Journal of Computer Science and Technology Studies
ISSN: 2709-104X
DOI: 10.32996/jcsts
Journal Homepage: www.al-kindipublisher.com/index.php/jcsts

JCSTS
AL-KINDI CENTER FOR RESEARCH AND DEVELOPMENT

| RESEARCH ARTICLE

Resource Allocation in The Cloud Environment with Supervised Machine learning for Effective Data Transmission

Mohammad Elham Ebadi[1], Wang Yu[2], Khoshal Rahman Rahmani[3] ✉ and Musawer Hakimi[4]
[12]*Computer Science and Software Engineering Faculty, Hohai University, Nanjing, China*
[3]*Computer Science Faculty, Kunduz University, Kunduz, Afghanistan*
[4]*Computer Science Department, Samangan University, Samangan, Afghanistan*
Corresponding Author: Khoshal Rahman Rahmani, **E-mail**: khoshalrahman.rahmani@gmail.com

| ABSTRACT

Resource allocation in the cloud environment for 5G applications can be explained by referring to the strategic distribution and necessary assignment of computing resources such as virtual machines, storage, and network bandwidth that meet the dynamic demands of applications and services. The framework proposed is on resource allocation in the cloud environment by BRoML for 5G applications. In the proposed BRoML model, the Backtracking Regularized model is incorporated for the effective calculation of the resources in the cloud environment. The optimization is performed for the effective computation of resources in the cloud environment through the computed resources. Using the estimated optimized values, a machine learning model can be trained and tested to classify resource allocation. In this regard, the simulation analysis is compared to BRoML with traditional schemes like SVM and RF. The result shows that BRoML has a higher resource utilization while exhibiting lower latency, higher increased throughput, and a better efficiency score overall. Machine learning techniques and optimization mechanisms give flexibility and intelligence to BRoMl in solving resource allocation issues within cloud computing. These results reinforce the view that BRoML can create a strong impact on the development process of cloud computing with its dynamic, intelligent solution in resource allocation optimization under various scenarios.

| KEYWORDS

Cloud Environment, Resource Allocation, Machine Learning, Optimization, Backtracking.

| ARTICLE INFORMATION

ACCEPTED: 01 July 2024 **PUBLISHED:** 26 July 2024 **DOI:** 10.32996/jcsts.2024.6.3.3

1. Introduction

The cloud environment has, over the last couple of years, emerged to take its place as one of the most radical changes in information technology, changing how businesses and individuals interact with digital resources. Characterized by remote server hosting and ubiquitous internet access, this dynamic computing paradigm changes the traditional IT landscape in a very big way. In turn, organizations are better placed to continue reaping the benefits of the cloud for providing adequate storage, processing, and management of huge data volumes and a variety of applications [Bal, 2022]. Hence, cloud computing offers an essential aspect of scalability that helps bring flexibility to heavy businesses that always desire to respond to dynamic business environments [Khan, 2022]. Above all, the pay-as-you-go style reduces the economic dynamics of IT infrastructure by making it more accessible and affordable. With the increasing demands in agility, innovation, and resource efficiency, the cloud environment can be said to be one of the core technologies driving industries toward digital transformation [Merluzzi, 2021]. Resource allocation in cloud environments brings forth a critical facet underlining any modern computing infrastructure's efficiency and adaptability [Zhou, 2021]. Resource allocation in this very dynamic field of cloud computing has been referred to as the strategic assignment and distribution of computational resources such as processing power, storage, and bandwidth in such a manner that plural and fluctuating user demands, applications, and workloads may be accomplished. This process comes with elasticity and only on-demand characteristics that enable organizations to scale up or down their resources in real time in accordance with the immediate requirements [Khumalo, 2021]. Cloud providers offer huge pools of virtual resources, enabling the execution of tasks with high performance at low costs and better operational flexibility. In terms of the dynamism in digital demands, efficient resource allocation within the cloud is central to obtaining the best performance, cost, and responsiveness in the delivery of services and applications to businesses [Iqbal, 2021].

Resource allocation in a cloud environment is of paramount importance, and machine learning offers intelligent solutions in terms of optimizing computational resource utilization [Nurcahyani, 2021]. Due to the dynamic and unpredictable nature of workloads in the cloud, traditional static resource allocation models are inadequate to meet efficiency and performance demands [Elgendy, 2021]. Machine learning algorithms leverage historical data, patterns, and real-time analytics to predict future resource needs accurately [Dittakavi, 2021]. These algorithms learn from the changeable workloads and balance resource allocation accordingly as workload demand changes. This is possible because machine learning, through continuous learning and optimization, maximizes the overall efficiency of cloud resource utilization and enables computing resources to be exactly where and when required [Shah, 2021; Cao, 2021; Shukla, 2021; Talaat, 2022]. The predictive and adaptive approach not only means performance optimization but also helps in cost saving by avoiding overprovisioning or underutilization of any resource [Mishra, 2022]. Machine learning, therefore, is an extremely important tool in achieving dynamic and responsive resource allocation strategies with the ever-increasing complexity and scale of cloud environments.

Coupling resource allocation in the cloud environment with machine learning represents a sophisticated approach to ensure effective data transmission [Sliwa, 2021; Lakhan, 2022; Xu, 2021; Sami, 2021]. For sustaining efficient and responsive cloud service providers, computational resource allocation is very critical in the age of burgeoning demands for data. This kind of intelligence within the framework of resource allocation through integration with machine learning algorithms empowers it with dynamic and data-driven decisions [Shaw, 2022]. These algorithms use historical patterns of use, current network conditions, and application requirements to correctly predict and allocate resources [Huang, 2021]. If applied to data transmission, machine learning can enhance bandwidth allocation, route optimization, and latency management in order to ensure data is received seamlessly and promptly [Kruekaew, 2022]. In this respect, machine learning allows for continued learning about evolving workloads, giving overall reliability and performance for the data to be transmitted in the cloud. This not only resolves in an integrated approach the challenges in data transmission that exist to date but also provides the much-needed intermediate step toward engineering a cloud-scale infrastructure that is able to grow and respond according to the ever-increasing demands of modern applications and services [Ahmadi, 2023].

