MATEC Web of Conferences **392**, 01155 (2024) https://doi.org/10.1051/matecconf/202439201155
ICMED 2024

Modernizing cloud computing systems with integrating machine learning for multi-objective optimization in terms of planning and security

Thirumalai Selvan[1*], S. *Siva* Shankar[2], S. *Sri* Nandhini Kowsalya[3], *Praseeda* Ravuri[4], *Deepak* Kumar Nayak[5], *Koppuravuri* Gurnadha Gupta[6], and M.N. *Sharath*[7]

[1]Department of Computer Science and Engineering, GITAM School of Technology, GITAM University -Bengaluru
[2]Department of CSE, KG Reddy College of Engineering and Technology, Chilukuru Village, Hyderabad -501504
[3]Vel Tech Multi Tech Dr. Rangarajan Dr. Sakunthala Engineering College, Chennai- 62
[4]Computer Science Engineer, Oregon State University, Corvallis, Oregon, USA 97331
[5]Department of ECE, Budge Budge Institute of Technology, Kolkata, West Bengal-700137, India
[6]Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur District, Andhra Pradesh - 522302, India
[7]Rajeev Institute of Technology, Hassan

Abstract. Cloud enterprises face challenges in managing large amounts of data and resources due to the fast expansion of the cloud computing atmosphere, serving a wide range of customers, from individuals to large corporations. Poor resource management reduces the efficiency of cloud computing. This research proposes an integrated resource allocation security with effective task planning in cloud computing utilizing a Machine Learning (ML) approach to address these issues. The suggested ML-based Multi-Objective Optimization Technique (ML-MOOT) is outlined below: An enhanced task planning, based on the optimization method, aims to reduce make-span time and increase throughput. An ML-based optimization is developed for optimal resource allocation considering various design limitations such as capacity and resource demand. A lightweight authentication system is suggested for encrypting data to enhance data storage safety. The proposed ML-MOOT approach is tested using a separate simulation setting and compared with state-of-the-art techniques to demonstrate its usefulness. The findings indicate that the ML-MOOT approach outperforms the present regarding resource use, energy utilization, reaction time, and other factors.

1 Introduction to cloud computing and resource allocation

Cloud computing is an innovative technology developed by leading data centers in the computer industry, which aids in advancing virtualization practices [1]. Proper handling is shown as a connection integrating software, infrastructure, and Platform as a Service (PaaS)

* Corresponding author: thirumalaiselvan027@outlook.com

[2]. Each individual has a unique perspective on business in general. Appropriate computing aims to construct a virtual infrastructure of computers, servers, and experts to produce applications that cater to customers, regardless of the acquisition model. Internet connection and equipment are crucial for the cloud based on computing power and communication. The network serves as a means for remote access and extra apps for several cloud-based applications. The cloud's Quality of Service (QoS) delivery system is interconnected with its infrastructure and competencies [3]. Many application service providers have recognized the difference between using and operating the necessary equipment and have opted to lease the required infrastructure from suppliers. Force Square has utilized Amazon EC2 Analysis for over 5 million days, resulting in a 53% cost savings to address specific requirements, establishing the first cloud resources measuring tool [4].

Resource allocation involves the scheduling and assignment of supplies [5]. Combining resource allocation with task scheduling helps minimize system latency. The virtualization system is used to allocate resources, such as inputting and output gadgets, the Central Processing Unit (CPU), and memories, by virtualizing them. Virtualization has led to enhancements in the structure, including reduced execution time, minimized energy consumption, and enhanced use of cloud resources in servers. Cloud computing scheduling issues mostly revolve around work allocation and server virtualization management [6]. Safety and energy efficiency are crucial factors in work scheduling, keeping data, and computation in contemporary cloud-based systems. Blockchain innovation offers a safe and efficient way to store data decentralized, eliminating the need for a central storage facility. This will enhance security and safeguard the database from assaults on the overall cloud infrastructure.

Scheduling and assigning resources in an integrated manner help minimize system latency. Scholars devised many methods for work scheduling systems. However, these approaches face challenges such as handling energy, server reorganization, privacy, information administration, controlling automated functions, and traffic concerns, which primarily impact the efficiency, protection, and accessibility of cloud computing. This research suggests a new method for efficient task planning with improved security in a cloud computing setting.

