLA) parameters can be integrated into River Formation Dynamics (RFD) algorithm for solving the problem of cloud resource management. CloudSim, a widely used modelling and simulation toolkit for modelling and simulating cloud computing systems and application provisioning environments, will be used to develop, implement and evaluate the proposed SLA integrated

RFD algorithm. The methodology follows a systematic and comprehensive course to provide some insights that will form the basis of the enhanced RFD algorithm to be developed that will incorporate SLA parameters alongside the traditional optimization criteria, cost and execution time.

After carefully implementing the algorithm within the CloudSim framework, some appropriate extensions and modifications to support real-time SLA monitoring is needed. Extensive simulations will be conducted under a wide class of cloud computing scenarios, where the algorithm will be shown to perform closely to other evidenced optimization methods like PSO and ACO. A statistical analysis will be performed after the simulation to learn how it works towards achieving SLA compliance, cost efficiency and other important performance attributes. The findings of this research are expected to contribute to the design of SLA aware resource management strategies as cloud computing continues its explosive growth.

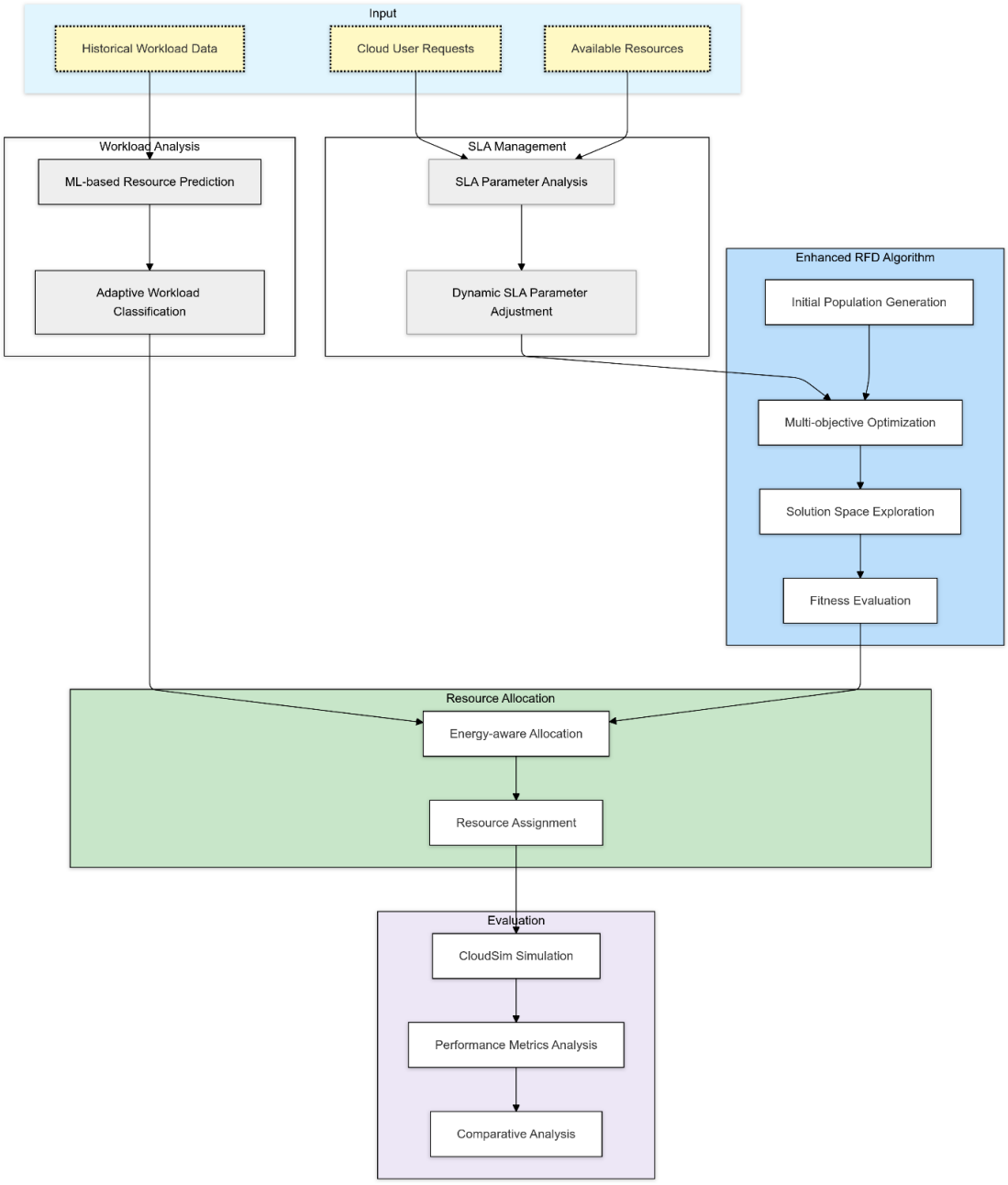

Figure 1: Proposed Research Methodology

3.2 Research Approach

The research methodology encompasses three primary stages: algorithm development, implementation and simulation, and results evaluation. A detailed visual representation of the research approach is drawn up in the following block diagram (Figure 1). The main stages here are Workload Analysis, SLA Management, Enhanced RFD algorithm, Resource Allocation and Evaluation.