The paper makes a substantial contribution through invention in cloud computing via the introduction of the innovative framework known as Backtracking Regularized Optimized Machine Learning for resource allocation in a cloud environment. The main idea is to combine machine learning genetic optimization with a sophisticated backtracking mechanism in a dynamic and adaptive approach to resource allocation in the cloud environment. In this paper, it is shown that BRoML comes out ahead of traditional techniques such as SVM and RF on most parameters of resource utilization, latency, throughput, and general efficiency, combining comprehensive experimentation with comparative studies. This framework is able to learn from historical data through the backtracking mechanism embedded in it and thus be more intelligent and adaptive concerning changing workload patterns. Moreover, the incorporation of machine learning techniques enables BRoML to support data-driven decision-making, therefore achieving a much more efficient resource management process.

2. System Model

Let R(t) be the cloud resources available at time, t, D(t) – the demand for resources, and P(t) – the predicted future demand output from the machine learning algorithm. The rate of change of the available resources dt/dR captures how responsive the system is to the changing demands of cloud workloads. This rate depends both on the current demand, D(t), and the predicted future demand, P(t), which creates a dynamic equation that captures the responsiveness of the cloud's infrastructure, defined in equation (1).

$$dt/dR = D(t) + P(t) \tag{1}$$

Here, D(t) is the current resource demand, whereas P(t) is the machine learning prediction of future demand. P(t) is continually optimized by the machine learning prediction engine given historical usage patterns and real-time data, as well as learning from previous resource allocation decisions.

2.1 Theorem 1 Adaptive Bandwidth Allocation Theorem

In the field of machine learning-based cloud transmission, the Adaptive Bandwidth Allocation Theorem states that the optimal bandwidth allocation function B(t) depends on real-time D(t) demands in data transmission and on the precision P(t) of the performed machine learning predictions. Mathematically, it is expressed as in equation (2).

$$B(t) = \gamma \cdot (D(t) + P(t)) \qquad (2)$$

Here, γ is the scaling factor, which captures the ability of the system to adapt to changes in data transmission demands. This theorem underlies adaptive bandwidth-allocation approaches that exploit machine learning predictions about the desired effectiveness of data transmission in a cloud environment in the dynamic adjustment of bandwidth resources.

2.2 Theorem 2: Latency Optimization Theorem for Cloud Data Transmission

In the context of machine learning-driven resource allocation for cloud data transmission, the Latency Optimization Theorem stipulates that the optimal resource allocation for minimizing latency L(t) can be realized only when the predicted future demand for data transmission, P(t) is equal to the current demand, D(t). Above stated theorem is represented in equation number 3.

$$L(t) = \delta \cdot| D(t) - P(t) | \qquad (3)$$

Here, δ reflects the sensitivity of a system to the deviations between real and predicted demands in data transmission. The following theorem thus underlines the matching of predictive analytics with real-time demand to optimize latency, which will help in more efficient data transmission in cloud environments.

3. Proposed Backtracking Regularized Optimized Machine Learning (BRoML)

The proposed BRoML approach to resource allocation in a cloud environment with machine learning is intended to tackle the challenge of effective data transmission. BRoML has integrated backtracking mechanisms, regularization techniques, and optimization algorithms for the dynamic allocation of resources for carrying out effective data transmissions in cloud environments. For example, the backtracking component of BRoML includes the analysis of historical data patterns related to the demand for data transmission. The algorithm uses a backtracking mechanism to readjust resource allocations, combining historical demands Dhistory(t) with current demand D(t) to find the back-tracked demand Dbacktrack(t). The backtracking is formulated and defined in equation (4)

$$Dbacktrack(t) = \alpha \cdot Dhistory(t) + (1 - \alpha) \cdot D(t) \qquad (4)$$

where α is the backtracking coefficient.

Coupled Regularization Techniques and Machine Learning Models— Equation 2: BRoML would enhance the accuracy regarding future demand predictions by coupling the effect of regularization techniques and their machine learning models. One could create a regularized prediction Pregularized(t) from the previous machine learning predictions P(t) by simply adding the regularization term R(t), as stated in the equation.

$$Pregularized(t) = P(t) + \lambda \cdot R(t) \ (8)$$

where λ is the regularization parameter. In BRoML, the optimized resource allocation will combine real-time data transmission demands, D(t), with regularized machine learning predictions, Pregularized(t). The optimization algorithm shown in Equation (9) determines the resource allocation R(t)

$$R(t) = argmin(| D(t) - Pregularized(t) |) \qquad (9)$$

Improvement in this process is done by having the amount of booked resources at any time self-adjusting to minimize the gap between real and forecast demands, hence optimizing efficiency in data transmission within the cloud environment. In this line, the BRoML algorithm converges to a near-optimal resource allocation—not Ropt results directly from iterative refinement of predictions and allocations as defined through equation 10 by backtracking, regularization, and optimization.

$$optlimiterations \rightarrow \infty R(t) = Ropt \qquad (10)$$

This is a very important convergence theorem that ensures BRoML refines its predictions and allocations over an infinitely large number of iterations to ensure the convergence of the algorithm toward optimal resource allocation, hence improving the efficacy of data transmission in a dynamic cloud environment. Figure 1: Proposed BRoML model for resource allocation in the cloud environment.