This research study makes the following features.

- This research proposed a Machine Learning (ML) based Multi-Objective Optimization Technique for task scheduling in cloud computing.
- Develop a task scheduling system using a Convolutional Neural Network (CNN) method to reduce reaction time and enhance resource usage.
- To evaluate the suggested approach with several task scheduling-based methods for performance indicators such as resource usage, response duration, and energy use.

The remaining sections are listed as follows: section 2 deals with the literature survey on cloud computing and resource optimization. The proposed ML-based Multi-Objective Optimization Technique (ML-MOOT) is discussed and analyzed in section 3. The experimental results of the proposed ML-MOOT are shown in Section 4. Section 5 completes the research with a conclusion and future scope.

2 Literature survey

Hosseini Shirvani et al. introduced an asset dispersion model that considers different asset valuations for several service providers and diverse asset allocations concurrently, resulting in increased benefits [7]. The analysis indicates that the evaluated cost closely aligns with the actual exchange cost, which is somewhat lower than the true value. The procedure is comparative for sensing-perceiving and intuitive types. They will update the application foundation for upcoming tasks and adjust the parameters to enhance efficiency.

Javaid et al. introduced a distributed computer system that provides asset allocation and appraisal and facilitates a feasible transaction based on customer feedback and characteristics [8]. Depending on the payment type, customers submit several requests simultaneously. They handle many requests, including one referred to as an ambiguous presentation.

Gopu et al. suggested using the Virtual Machine (VM) consolidation asset allocation calculation to enhance energy efficiency and reduce data center management complexity by considering fragments, upgrade frequency, and transportation route length [9].

Hosseini et al. introduced a resource-based task control method to manage changing weight and resource needs [10]. It enables different sources to address various instabilities resulting from diverse impediments. It provides unified support to connect and guarantee that QoS does not provide enough aid for a single service.

Beniwal et al. explored a predictive method for configuration recommendations using Genetic Algorithms (GA) and Support Vector Regression (SVR) [11]. This tool combines idle time calculations with suggestions for the best setup of cloud resources based on time and cost.

Güçyetmez et al. suggested conducting load and resource allocation experiments on a compact C-Ran running to enhance single-phase energy utilization [12]. A conventional mixed non-software application was created to strengthen load unloading outcomes, scheduling resources, and radio allocation of resources.

Praveenchandar et al. suggested an energy-efficient strategy for efficient planning and distribution of resources [13]. A forecasting method and variable resource update algorithm were used to allocate resources to carry out tasks and manage response time. This method is beneficial for optimizing energy efficiency in data centers.

Qu et al. introduced the Integrative Federated Model (InFeMo) to combine several cloud models inside a federated learning setting and other relevant technologies that are used together, providing a new integrated situation [14]. The suggested approach aims to provide consumers with a more energy-efficient system structure and setting.

When organizing, the planner should consider several obstacles, such as the nature of the project, the scale of the task, the time required to complete it, the availability of resources, the sequence of the task, and the scheduling. Task scheduling is a crucial concern in cloud computing. Efficient resource use results from the proper organization of functions.

3 ML-based multi-objective optimization technique

This research aims to provide an efficient task-scheduling approach that minimizes reaction time and energy use while maximizing the use of resources. Figure 1 illustrates the procedure of the suggested ML-MOOT system, which includes different task managers dividing the work into many clusters. This study suggests using a CNN-optimized method for task planning and modified RSA to secure data in online storage.

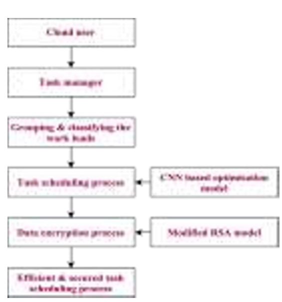

Fig. 1. Workflow of the ML-MOOT system

3.1 Task scheduling process

Task scheduling can be improved by considering many factors from numerous vendors, including user requests, task types, and dependencies. The user demand was initially established with 1 to N task units. The task category is therefore defined as consisting of 1 to t tasks. t_{max} represents the total quantity of tasks inside the task component. The task dependence (t_s) obtained through the task unit is denoted as t_s^{ij} in Eqn. (1).

$$t_s^{ij} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad (1)$$

The components show the interdependencies of task units. Thus, the research used the ML-MOOT method to improve work scheduling efficiency in a cloud computing setting. It improves preparation efficiency and minimizes time wastage.