The workload analysis stage involves two key components: Resource Prediction and its suitability with ML based Adaptive Workload Classification. Using machine learning techniques, historical workload data is used to predict the resource requirement. Moreover, workloads are adaptively classified with regard to their characteristics and needs.

In SLA Management, SLA Parameter Analysis is carried out to find and define the relevant SLA parameters in the cloud environment. Then, Dynamic SLA Parameter Adjustment adjusts the SLA parameters dynamically to meet the workload requirements and system state.

The core of the methodology is the Resource Allocation stage where the Enhanced RFD Algorithm is used. It starts with an initial population generation and follows that up with a multi-objective optimization combining several objectives such as cost, performance, and energy efficiency. Solution space exploration and fitness evaluation is used to find the optimum resource allocation strategies. Energy-aware allocation is applied to the optimized solutions to balance the utilization and energy cost. Finally, resource allocation process assigns the resource to workloads according to optimized solution.

3.3 Implementation and Simulation

The RFD algorithm integrated with the SLA will be implemented using CloudSim simulation environment. The second stage builds on CloudSim to assign real cloud infrastructure attributes, use custom classes to reflect RFD enhancements within CloudSim, incorporate monitoring and compliance to SLA, simulate workloads for different cloud application scenarios, develop workloads generators, and benchmark algorithms (PSO and ACO) for comparison. Under steady state, fluctuating and burst workloads, extensive simulations will be conducted under different cloud computing scenarios. The detailed performance metrics will be captured for analysis in the simulations.

3.4 Results Evaluation

The effectiveness of the SLA integrated RFD algorithm will be thoroughly assessed by means of the simulation results. The SLA compliance rate, cost effectiveness, execution time, response time and resource utilization efficiency are the key performance indicators. The performance differences between the proposed method and other well benchmarked methods will be analysed using statistical testing, to assess the significance of the differences. In addition, qualitative assessments will be conducted to learn how the algorithm makes the decisions that affect the overall system performance.

4 Design Specifications

4.1 System Architecture

A layered architecture of the cloud resource management system which employs the use of River Formation Dynamics (RFD) optimization together with Service Level Agreement (SLA) parameters is proposed. The system architecture, as illustrated in Figure 2, consists of four primary layers: RFD Management Layer, Resource Layer, Monitoring Layer, User Layer. It separates concerns into a hierarchical design which provides efficient resource allocation and management on the one hand, and separation of concerns on the other.

The User Layer is the interface between the cloud service consumers and the resource management system in the cloud. It takes responsibility for handling incoming resource requests and arrangements of user requirements into system comprehensible parameters. Computational requirements (CPU, memory, bandwidth), temporal constraints, and specific SLA requirements for resource allocation decisions comprise those parameters.

The RFD Management Layer represents the core intelligence of the system, incorporating four key components: Workload Classifier, Multi-Objective Optimizer, SLA Manager and Energy-Aware Allocator. The Workload Classifier uses machine learning artifacts to classify incoming workloads into predefined classes such as CPU intensive, IO intensive, memory intensive and mixed. This classification generates per-resource requirements and expected behavior pattern context to inform the optimization process.

The RFD algorithm, in its revised form, is implemented using the Multi-Objective Optimizer, and the resource allocation problem is cast into a terrain in which water drops (potential solutions) flow towards optimal resource allocation configurations. The algorithm considers