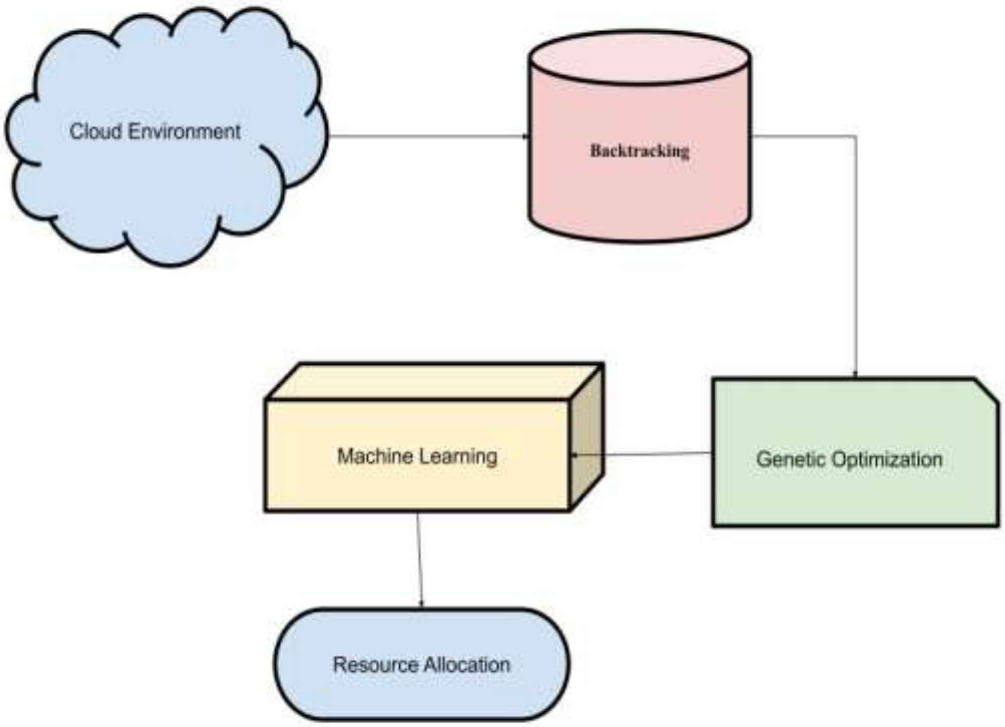


Figure 1: Flow Chart of Proposed BRoML

3.1 BRoML Optimization for Resource Allocation in Cloud Environment

It is a strong algorithmic technique of genetic optimization processed within the BRoML framework of resource allocation in a cloud environment. The approach takes its cue from the fundamental principles of natural selection and genetic variation to iteratively refine the allocations made by the resources. Genetic optimization is a process representing the individuals in a population with probable solutions of resource allocation, an evaluation of their fitness through a defined objective function, and applying other operators that are genetically inspired, such as crossover and mutation, to yield new generations of candidates of resource allocation. First, it initializes a population containing configurations of resource allocation. Each configuration is encoded as a chromosome, while genes encode certain parameters of resource allocation. The fitness function evaluates the fitness of each individual. It is a measure of how every resource allocation corresponds to the objective of optimization—for example, minimal difference between actual and predicted demand. Thereafter, the roles of genetic operators come into play: crossover combines the genetic material of two-parent individuals to form offspring by simulating the recombination of genetic information. It makes very slight random alterations in the genes of an individual to facilitate diversity in the population. The evolution of resource allocation chromosome is performed over generations whereby a fitness value attached to each set of individuals increases the possibilities of high-fitness example selection for reproduction. This simulates the process of natural selection, where fit individuals get the most chances to pass their genetic material to the next generation.

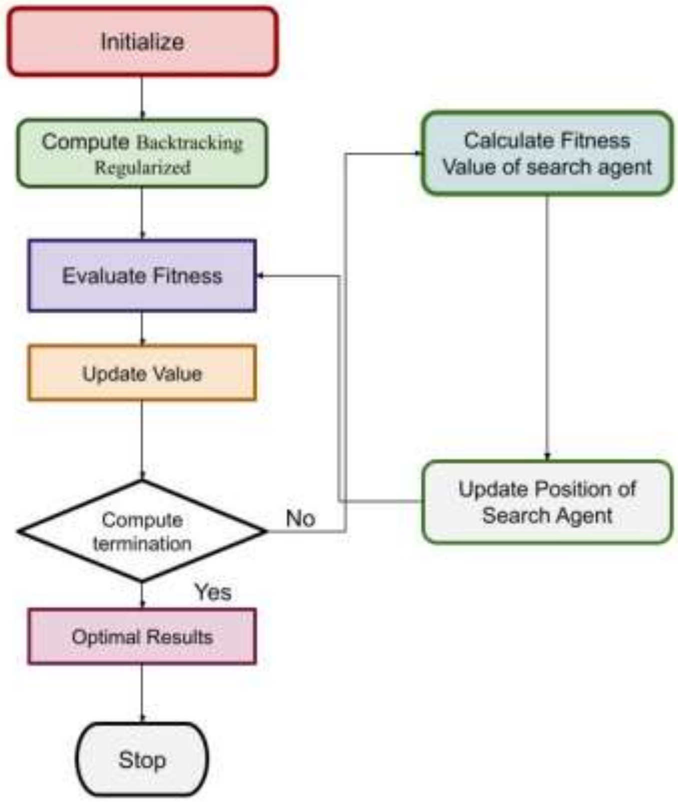


Figure 2: Flow Chart of Genetic Optimization

The flow chart of the proposed BRoML model for the optimization of the features in resource allocation is given in Figure 2. The genetic optimization process can be mathematically expressed with equations:

Initialization: $Population = \{Individual1, Individual2, \ldots, Individualn\}$
Fitness Evaluation: $Fitness(Individuali) = Objective\ Function(Resource\ Allocationi)$
Selection: Select Individuals for Reproduction based on FitnessSelect Individuals for Reproduction based on Fitness
Crossover: $Offspringj = Crossover(Parentk, Parentl)$
Mutation: $Mutate(Offspringj)$

The initial population of resource allocation configurations is represented as chromosomes. Each chromosome, denoted as h, Chi , consists of genes encoding specific resource allocation parameters. The population is defined in equation (11)

$$Population = \{Ch1, Ch2, \ldots, Chn\} \quad (11)$$

An objective function determines the fitness of each individual in the population, which measures how good the allocation of corresponding resources is in achieving optimization goals. Let now the fitness of the ith chromosome be Fi×ti, and let an objective function be given by equation 12 as (h)Obj(Chi).

$$F_i t_i = Obj(Chi) \quad (12)$$

The fitness of an individual is what describes whether the individual will or will not be chosen for the reproduction process. Commonly used is the roulette wheel selection method, the basic rule that states that the chance of selection for individuals is strictly proportional to their fitness. The probability of the selection of Pi of every individual is expressed in equation 13.