3.2 Convolutional neural network

CNNs differ in gathering data and processing. CNNs typically consist of convolutional, pooling, and fully interconnected tiers. Convolutional tiers use cores that move horizontally and vertically over the input data. The output is generated by combining each weight in the kernel with the associated element in the input matrix that the kernel is applied to. The result is added together to provide the outcome. Throughout the training phase, the kernel's weights are changed via backpropagation. Pooling tiers are employed to decrease the complexity of characteristic maps and enhance the robustness of feature identification. CNNs have been less often used for task planning issues than other neural networks.

3.2.1 Convolutional tier

The convolutional tier is a fundamental component of CNN. The tiers are thinner and include filters that cover the whole picture via shifting. The convolutional process occurs by computing the dot product between the filter and the image itself. The dot products between the filter and the picture are consolidated inside the filtering area in Eqn. (2).

$$a_k^n = \alpha\left(y_j^{n-1} x_{jk}^n + c_k^n\right) \quad (2)$$

3.2.2 Pooling tier

The pooling tier performed downsampling. Various categories of the pooling operation are available. The most often-used algorithm is maximal pooling. The maximal pooling filtering provided the maximum values for each subregion. They are using 2 × 2 × 1 maximum pooling filtering on a 4 × 4 × 1 size characteristic results in a 2 × 2 × 1 size pattern.

3.2.3 Fully connected tier

As shown in Eqn. (3) the neurons in the previous tier are linked to every neuron in the following levels in the completely interconnected tier.

$$z_k^n = \alpha\left(a_j^{n-1} x_{jk}^n + c_k^m\right) \quad (3)$$

The bias is written as c_k^m, while the input from the preceding tier is expressed as a_j^{n-1}.. Sparseness in the concealed tier of the fully interconnected tier helps prevent overfitting in CNN by addressing the issue of k excessive fitting.

3.2.4 Softmax function tier

The softmax function tier calculates the distributions of probabilities of different occurrences. The functioning of the softmax tier is computed and presented in Eqn. (4).

$$Q(z_k^n) = \frac{\exp(z_k^n)}{\sum_{k=0}^{n-1} \exp(z_k^n)} \tag{4}$$

3.2.5 Classification output tier

The loss is calculated during learning the output tier of the CNN. The objective of CNN is to minimize the cost function to make accurate predictions on the information. The current cost function is computed and represented in Eqn. (5).

$$e^E(i, c) = c_e + \beta \sum_{i=0}^{n-1} i^2 \tag{5}$$

The cross-entropy is represented in Eqn. (6).

$$c_e = -\sum_{i=0}^{n-1} z_i^n \log(z_i^U) \tag{6}$$

The target denoted as z_i^n and the prediction value characterized as z_i^U.

3.3 CNN-optimized approach for task scheduling

Figure 2 shows the flow architecture of the suggested CNN-optimized MBO technique for an efficient task-planning procedure. The input database is first inputted into the neural network to gather essential data characteristics. The data in CNN is transformed into vectors before being sent to a fully linked tier. Because of its high level of training, the complex, fully interconnected tier requires many variables, leading to overfitting issues. The modified optimization technique ensures continuous depiction and improves the weight value. The ML-MOOT method efficiently improves the task-scheduling procedure.