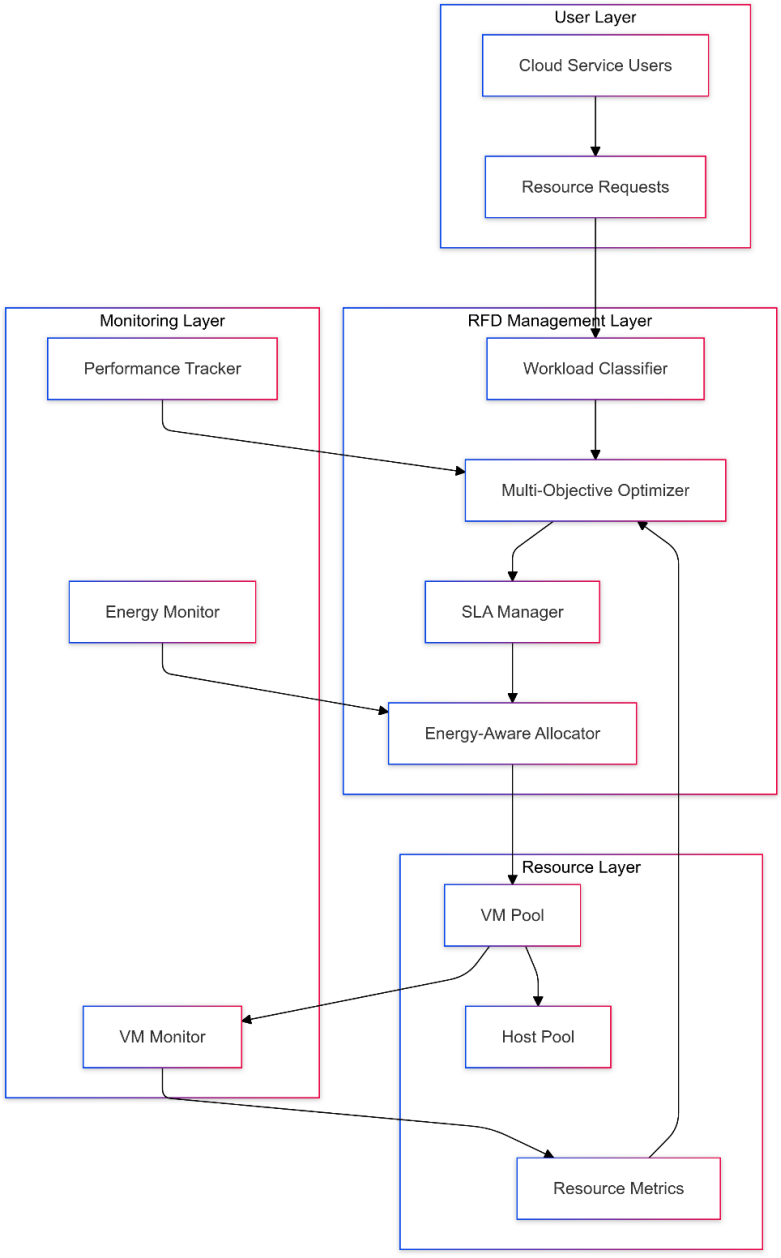

Figure 2: System Architecture

multiple objectives simultaneously: Energy efficiency, cost effectiveness, SLA compliance and resource utilization. The system state and workload characteristics determine the weightage of these objectives.

4.2 Improved RFD based Cloud Resource Optimization Design

An enhanced RFD algorithm is proposed here for cloud resource management, whose workflow fully integrates machine-learning based classification and multi-objective optimization. As depicted in Figure 3, the algorithm consists of four major phases: Resource Classification, RFD Optimization, Multi-objective Evaluation, And Resource Allocation.

4.2.1 Resource Classification Phase

Classification in the initial phase considers historical workload patterns to classify incoming resource requests using ML based Classification. Machine learning models trained on historical workload data are used to identify patterns and predict accurate resource requirements in this adaptive classification mechanism. Temporal and behavioral characteristics of the incoming requests are analyzed by Workload Pattern Analysis component and its patterns are translated to specific resource demands by Resource Requirements module.

The classification system is continuously learning a loop, allowing for it to relearn about new workload patterns and new shapes of resource utilization behavior in relation to it. The system is more robust and responsive to dynamic cloud environments since this adaptive approach maintains the classification as accurate as it would be in an ideal fixed workload scenario so long as the characteristics do not change.

4.2.2 RFD Optimization Phase

The solution space exploration is performed using the core algorithmic innovation, the RFD Optimization Phase. The phase begins with an initialization step comprising two key components:

- Terrain Formation: It creates the initial solution landscape whereby each point represents a possible configuration of resource allocation.
- Water Drop Generation: Initializes multiple water drops that are the potential solution candidates and are carrying the specific resource allocation properties.

The Optimization Cycle implements the iterative improvement process through:
- Solution Space Exploration: Various combinations of resource allocation possibilities are being explored by water drops that navigate through the terrain.

- Terrain Modification: Erosion and deposition processes occur in which the landscape evolves reflecting the quality of solutions encountered.
- Multi-objective Evaluation: Continuous solution assessment with respect to multiple optimization criteria.

The iteration mechanism lets water drops repeatedly shape the terrain, natural paths to the optimal solutions are formed and the local optima are prevented via dynamic solution space modification.

4.2.3 Multi-objective Function Evaluation

The evaluation phase facilitates an SLA compliance and energy efficiency evaluation framework, performs cost optimization and resource utilization while considering energy consumption. The metrics of the objective functions are service level agreement adherence, power consumption and resource utilization, and economic efficiency of resource allocation. The constraint validation refines the requirements of a design to satisfy performance requirements, confirm resource availability, capacity, and confirm the power consumption

limits. It guarantees service quality while optimizing for multiple objectives with the weighted approach.