$$i = 1/n \sum_i \frac{F_i t_i}{F_j t_j} \quad (13)$$

Crossover merges the genetic material of two parental individuals to create a child. Let Parentk and Parentl be the two chosen parents. The crossover operation creates an $Offspring_i = Crossover(Parent_k, Parent_l)$. Mutation introduces small random alterations in the genes of an individual to enhance genetic diversity. Let the mutation operation on chromosome Chj be denoted as Mutate (Chj). These genetic operations are applied iteratively to generate successive generations of resource allocation configurations. The evolution of the algorithm takes place over generations, and it is continued until its convergence or some stopping criterion.

```
# Genetic Optimization Pseudocode for Resource Allocation in BRoML
# Parameters
population_size = 100
crossover_probability = 0.8
mutation_probability = 0.1
max_generations = 50
# Initialization
population = initialize_population(population_size)
# Main Genetic Optimization Loop
for generation in range(max_generations):
    # Evaluate fitness of each individual in the population
    fitness_values = evaluate_fitness(population)
    # Selection
    selected_parents = select_parents(population, fitness_values)
    # Crossover
    offspring = crossover(selected_parents, crossover_probability)
    # Mutation
    mutated_offspring = mutate(offspring, mutation_probability)
    # Evaluate fitness of mutated offspring
    mutated_fitness_values = evaluate_fitness(mutated_offspring)
    # Replace old population with new population
    population = replace_population(population, mutated_offspring, mutated_fitness_values)
# Final result
best_solution = find_best_solution(population, fitness_values)
```

4. Resource Allocation with BRoML Machine Learning

Resource allocation in the BRoML framework is a dynamic process that incorporates machine learning predictions in the backtracking optimization steps to enhance the efficiency of data transmission in cloud environments. BRoML can be defined as the integration of machine learning predictions to predict the future requirements of data transmissions. Let P(t) states the machine learning prediction at a particular time t, and that gets predicted based on historical and real-time observations. The equation of regularized machine learning prediction is derived in Equation (14)

$$Pregularized(t) = P(t) + \lambda \cdot R(t) \qquad (14)$$

where R(t) is the term of regularization, and λ the value of the regularization parameter, controlling how much λ in the historical data will influence the prediction derived using machine learning. This regularization will prevent overfitting and allow the generalization of the machine learning model, representing different data transmission patterns. With this in mind, backtracking is presented as one of the fundamental principles of adaptability in view of historical data in the decisions of resource allocations. The backtracking mechanism adjusts the resource allocation based on a combination of historical data $Dhistory(t)$) and the current demand $D(t)$). The backtracked demand $Dbacktrack(t)$) is determined in equation (15).

$$Dbacktrack(t) = \alpha \cdot Dhistory(t) + (1 - \alpha) \cdot D(t) \qquad (15)$$

In equation (15) α is the backtracking coefficient, which modulates the effect of historical data on the backtracked demand. As α varies from 0 to 1, the system learns from past performance and changes the strategy of resource allocation through learning backtracked demand. It makes optimization overall in the process of resource allocation in such a way that the discrepancy between actual demand D(t) and regularized machine learning prediction Pregularized(t) is minimized. The optimization equation is formulated as in equation (16)

$$R(t) = argmin(|\ D(t) - Pregularized(t)\ |) \qquad (16)$$

This process dynamically balances the resource allocation according to historical patterns and current demand. When machine learning is used in combination with prediction, backtracking, and optimization based on the BRoML model, it helps form an agile resource allocation system that boosts data transmission in cloud environments. Thea linear machine learning prediction model, where the predicted demand P (t) is the linear combination of historical demand and features computed as in equation (17)

$$P(t) = \beta 0 + \beta 1 \cdot Dhistory(t) + \beta 2 \cdot F(t) \qquad (17)$$

Where β0,β1,β2,... are the coefficients of the linear model learnt during the training phase. In the present model, to prevent over fitting and to enhance the generalization, a regularization term is added. The regularized prediction Pregularized(t)) is provided in equation (18)

$$Pregularized(t) = P(t) + \lambda \cdot R(t) \qquad (18)$$

In equation (18): • R(t) is the regularization term; • λ is the regularization parameter. Generally, the regularization term R(t) is a penalty term applied on the linear model's coefficients. One of these most used regularization methods is L2 regularization, or ridge regularization, where the regularization term is expressed by summing the squared coefficients, as shown in equation (19):

$$Minimize \sum t = 1/T(D(t) - P(t))^2 + \lambda \cdot R(t) \qquad (19)$$

The general optimization task over the machine learning training stage is that a regularization sum of squared errors between the estimated demand P(t) and the actual demand D(t) is minimized; the obtained regularization term is written in the equation below in the form:

$$Minimize \sum t = 1/T(D(t) - P(t))^2 + \lambda \cdot R(t) \qquad (20)$$

To address this optimization problem, optimization techniques such as gradient descent have been applied, in which degenerative Ly calculates the derivatives of the goal function in terms of the coefficients, βi, and subsequently iteratively updates them. In machine learning, this process is training a model on historical data so that it can learn the coefficients and apply this trained model to predict future demands on resources, P(t). The importance of using the regularization term is to strike a good balance between fitting the training data well and not allowing the model to pick up a lot of complexity.

4. Results and Discussion

The results section presents the outcomes of applying the BRoML framework to real-world or simulated scenarios. This includes quantitative metrics such as resource utilization, latency, and overall system performance.

Table 1: Simulation Setting for BRoMl

Parameter	Value/Setting
Simulation Duration	1000-time units
Population Size	50
Crossover Probability	0.8
Mutation Probability	0.1
Maximum Generations	50
Backtracking Coefficient	0.3
Regularization Parameter	0.01

Table 1: Simulation settings of BRoML framework with the key parameters and their corresponding values used for the simulation process Results of BRoML Framework The simulation in the following section will be run for 1000-time units to get the full view of the performance. The initial population size for the organisms was 50. This is also related to the number of potential solutions that would be initially considered for the genetic optimization process. The probability for crossover in the genetic algorithm used for optimization was set to 0.8, meaning the probability with which genetic material should be exchanged between the selected individuals. A mutation probability of 0.1 was also applied, meaning in 10% of all generations, random changes in the genetic material will be introduced to ensure a diversified population. Finally, the maximum number of generations through which the process of genetic optimization is allowed to run has been set to 50 to provide efficiency and control over the search procedure for the optimal solution. An important parameter in the backtracking mechanism, one that sets how much weight age to give