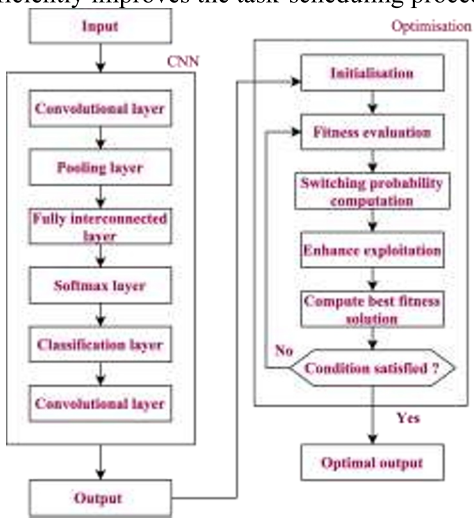

Fig. 2. Optimization model for the task scheduling

4 Simulation analysis and findings

This section outlines the investigation to assess the effectiveness of the suggested ML-MOOT. Metrics are calculated, including resource use, response time, and energy usage. The ML-MOOT model is being tested in a cloudlet simulation. The machine has a Windows 64-

bit operating system, an Intel Pentium CPU clocked at 3.00 GHz, 4 GB of RAM, and a 1 TB hard drive. The methods used include a learning rate of 0.005, a Sigmoid activation operation, and an abandonment rate of 0.5. The model consists of 7 hidden tiers with a batch size of 32, and the training process involves three epochs for the CNN.

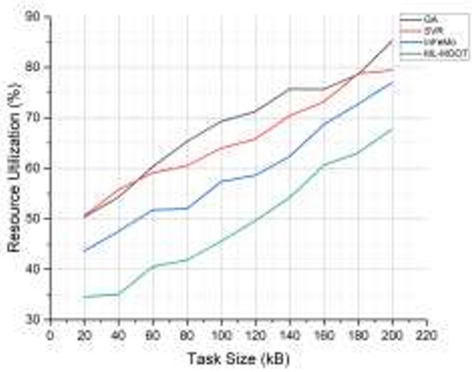

Fig. 3. Resource utilization analysis of ML-MOOT

Figure 3 displays the impact of varying task sizes (ranging from 20 to 200kB with a step size of 20kB) on Resource Utilization (%) using various algorithms such as GA, SVR, InFeMo, and ML-MOOT. The measure is calculated by analyzing the proportion of resources used for each task size and approach. ML-MOOT has superior performance in resource usage compared to GA, SVR, and InFeMo, achieving an impressive 42.54%. ML-MOOT's remarkable excellence stems from its skillful use of multi-objective optimization, which optimizes resource allocation and enhances efficiency across various task sizes.

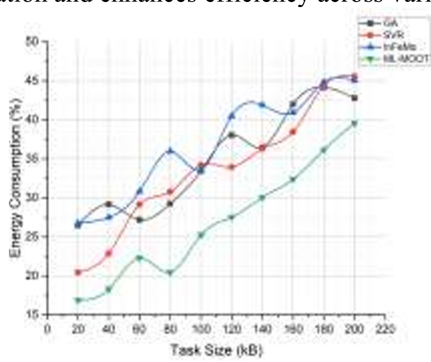

Fig. 4. Energy consumption analysis of ML-MOOT

Figure 4 displays the Energy Consumption (%) findings for multiple algorithms (GA, SVR, InFeMo, and ML-MOOT) over various task sizes (range from 20 to 200kB with a step size of 20kB). The measure is calculated by analyzing the proportion of energy used for each task size and approach. ML-MOOT excels in energy efficiency, using an average of 22.48%, surpassing GA (35.48%), SVR (35.82%), and InFeMo (35.92%). The success of ML-MOOT is due to its skillful multi-objective optimization method, which efficiently reduces energy consumption for various task sizes, thereby improving the overall efficiency of the cloud computing system.

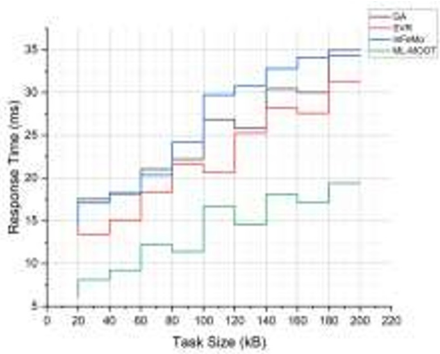

Fig. 5. Response time analysis of ML-MOOT

Figure 5 displays the performance of multiple approaches (GA, SVR, InFeMo, and ML-MOOT) in terms of Response Time (ms) for various task sizes (ranging from 20 to 200kB with a step size of 20kB). The statistic, derived from the millisecond reaction time for each task size and approach, highlights the efficiency of ML-MOOT. ML-MOOT demonstrates notably reduced reaction times, averaging 11.14% and outperforming GA (25.81%), SVR (26.64%), and InFeMo (27.09%). ML-MOOT's competent multi-objective optimization minimizes reaction times for various task sizes, thereby improving the overall responsiveness of the cloud computing system.