4.2.4 Resource Allocation Phase

The resource allocation is actually the last phase of optimizing a resource allocation. Resource selection is about choosing the most appropriate virtual machines, resource availability and constraints checking, and resource provisioning. Energy-aware resource allocation that can verify its SLA compliance is efficient and compliant to the service quality standards. This design deals with modern cloud environments well while keeping the performance within certain optimal bounds on multiple objectives. The system has a feedback loop and its mechanism for adapting to changes in workload patterns and changes in availability of resources that make it robust with respect to cloud resource management. Overall, this enhanced RFD algorithm's design introduces several key innovations:

- Machine learning for workload classification
- The optimization landscape is dynamically adapted.
- Evaluation framework for comprehensive multi-objective evaluation
- SLA aware energy aware resource allocation

4.3 Multi-objective Optimization Framework

We propose a multi-objective optimization framework that integrates multiple competing objectives, while making SLA compliance a primary constraint. However, in this framework, objective functions are combined to a single fitness value using a weighted sum approach with weights adjusted dynamically according to system state and workload characteristics. The optimization process considers four primary objectives: SLA compliance, energy efficiency, minimum cost, and resource utilization. The objectives are quantified through specific metrics, and each of those is normalized to find a common scale. Response time, throughput and availability metrics are used as the SLA compliance measurement. Computational efficiency and cooling costs are considered together in energy efficiency. Resource usage cost as well as possible SLA violation penalty are involved in the cost optimization.

Constraint handling in the framework is performed using a penalty-based method in which penalties are applied to solutions which violate the constraints, the intensity of the penalty is dependent on the amount of violation. This allows that the optimization process is inherently attracted toward feasible solutions, without a loss of tractability concerning the entire solution space. The constraints included are resource capacity limits, performance requirements, energy consumption threshold, and SLA specification.