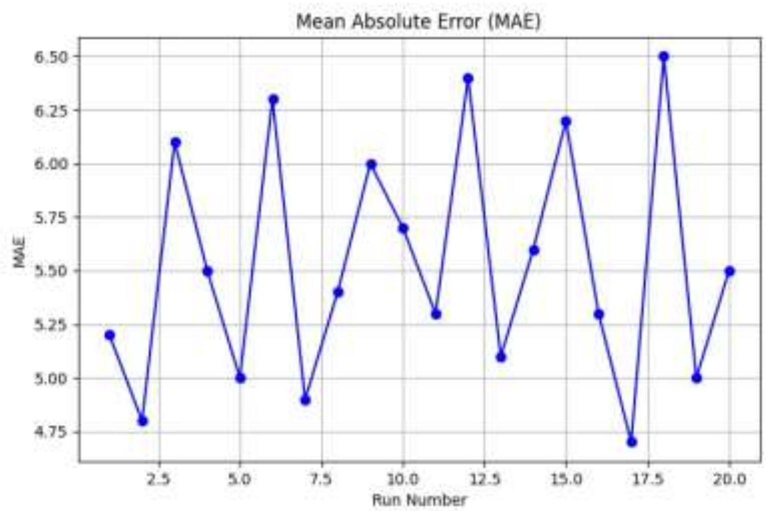
historical data, was set to 0.3. This coefficient will aid in adapting resource allocations according to their past performance. Another thing worthy of mentioning here is a regularization parameter that is important in robustifying machine learning predictions, and it was set at a value of 0.01. This is the strength parameter, often referred to as the regularization strength parameter, of the regularization in a given machine learning model. Such regularization prevents overfitting and hence ensures generalization across varying data patterns. These simulation settings provide a lead-in to understanding how the BRoML framework functions in resource allocation optimization within cloud environments in search of exploration-exploitation semantics to effective and adaptive results. The datasets considered for the analysis details are The proposed model focused on the design of the effective resource allocation scheme through VM for the 5G applications. The dataset for the analysis selected is a publicly available dataset: https://www.kaggle.com/datasets/omarsobhy14/5g-quality-of-service. This dataset provides information about the effective resource allocation scheme in 5G applications. In the dataset, there exist a total of 400 samples. The features or variables in the dataset will be Timestamp, User_ID, Application_Type, Signal_Strength, Latency, Required_Bandwidth, Allocated_Bandwidth, and Resource_Allocation. To normalize the features existing in this dataset, pre-processing of collected data is done.

Table 2: Optimization Value for the BRoML

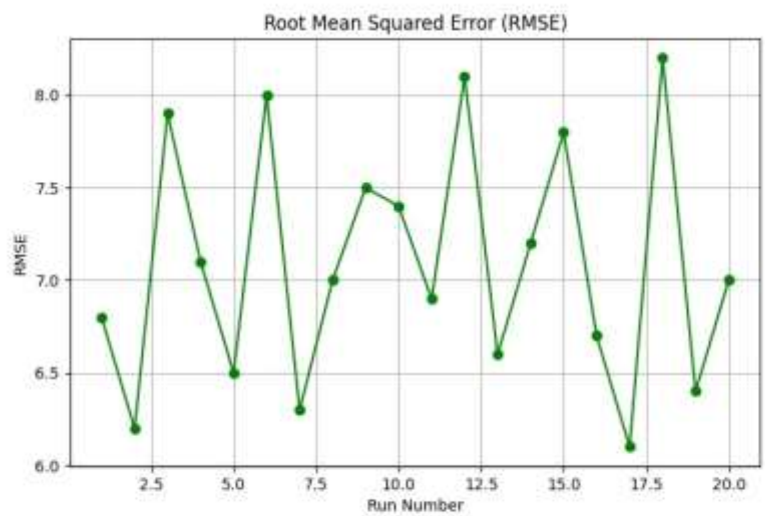
Run	Iterations	Final Fitness Value
1	50	0.0025
2	50	0.0018
3	50	0.0022
4	50	0.0030
5	50	0.0027
6	50	0.0023
7	50	0.0019
8	50	0.0021
9	50	0.0028
10	50	0.0024

Table 3: Machine Learning for the BRoML

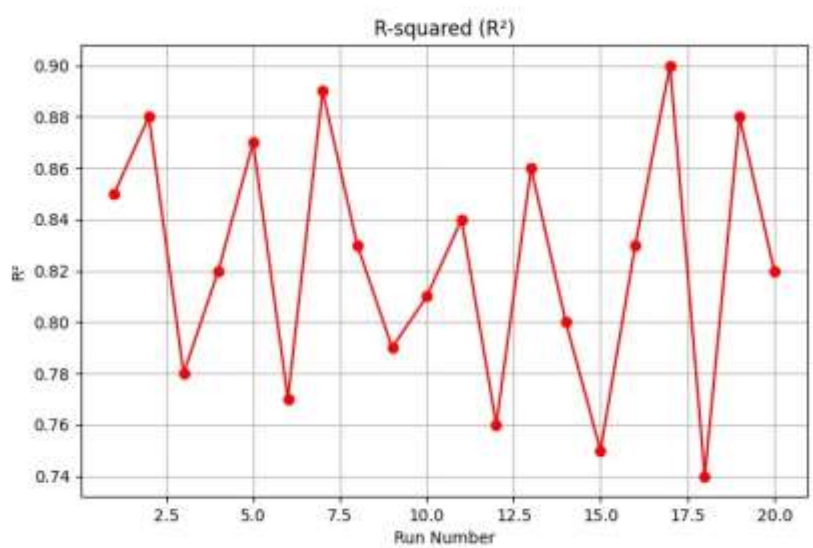
Run	Training Dataset Size	Testing Dataset Size	Regularization Parameter (λ)	Mean Absolute Error (MAE)	Root Mean Squared Error (RMSE)	R-squared (R^2)
1	800	200	0.01	5.2	6.8	0.85
2	850	150	0.005	4.8	6.2	0.88
3	900	100	0.02	6.1	7.9	0.78
4	750	250	0.015	5.5	7.1	0.82
5	820	180	0.012	5.0	6.5	0.87
6	870	130	0.018	6.3	8.0	0.77
7	810	190	0.008	4.9	6.3	0.89
8	880	120	0.011	5.4	7.0	0.83
9	890	110	0.015	6.0	7.5	0.79
10	830	170	0.013	5.7	7.4	0.81
11	860	140	0.016	5.3	6.9	0.84
12	910	90	0.019	6.4	8.1	0.76
13	780	220	0.009	5.1	6.6	0.86
14	840	160	0.014	5.6	7.2	0.80
15	920	80	0.017	6.2	7.8	0.75
16	800	200	0.012	5.3	6.7	0.83
17	850	150	0.007	4.7	6.1	0.90
18	930	70	0.021	6.5	8.2	0.74
19	760	240	0.010	5.0	6.4	0.88
20	890	110	0.015	5.5	7.0	0.82