5 Conclusion and future study

The article introduced a new CNN, the ML-MOOT method, designed to reduce reaction time and increase resource usage. A revised RSA technique encrypts the data, ensuring safe data transfer. The suggested method is tested using a cloudlet simulator, and outcomes are examined to assess its efficacy. Resource usage, reaction time, and energy consumption metrics are calculated to determine the efficiency of the ML-MOOT model. The suggested work has low energy consumption and little reaction time. The proposed ML-MOOT model achieves a lower resource utilization efficiency for a task size 200, surpassing GA, SVR, and InFeMo. The task's time of completion is postponed. In the future, the suggested method will be used in real-time to assess the system's efficiency and improve QoS using task scheduling methods that consider task arrival times.

References

1. A. Katal, S. Dahiya, T. Choudhury. *Energy efficiency in cloud computing data centers: a survey on software technologies.* Cluster Computing, **26**, 3, 1845-1875, (2023)
2. M.G. Lanjewar, K.G. Panchbhai. *Convolutional neural network-based tea leaf disease prediction system on the smartphone using PaaS cloud.* Neural Computing and Applications, **35**, 3, 2755-2771, (2023)
3. P. Asghari, A.M. Rahmani, H.H.S. Javadi. *Privacy-aware cloud service composition based on QoS optimization in the Internet of Things.* Journal of Ambient Intelligence and Humanized Computing, **13**, 11, 5295-5320, (2022)
4. S. Pathak, S.A. Islam, H. Jiang, L. Xu, E. Tomai, (2022). *A survey on security analysis of Amazon Echo devices.* High-Confidence Computing, (2022)
5. A. Belgacem. *Dynamic resource allocation in cloud computing: analysis and taxonomies.* Computing, **104**, 3, 681-710, (2022)

6. A. Pradhan, S.K. Bisoy, A. Das. *A survey on PSO-based meta-heuristic scheduling mechanism in a cloud computing environment.* Journal of King Saud University-Computer and Information Sciences, **34**, 8, 4888-4901, (2022)
7. M. Hosseini Shirvani, G.R. Amin, S. Babaeikiadehi (2022). *A decision framework for cloud migration: A hybrid approach.* IET Software, **16**, 6, 603-629, (2022)
8. M. Javaid, A. Haleem, R.P. Singh, R. Suman, S. Khan. *A review of Blockchain Technology applications for financial services.* Bench Council Transactions on Benchmarks, Standards and Evaluations, (2022)
9. A. Gopu, K. Thirugnanasambandam, A.S. AlGhamdi, S.S. Alshamrani, K. Maharajan, M. Rashid. *Energy-efficient virtual machine placement in the distributed cloud using NSGA-III algorithm.* Journal of Cloud Computing, **12**, 1, 1-20, (2023)
10. Y. Hosseini, R.K. Mohammadi, T.Y. Yang. *Resource-based seismic resilience optimization of the blocked urban road network in the emergency response phase considering uncertainties.* International Journal of Disaster Risk Reduction, **85**, (2023)
11. M. Beniwal, A. Singh, N. Kumar. *Forecasting long-term stock prices of global indices: A forward-validating Genetic Algorithm optimization approach for Support Vector Regression.* Applied Soft Computing, (2023)
12. M. Güçyetmez, H.S. Farhan. *Enhancing smart grids with a new IoT and cloud-based smart meter to predict energy consumption with time series.* Alexandria Engineering Journal, **79**, 44-55, (2023)
13. W. Hua, P. Liu, L. Huang. *Energy-efficient resource allocation for heterogeneous edge-cloud computing.* IEEE Internet of Things Journal, **11**, 2, 2808-2818, (2023)
14. L. Qu, Y. Wang, J. Yang, M. Zhao. *A heterogeneous network structure publishing security framework based on cloud-edge collaboration.* Computer Networks, **234**, (2023)