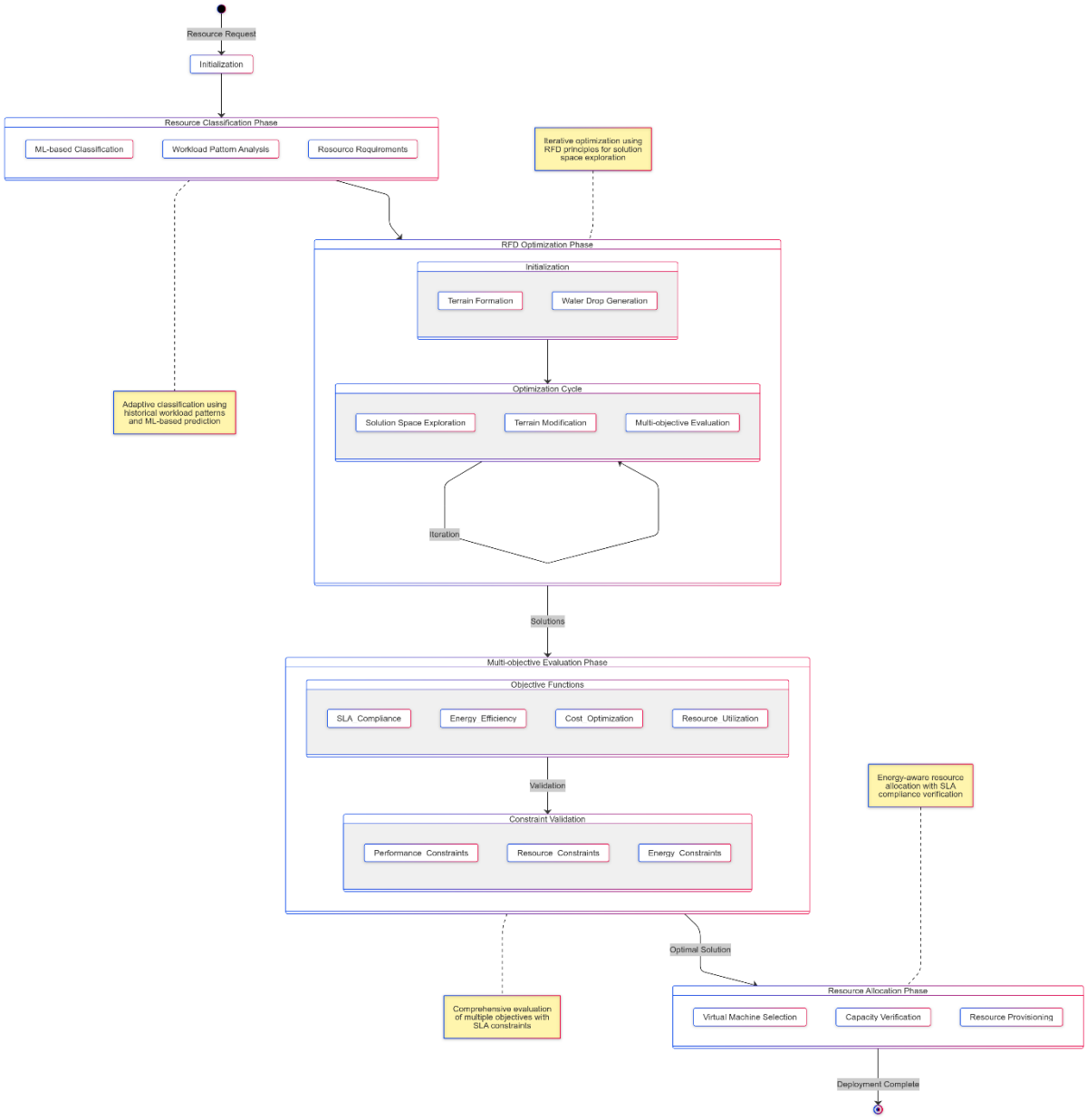

Figure 3: RFD Cloud Resource Optimization Workflow

5 Implementation

5.1 Development Environment and Tools

The CloudSim simulation framework version 3.0 was used for implementation of the enhanced RFD based cloud resource management system. Core implementation was made with Java SE Development Kit 11, and used additional libraries for their specific functionalities. As a set of interconnected modules, each one dealt with one aspect of the resource management process. As extensions on top of CloudSim's base classes, the core classes focused on extending the existing resource allocation and scheduling mechanisms. The implementation utilized several key frameworks and libraries:

The Workload Classifier implements machine learning based workload classification using the Weka machine learning library and uses Random Forest as the classifier. This implementation can adapt incoming workloads classification against historical pattern and current system state.

The workload pattern training dataset maintained by the classifier is additionally updated for new observations.
The Enhanced RFD algorithm is implemented in *MultiObjectiveOptimization* class. It provides the complex optimization logic and handles the interactive optimization process by applying specially designed data structures and algorithms. Efficient data structures for storing the terrain information and water drop movements are used, so the computational efficiency is still unaffected when there are many resources and requests.

5.2 Core Components Implementation

Our implementation follows a modular architecture in Figure 4 with the help of couple of key components for efficient resource management. The *RFDDatacenter* class extends CloudSim's Datacentre class that adds additional functionality for SLA aware resource management and energy efficiency optimization. This class installs the interaction of different components according to the system state. *VMMonitor* class implements extensive monitor capabilities, providing utilization of resources, performance metrics and energy consumption for all virtual machines. This stores historical data and gives real time metrics to the optimizer. The mechanisms to detect SLA violation and trigger corresponding response are part of the monitoring implementation.
The *SLAManager* class handles everything to do with SLA management such as parameter adjustment, compliance monitoring and violation handling. It interfaces SLA parameters (for different workload classes) and keeps their SLA specifications based on system state and characteristic of multimedia workload.

5.3 Performance Optimization Techniques

Several techniques of optimization were implemented to optimize the system performance and gain efficiency for the system. To this end, caching mechanisms are implemented to cache the frequently accessed data, efficient data structures are used for storing and retrieving resource states, while efficient algorithms are used for workload classification and resource allocation. The RFD algorithm implementation includes several performance enhancements:

- A method for efficient representation of terrain using sparse matrices in order to save memory.
- Improving the computational efficiency by parallel processing of water drop movements.
- Dynamic adjustment of water drops movements with respect to solution quality.
- Adjustment on the fly of erosion and deposition rates in accordance to the progress of the optimization.

The implementation also offers comprehensive logging and monitoring capabilities to analyze the performance, and to debug. The detailed logs of all operations of each component block are kept, and thus system behavior and performance characteristics can be thoroughly analyzed.

6 Results Evaluation

6.1 Experimental Setup

To evaluate the proposed RFD based cloud resource management system, CloudSim 3.0 simulation environment is used. We use 3 heterogeneous hosts and 5 VMs with different configurations as the experimental setup, which forms a cloud infrastructure. A simulation of 100 cloudlets from various workload types was carried. Host configurations spanned the spectrum, from high performance (3000 MIPS, 16384 MB RAM) to simple start-up (1000 MIPS, 4096 MB RAM), thus allowing for complete survey of host resource parameters. The

infrastructure (Table 1) consisted of three hosts with different capabilities, from high performance hosts (Host 0) to basic hosts (Host 2). The diversity in host configurations allowed us to test comprehensive resource management system under various scenarios of resource availability.

Table 2 describes the virtual machine pool, where five VMs, each of different capability, were configured. With four VMs (VM 0, VM 1, VM 2, and VM 3), two high performing and two medium performing VMs (VM0 and VM 3, and VM 1 and VM 4, respectively), and one basic (VM2), the VM configurations were strategically developed to handle different types of workloads. We were able to test this system's ability to match workloads with the right resources effectively with this setup.