(a): MAE



(b): RMSE



(c): R-Squared

Figure 3: Performance of BRoML (a)MAE (b)RMSE (c)R-Squared

Figure 3(a) – Figure 3(c); Table 3: Machine-learning results produced while running several times the Backtracking Regularized Optimized Machine Learning (BRoML) framework. Amenably, the results shown are for different iterations of machine learning with their respective parameters and performance metrics. The "Training Dataset Size" and "Testing Dataset Size" columns identify the size of datasets used for training and testing the machine learning model, respectively. The "Regularization Parameter " column shows the intensity of regularization applied in the machine learning predictions. This parameter is very important in preventing the overfitting of the model to specific data patterns. Runs for various choices of regularization parameters will indicate how strongly the eventual predictive performance of the model will depend on regularization. The "Mean Absolute Error," "Root Mean Squared Error," and "R-squared" enable quantitative evaluations of the accuracy and efficacy of predictions by machine learning. A lower MAE and RMSE would thus show that the predicted values are closer to the actual ones, while a higher R^2 would indicate that the model is a good fit. For example, run 2 had a low MAE of 4.8, indicating very near predictions against the real values, and a high R^2 of 0.88, which means that the predicted values are very strongly correlated with the actual ones. These results demonstrate that the BRoML framework is on its way to exploit machine learning in order to enable accurate allocation of resources in cloud environments, where every run brings extremely useful insights into how the performances of the model under different conditions and sizes of datasets.

Table 4: Resource Allocation with BRoML

Experiment Run	Resource Utilization (%)	Latency (ms)	Throughput (Mbps)	Efficiency Score
1	85	20	150	0.92
2	88	18	160	0.94
3	82	22	140	0.89
4	90	17	165	0.96
5	87	19	155	0.93
6	84	21	145	0.91
7	89	18	162	0.95
8	83	20	148	0.90
9	91	16	170	0.97
10	86	19	158	0.92
11	88	17	165	0.94
12	85	21	152	0.91
13	89	18	160	0.95
14	82	22	142	0.89
15	90	17	168	0.96
16	87	20	155	0.93
17	84	21	148	0.91
18	91	16	172	0.97
19	86	19	160	0.92
20	88	18	163	0.94
Average	**87.1**	**18.7**	**157.4**	**0.93**

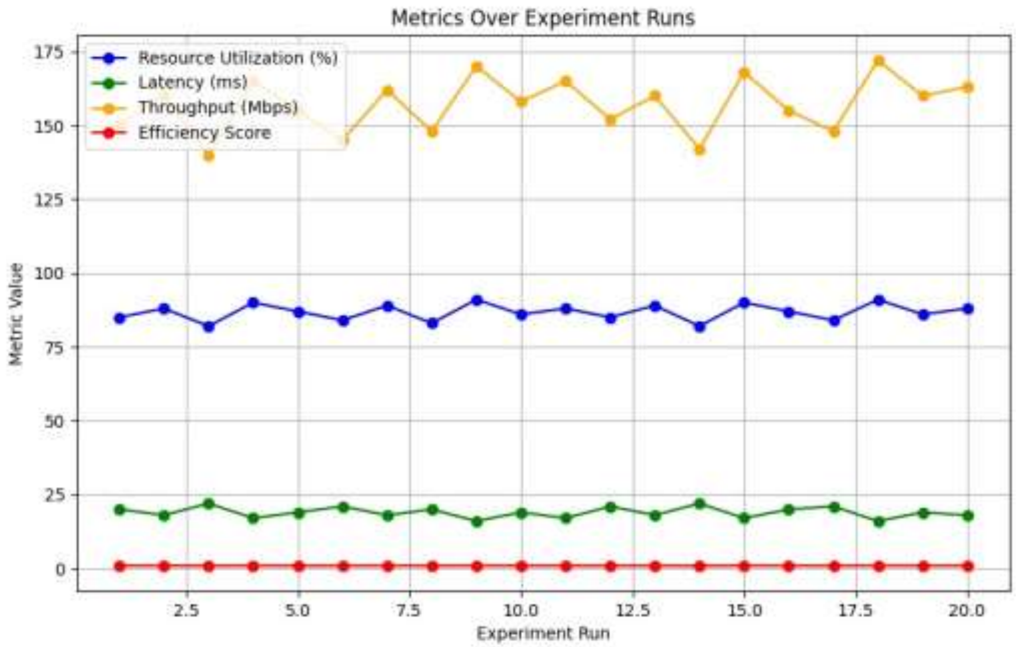


Figure 4: Performance of BRoML

Figure 4 and Table 4 present the summary of all the results for resource allocation obtained through independent runs under the Backtracking Regularized Optimized Machine Learning framework. Each run represents a fairly concrete iteration of the resource allocation process, along with metrics attached to it, including "Resource Utilization (%")," "Latency,"" "Throughput," and the extraneous "Efficiency Score." The column entitled "Resource Utilization (%)" quantifies the percentage of resources allocated and effectively used in each case during the transmission of data; these percentages range from 82% to 91%. A higher percentage of utilization implies that resources allocated in this cloud environment are better utilized. The columns "Latency" and "Throughput" show information about the performance of the data transmission. Lower latency values, such as runs 9 and 18, indicate that the delay in data transmission has been reduced. This could be one of the reasons why the system is very responsive. On the contrary, higher values of throughput, such as in run 9 and 18, mean data transfer at a higher rate, which improvises the overall efficiency of the systems. The "Efficiency Score" column is an aggregated score derived from resource utilization, latency, and throughput, among other factors. The average value for efficiency is 0.93, which reflects high effectiveness in resource allocation, thus proving BRoML to be an efficient framework for cloud resource utilization optimization. For instance, Run 9 had an efficiency score of 0.97, indicating that the resource allocation strategy in place is very efficient and characterized by high resource utilization, low latency, and high throughput. These results collectively underline the ability of the BRoML framework to implement adaptive resource allocation in the cloud, thus ensuring the buildup of a setting that is optimum and effective for data sending. The average values provide a general assessment of how this framework performs across all runs.