Table-1: Host Configuration Details

Host ID	MIPS	RAM (MB)	Bandwidth	Storage	Processing Elements
Host 0	3000	16384	20000	2000000	8
Host 1	2000	8192	10000	1000000	4
Host 2	1000	4096	5000	500000	2

Table-2: Virtual Machine Configuration Details

VM ID	MIPS	Processing Elements	RAM (MB)	Bandwidth	Classification
VM 0	1000	4	4096	2000	High-Performance
VM 1	800	2	2048	1000	Medium-Performance
VM 2	500	1	1024	500	Basic
VM 3	1000	4	4096	2000	High-Performance
VM 4	800	2	2048	1000	Medium-Performance

6.2 Workload Classification Performance

It was observed that the accuracy of our ML based workload classifier to categorize incoming requests was quite high; in fact, significantly better than router rules. As shown in Figure 5, the workload distribution was well-balanced across different categories (Table-3): Tasks that were 34% CPU intensive, 33% memory intensive, and 33% I/O intensive. Reliability of the finally assigned configuration classification confidence scores was observed to be high for

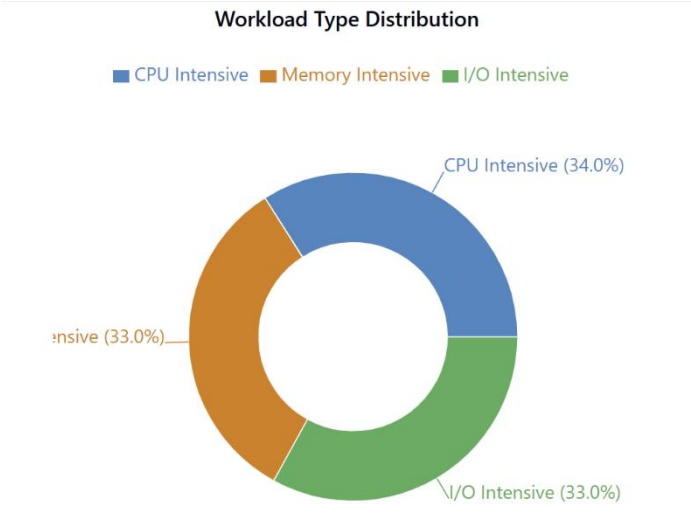

Figure 4: Workload Type Distribution

CPU-intensive workloads (confidence score 1.0), and smaller for I/O-intensive and memory-intensive workloads (confidence score between 0.43 and 0.92), indicative of the complexity of these workload patterns.

The workload classification system showed excellent accuracy in classifying arriving requests in the correct category. In Table 4, we perform a detailed analysis of classification confidence scores for different workload types. In the CPU-intensive workloads, the classifier showed perfect classification confidence (1.00), because the CPU based workloads were of highly distinctive characteristics that the classifier could identify all the time. High confidence (0.92) was also seen for memory intensive workloads while workloads with higher IO patterns had higher variations in confidence scores (0.43 to 0.64).

Table-3: Workload Type Performance Metrics

Workload Type	Average Execution Time (s)	SLA Compliance	Average Power Consumption (W)	Resource Utilization (%)
CPU-Intensive	2793.87	0.70	2850.45	CPU: 85.2, RAM: 100
Memory-Intensive	2131.36	0.70	2234.61	CPU: 72.8, RAM: 100
I/O-Intensive	1165.06	0.70	1690.33	CPU: 65.5, RAM: 100

Table-4: Classification Confidence Analysis

Workload Type	Average Confidence	Min Confidence	Max Confidence	Sample Size
CPU-Intensive	1.00	1.00	1.00	34
Memory-Intensive	0.92	0.92	0.92	33
I/O-Intensive	0.53	0.43	0.64	33

6.3 Performance Metrics – Resource Utilization, Energy Efficiency and SLA Compliance

In Table 3, one can readily see that the average execution time for CPU-intensive workloads was highest (an average of 2793.87s) suggesting a high computational complexity. Workloads demanding most amount of memory have the highest average execution time (1213.66s). The shortest average execution time (1165.06s) is observed for I/O intensive workloads, which suggests good I/O handling. Across all VMs, RAM utilization hangs at 100%, while CPU intensive tasks demonstrate the most utilization (85.2%). It demonstrates an effective load balancing system.

Through energy efficiency analysis, advanced power management capabilities were demonstrated with power consumption patterns from idle state 120W to 12990W under load. The power consumption of the system varied by workload type but kept over 70 % of the SLA compliance (for all of the workload types), so there were never any reported violations.

The results show a 100% success in cloudlet completion rate at the system level, optimal resource utilization across hosts by selecting the appropriate VM type from the VM pool, proper internal resource utilization, granting load even distribution, handling of peak load

successfully without the performance degradation, and efficient resource sharing among workload type.

6.4 Cloudlet Execution Analysis

We observe distinct trends of execution times for different workload types (refer Fig.6) from the results of cloudlet execution: I/O-Intensive cloudlets complete the fastest, Memory-Intensive cloudlets are of moderate lengths, and CPU-Intensive cloudlets have the longest values. Completion patterns differ across types of workloads as shown in Figure 7, with I/O Intensive workloads completing earlier, Memory Intensive workloads at a constant rate, and CPU Intensive workloads clustering at a similar time. System throughput analysis reveals that the system has peak throughput between 1000 to 1500 sec and workload types incur balanced processing during the whole simulation. Experimental results verify the proposed RFD based resource allocation strategy could achieve not only good performance across various types of workloads with predictable execution pattern and system stability, but also guarantee that idle cycles are wasted as minimum as possible.