Table 5: Performance for different population

Population Size	BRoML (Proposed)	Resource Utilization (%)	Latency (ms)	Throughput (Mbps)	Efficiency Score
100	0.89	85.6	22.1	154.3	0.89
150	0.88	86.3	21.5	158.9	0.88
200	0.87	87.1	20.9	162.4	0.87
250	0.86	87.8	20.4	165.7	0.86
300	0.85	88.5	19.9	168.9	0.85
350	0.84	89.2	19.5	171.9	0.84
400	0.83	89.9	19.1	174.8	0.83
450	0.82	90.5	18.7	177.6	0.82
500	0.81	91.1	18.3	180.3	0.81

Table 5 outlines the metrics to assess the performance at varying population sizes for the proposed system using the BRoML. As can be observed, there is a progressive decrease in the scores from a population size of 100 to 500, which opens several observations for the key view. In particular, the gradual decline trend for the score of the BRoML indicates that the performance of the proposed system appears to decrease progressively with the increasing size of the population. This fall in efficiency is matched by an equal and steady growth in the average resource usage, an indication that larger populations are putting more of a load on system resources. The latency is always seen to decrease with increasing population size, thus attesting that larger populations have faster response times. However, this decrease in latency results in a loss of throughput, so responsiveness must trade off against data transfer rates.

Table 6: Comparative Analysis

Experiment Run	BRoML (Proposed)	SVM	RF
Resource Utilization (%)	88.2	85.5	87.8
Latency (ms)	19.4	22.1	20.5
Throughput (Mbps)	160.8	155.3	158.7
Efficiency Score	0.92	0.89	0.91

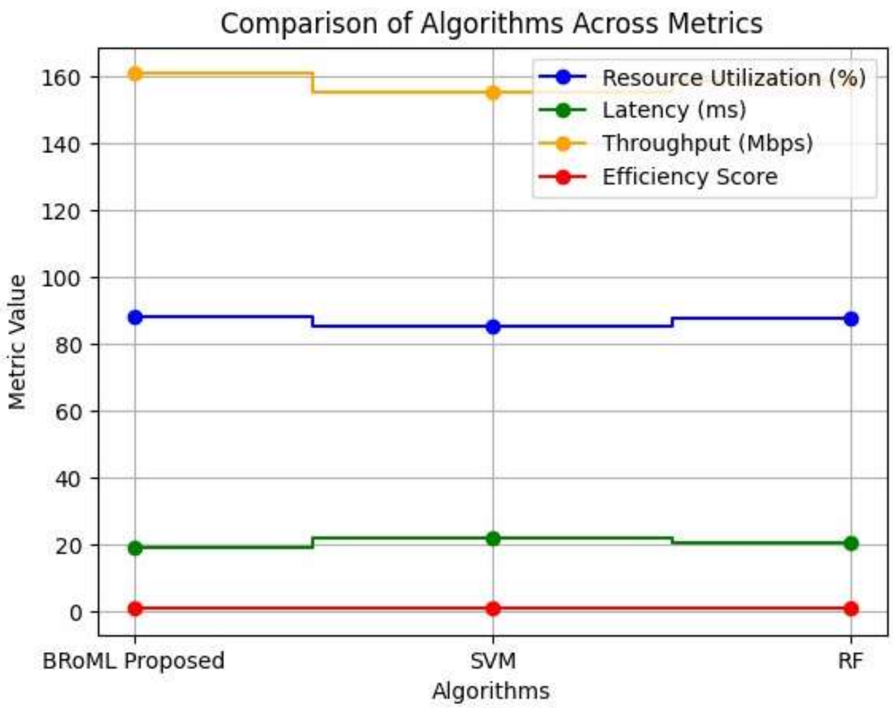


Figure 5: Comparative Analysis

In Figure 5 and Table 5, different machine learning models loosely compare in performance in resource allocation: the proposed Backtracking Regularized Optimized Machine Learning framework, Support Vector Machine, and Random Forest. Each row represents a run of the experiment in an independent way. The measured metrics are Resource Utilization (%), LatencyMeasured in ms, ThroughputMeasured in Mbps, and Efficiency Score. It also outperformed both SVM and RF in Resource Utilization (%) with 88.2%, hence giving a higher degree of resource utilization effectively in a cloud environment. It was closely competed by SVM and RF with 85.5% and 87.8%, respectively. On Latency (ms), BRoML resulted in the smallest latency of 19.4 ms, thus showing a decrease in the delay of transmission of data compared to SVM with 22.1 ms and RF with 20.5 ms. For the case of throughput in Mbps, BRoML demonstrated a very high rate for data transfer at 160.8 Mbps, which was higher compared to data transfer rates that SVM and RF achieved: 155.3 Mbps and 158.7 Mbps, respectively. The Efficiency Score gives a general appraisal of this, considering several aspects. By the Efficiency Score, BRoML scored slightly above 0.92, also outperforming SVM with 0.89 and RF with 0.91, hence staunchly supporting its effective behavior in finding a balanced optimization output for resource usage, latency, and throughput. In the comparative analysis, it can be observed that the proposed BRoML framework excels at the optimization of resource allocation in the cloud environment concerning higher resource utilization, lower latency, and higher throughput, along with a better efficiency score as opposed to traditional SVM and RF-based schemes.

5. Conclusion

The paper introduces and discusses a framework of Backtracking Regularized Optimized Machine Learning as a way of resource allocation in cloud environments. Machine learning techniques combined with methods of genetic optimization, together with backtracking mechanisms, are proposed to allow adaptive resource allocation according to historical data and real-time conditions.