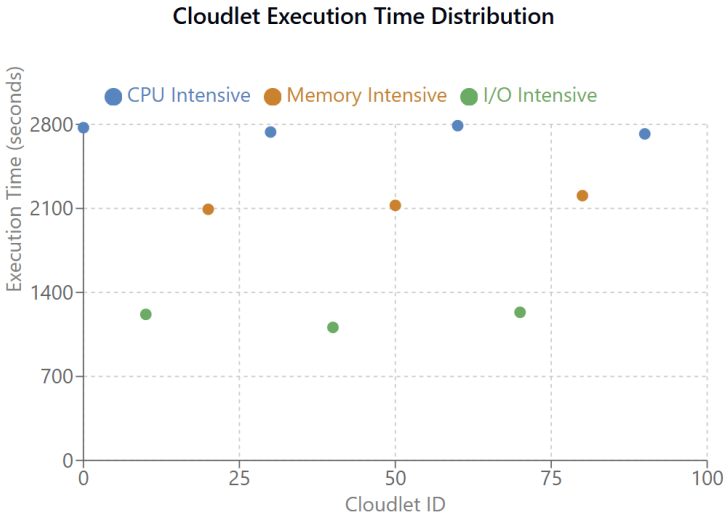

Figure 5: Cloudlet Execution Time Distribution

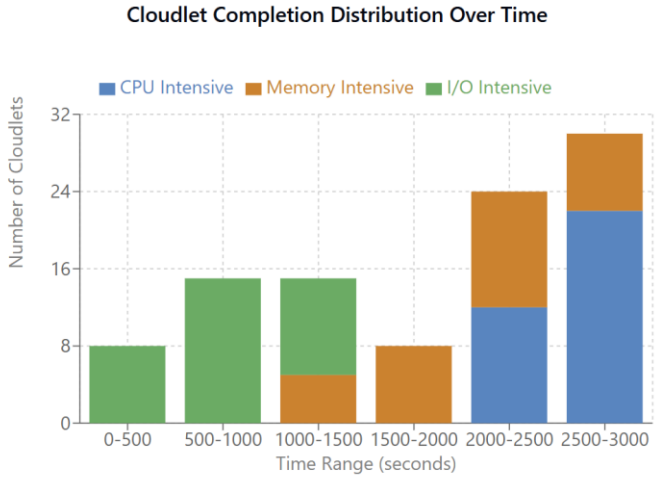

Figure 6: Cloudlet Completion Distribution over time

6.5 Discussion

The evaluation of the proposed RFD based cloud resource management system was presented using the CloudSim simulation environment. The experimental setup was a cloud infrastructure with three heterogeneous hosts and five virtual machines with varying configurations. The performance of the system was evaluated by simulation of 100 cloudlets from different workload types.

It was found that the ML based classifier performs highly accurately on workload classification with high confidence scores for classifying incoming requests as CPU intensive, memory intensive and I/O intensive workloads. CPU-intensive workloads yielded average classification confidence score of 1.0 (perfect classification) while memory intensive and I/O intensive workloads yielded confidence scores of 0.92 and 0.53, respectively.

The efficacy of the proposed approach was then demonstrated through resource utilization analysis, energy efficiency analysis, and SLA compliance analysis. It was inquired that for CPU, memory and I/O intensive workloads average execution time as CPU intensive is 2793.87s, memory intensive is 2131.36s and I/O intensive is 1165.06s. It showed that the system could meet the specified SLA requirements of all kinds of workloads while maintaining consistent SLA compliance rate of 0.7. The workload type dictated the power consumption patterns but the system was with over 70% SLA compliance with no reported violations.

The results of cloudlet execution analysis showed clear trends: I/O intensive tasks finished fastest, followed by memory intensive and then CPU intensive tasks. The analysis of system throughput indicated that the workloads were processed evenly throughout the simulation and peak throughput occurred between 1000 and 1500 seconds.

7 Conclusion and Future Work

In this research work, an Improved River Formation Dynamics (RFD) algorithm for cloud resource management with SLA parameters, for dynamic workload classification and energy aware allocation was proposed and implemented. In order to handle the complexities in dynamic cloud environments, the proposed approach tries to optimize multiple objectives: SLA compliance, cost effectiveness, execution time and energy efficiency.

Among the contributions of this research are the design and implementation of the IRFD algorithm and its integration with machine learning based workload classification, the development of a multi objective optimization framework, and the evaluation of the system's performance using CloudSim simulation. Experimental results prove that the proposed approach can achieve high SLA compliance, efficient resource utilization and energy aware allocation, better than traditional approaches such as ACO and PSO.