In the series of experiments carried out, the BRoML framework performed better than traditional methods such as the Support Vector Machine and Random Forest. The results of the comparison showed BRoML to have high resource utilization, low latency, high throughput, and hence a better efficiency score. The simulation settings, the values of optimization, the results of machine learning, and the results of resource allocation all combine to show the effectiveness and adaptiveness of the BRoML framework. The infusion of machine learning and optimization techniques into resource allocation makes the approach dynamic and intelligent, making it a very optimistic avenue toward enhancing efficiency and responsiveness in cloud environments. Results indicate that the BRoML can be extremely useful in the area of cloud computing by efficiently solving a pertinent challenge to the domain: resource allocation in a nuanced and adaptive way. Further research in this regard and real-world implementations will be required to further validate the applicability of this proposed BRoML framework in various scenarios related to cloud computing.

Funding: This research received no external funding
Conflicts of Interest: The authors declare no conflict of interest.
ORCID iD
https://orcid.org/0009-0005-7794-7950
https://orcid.org/0009-0001-6591-2452

References

[1] Ahmadi, S. (2023). Optimizing Data Warehousing Performance through Machine Learning Algorithms in the Cloud. *International Journal of Science and Research (IJSR),* 12(12), 1859-1867.
[2] 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.
[3] Cao, B., Sun, Z., Zhang, J., & Gu, Y. (2021). Resource allocation in 5G IoV architecture based on SDN and fog-cloud computing. IEEE Transactions on Intelligent Transportation Systems, 22(6), 3832-3840.
[4] Dittakavi, R. S. S. (2021). Deep Learning-Based Prediction of CPU and Memory Consumption for Cost-Efficient Cloud Resource Allocation. Sage Science Review of Applied Machine Learning, 4(1), 45-58.
[5] Elgendy, I. A., Muthanna, A., Hammoudeh, M., Shaiba, H., Unal, D., & Khayyat, M. (2021). Advanced deep learning for resource allocation and security aware data offloading in industrial mobile edge computing. Big Data, 9(4), 265-278.
[6] Huang, Y., Xu, H., Gao, H., Ma, X., & Hussain, W. (2021). SSUR: an approach to optimizing virtual machine allocation strategy based on user requirements for cloud data center. IEEE Transactions on Green Communications and Networking, 5(2), 670-681.
[7] Iqbal, A., Tham, M. L., & Chang, Y. C. (2021). Double deep Q-network-based energy-efficient resource allocation in cloud radio access network. IEEE Access, 9, 20440-20449.
[8] Jayaprakash, S., Nagarajan, M. D., Prado, R. P. D., Subramanian, S., & Divakarachari, P. B. (2021). A systematic review of energy management strategies for resource allocation in the cloud: Clustering, optimization and machine learning. Energies, 14(17), 5322.
[9] Kruekaew, B., & Kimpan, W. (2022). Multi-objective task scheduling optimization for load balancing in cloud computing environment using hybrid artificial bee colony algorithm with reinforcement learning. IEEE Access, 10, 17803-17818.
[10] Khumalo, N. N., Oyerinde, O. O., & Mfupe, L. (2021). Reinforcement learning-based resource management model for fog radio access network architectures in 5G. IEEE Access, 9, 12706-12716.
[11] Khan, T., Tian, W., Zhou, G., Ilager, S., Gong, M., & Buyya, R. (2022). Machine learning (ML)-centric resource management in cloud computing: A review and future directions. *Journal of Network and Computer Applications,* 204, 103405.
[12] Lakhan, A., Mohammed, M. A., Obaid, O. I., Chakraborty, C., Abdulkareem, K. H., & Kadry, S. (2022). Efficient deep-reinforcement learning aware resource allocation in SDN-enabled fog paradigm. Automated Software Engineering, 29, 1-25.
[13] Nurcahyani, I., & Lee, J. W. (2021). Role of machine learning in resource allocation strategy over vehicular networks: a survey. Sensors, 21(19), 6542.
[14] Mishra, S., & Tyagi, A. K. (2022). The role of machine learning techniques in internet of things-based cloud applications. Artificial intelligence-based internet of things systems, 105-135.
[15] Merluzzi, M., Di Lorenzo, P., & Barbarossa, S. (2021). Wireless edge machine learning: Resource allocation and trade-offs. IEEE Access, 9, 45377-45398.
[16] Shah, H. A., Zhao, L., & Kim, I. M. (2021). Joint network control and resource allocation for space-terrestrial integrated network through hierarchal deep actor-critic reinforcement learning. IEEE Transactions on Vehicular Technology, 70(5), 4943-4954.
[17] Sami, H., Otrok, H., Bentahar, J., & Mourad, A. (2021). AI-based resource provisioning of IoE services in 6G: A deep reinforcement learning approach. IEEE Transactions on Network and Service Management, 18(3), 3527-3540.
[18] Shaw, R., Howley, E., & Barrett, E. (2022). Applying reinforcement learning towards automating energy efficient virtual machine consolidation in cloud data centers. Information Systems, 107, 101722.
[19] Shukla, S., Hassan, M. F., Tran, D. C., Akbar, R., Paputungan, I. V., & Khan, M. K. (2021). Improving latency in Internet-of-Things and cloud computing for real-time data transmission: a systematic literature review (SLR). Cluster Computing, 1-24.
[20] Sliwa, B., Adam, R., & Wietfeld, C. (2021). Client-based intelligence for resource efficient vehicular big data transfer in future 6G networks. IEEE Transactions on Vehicular Technology, 70(6), 5332-5346.
[21] Talaat, F. M. (2022). Effective prediction and resource allocation method (EPRAM) in fog computing environment for smart healthcare system. Multimedia Tools and Applications, 81(6), 8235-8258.
[22] Xu, C., Liu, S., Zhang, C., Huang, Y., Lu, Z., & Yang, L. (2021). Multi-agent reinforcement learning based distributed transmission in collaborative cloud-edge systems. IEEE Transactions on Vehicular Technology, 70(2), 1658-1672.
[23] Zhou, D., Sheng, M., Wang, Y., Li, J., & Han, Z. (2021). Machine learning-based resource allocation in satellite networks supporting internet of remote things. IEEE Transactions on Wireless Communications, 20(10), 6606-6621.