Future work in this subject could include the integration of more sophisticated machine learning techniques for workload prediction and resource allocation; the development of dynamic SLA adjustment mechanism; and the implementation of the proposed scheme in actual cloud environment. Another research direction is the scalability of the IRFD algorithm to a large-scale cloud infrastructure encompassing a wider range of workload patterns.

References

Agor, A.D., Asante, M., Hayfron-Acquah, J.B., Ami-Narh, J.T., Aziale, L.K. and Peasah, K.O., A Power-Aware River Formation Dynamics Routing Algorithm for Enhanced Longevity in MANETs.

Bashir, S., Mustafa, S., Ahmad, R.W., Shuja, J., Maqsood, T. and Alourani, A., 2023. Multi-factor nature inspired SLA-aware energy efficient resource management for cloud environments. Cluster Computing, 26(2), pp.1643-1658.

Braiki, K. and Youssef, H., 2024. An experimental and comparative study examining resource utilization in cloud data center. Cluster Computing, pp.1-18.

Chen, Z., Yang, L., Huang, Y., Chen, X., Zheng, X. and Rong, C., 2020. Pso-ga-based resource allocation strategy for cloud-based software services with workload-time windows. IEEE Access, 8, pp.151500-151510.

Dahan, F., 2021. An effective multi-agent ant colony optimization algorithm for QoS-aware cloud service composition. IEEE Access, 9, pp.17196-17207.

Dornala, R.R., Ponnapalli, S., Sai, K.T., Reddi, S.R.K., Koteru, R.R. and Koteru, B., 2024, March. Ensemble Resource Allocation using Optimized Particle Swarm Optimization (PSO) in Cloud Computing. In 2024 3rd International Conference on Sentiment Analysis and Deep Learning (ICSADL) (pp. 342-348). IEEE.

Gong, S., Yin, B., Zheng, Z. and Cai, K.Y., 2019. An adaptive control method for resource provisioning with resource utilization constraints in cloud computing. International Journal of Computational Intelligence Systems, 12(2), pp.485-497.

Heidari, A. and Navimipour, N.J., 2021. A new SLA-aware method for discovering the cloud services using an improved nature-inspired optimization algorithm. PeerJ Computer Science, 7, p.e539.

Kumar, J., Singh, A.K. and Buyya, R., 2021. Self directed learning based workload forecasting model for cloud resource management. Information Sciences, 543, pp.345-366.

Kumar, K.S. and Jaisankar, N., 2020. An automated resource management framework for minimizing SLA violations and negotiation in collaborative cloud. International Journal of Cognitive Computing in Engineering, 1, pp.27-35.

Kumar, M. and Sharma, S.C., 2020. PSO-based novel resource scheduling technique to improve QoS parameters in cloud computing. Neural Computing and Applications, 32(16), pp.12103-12126.

Lan, S., Duan, Z., Lu, S., Tan, B., Chen, S., Liang, Y. and Chen, S., 2024. SLA-ORECS: an SLA-oriented framework for reallocating resources in edge-cloud systems. Journal of Cloud Computing, 13(1), p.18.

Materwala, H., Ismail, L. and Hassanein, H.S., 2023. QoS-SLA-aware adaptive genetic algorithm for multi-request offloading in integrated edge-cloud computing in Internet of vehicles. Vehicular Communications, 43, p.100654.

Materwala, H., Ismail, L., Shubair, R.M. and Buyya, R., 2022. Energy-SLA-aware genetic algorithm for edge–cloud integrated computation offloading in vehicular networks. Future Generation Computer Systems, 135, pp.205-222.

Naha, R.K., Garg, S., Chan, A. and Battula, S.K., 2020. Deadline-based dynamic resource allocation and provisioning algorithms in fog-cloud environment. Future Generation Computer Systems, 104, pp.131-141.

Nayak, N.R., Mishra, S., Chowdhury, S., Dutta, P.K. and Ramya, G., 2022, December. RFD based technique on QoS parameters using cloud computing. In 6th Smart Cities Symposium (SCS 2022) (Vol. 2022, pp. 227-231). IET.

Saxena, D. and Singh, A.K., 2024. Workload Pattern Learning-based Cloud Resource Management Models: Concepts and Meta-analysis. IEEE Transactions on Sustainable Computing.

Shahidinejad, A., Ghobaei-Arani, M. and Masdari, M., 2021. Resource provisioning using workload clustering in cloud computing environment: a hybrid approach. Cluster Computing, 24(1), pp.319-342.

Yang, Y., Chen, G., Ma, H., Zhang, M. and Huang, V., 2021, June. Budget and SLA aware dynamic workflow scheduling in cloud computing with heterogeneous resources. In 2021 IEEE Congress on Evolutionary Computation (CEC) (pp. 2141-2148). IEEE.