OPTIMIZATIONS IN APPLIED MACHINE LEARNING
Research Article* | *Volume 5* | *Issue 5* | *May 2025
Received: 08 Apr 2025 | Revised: 29 Apr 2025
Accepted: 16 May 2025 | Published Online: 24 May 2025

Autonomous Cloud Resource Management through DBSCAN-based unsupervised learning

Qinyi Zhu *

Indiana University, 107 S Indiana Ave, Bloomington, IN 47405, USA

*Corresponding Author, Email: qinyzhu@iu.edu

Abstract: Autonomous resource management in cloud computing is crucial for optimizing performance and resource utilization. Current research primarily focuses on supervised learning techniques, which require labeled data and manual intervention. However, unsupervised learning methods have the potential to autonomously adapt to dynamic cloud environments without the need for prior training data. In this context, this paper proposes a novel approach utilizing DBSCAN-based unsupervised learning for autonomous cloud resource management. This innovative method aims to cluster cloud resources based on their utilization patterns, enabling proactive resource allocation and dynamic scaling. By leveraging unsupervised learning, our approach addresses the challenges of scalability and real-time resource management in cloud environments, contributing to the advancement of autonomous cloud computing systems.

Keywords: *Autonomous; Resource Management; Cloud Computing; Unsupervised Learning; DBSCAN*

1. Introduction

Autonomous Cloud Resource Management is a research field dedicated to developing automated systems and algorithms for optimizing resource allocation in cloud computing environments without human intervention. Current challenges and bottlenecks in this area include the complexity of cloud infrastructures, the dynamic nature of workloads, the need for real-time decision-making, the security and privacy concerns associated with autonomous systems, and the lack of standardized approaches for autonomous resource management. Researchers in this field are actively working to overcome these obstacles through innovations in machine learning, artificial intelligence, decentralized systems, and policy-based management strategies to achieve more efficient, cost-effective, and reliable cloud resource management solutions.

To this end, research on Autonomous Cloud Resource Management has advanced to the stage where machine learning algorithms and AI techniques are being integrated to optimize resource allocation, enhance scalability, and improve efficiency in cloud computing environments. In the literature review, there are several key studies discussing different aspects of cloud resource management. Zaker et al. [1] proposed a formally verified scalable look ahead planning for cloud resource management, demonstrating the elasticity and flexibility of their autonomic manager in various cloud applications. Cho and Baffier [2] introduced an autonomous resource management model for edge clouds, focusing on resource allocation based on load and cost functions. Xia [3] designed a cloud-based human resource management model using big data technology to enhance HR functions such as recruitment and performance evaluation. Dong [4] presented an agent-based cloud simulation model for resource management, evaluating resource allocation strategies and their impact on energy consumption and resource utilization in heterogeneous clouds. Bucur and Miclea [5] discussed multi-cloud resource management techniques for cyber-physical systems, emphasizing the importance of managing resources for complex software projects like autonomous vehicles. Xiu et al. [6] proposed a task-driven computing offloading and resource allocation scheme for maritime autonomous surface ships, highlighting the importance of efficient resource allocation in a cloud–shore–ship collaboration framework. Hasan et al. [7] addressed computational offloading and resource management in vehicular edge computing, focusing on federated learning for better resource utilization while maintaining security and privacy. Lastly, Liu et al. [8] developed a data-connector framework for autonomous smart management in the cloud-edge continuum, showcasing the benefits of ML-based decision-making for resource adaptation scenarios. These studies collectively contribute to the advancement of cloud resource management through innovative models, algorithms, and frameworks. DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a popular clustering algorithm in data mining for its ability to effectively identify clusters of varying shapes and sizes, especially in noisy and large datasets. This technique is preferred in cloud resource management studies to efficiently group resources based on their proximity and density, enabling better resource allocation strategies and optimizing energy consumption. Its adaptability to different data distributions and robustness against outliers make DBSCAN a valuable tool for improving resource utilization and performance in complex cloud environments.

Specifically, DBSCAN, as a clustering algorithm, plays a crucial role in Autonomous Cloud Resource Management by efficiently identifying groups of cloud resources based on their similarities. This enables automated resource allocation and optimization strategies to enhance cloud system performance and scalability. The literature review on DBSCAN clustering algorithms provides insights into various advancements in the field. DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a crucial algorithm for unsupervised machine learning due to its capability to cluster datasets with different densities, shapes, and sizes, without requiring the predefined number of clusters [9]. An improved version, Enhanced DBSCAN-based Histogram (EDBSCAN-H), addresses challenges of DBSCAN in processing satellite images by incorporating a histogram-based approach [10]. Additionally, researchers have explored variations like MDBSCAN, a multi-density DBSCAN based on relative density, and DBSCAN Revisited, Revisited, which aim to enhance the performance of the original algorithm [11] [12]. Furthermore, new approaches like Block-Diagonal Guided DBSCAN Clustering leverage the block-diagonal

property of similarity graphs to improve clustering outcomes, and Anomaly Detection Based on GCNs and DBSCAN combines Graph Convolutional Networks with DBSCAN for efficient anomaly detection on graphs [13] [14]. Finally, a survey of DBSCAN clustering algorithms for big data highlights the advancements and provides a comprehensive comparison among the algorithms [15,16]. However, current limitations in DBSCAN clustering algorithms include scalability issues with large datasets, sensitivity to parameter settings, and challenges in handling high-dimensional data effectively.

To overcome those limitations, this paper aims to explore and develop a novel approach to autonomous resource management in cloud computing by utilizing DBSCAN-based unsupervised learning techniques. The primary objective of this research is to address the challenges of scalability and real-time resource management in dynamic cloud environments without the need for labeled data or manual intervention. By focusing on unsupervised learning methods, this approach seeks to autonomously adapt to changing cloud conditions and optimize performance and resource utilization. The proposed method involves clustering cloud resources based on their utilization patterns, enabling proactive resource allocation and dynamic scaling. By leveraging the capabilities of unsupervised learning, the approach aims to contribute to the advancement of autonomous cloud computing systems by offering a more flexible and efficient resource management solution. This research highlights the potential for unsupervised learning techniques to enhance the autonomy and adaptability of cloud computing systems, ultimately improving overall performance and resource utilization in cloud environments.

Section 2 of this study presents the problem statement focusing on autonomous resource management in cloud computing to optimize performance and resource utilization. In Section 3, the proposed method is introduced, highlighting the utilization of DBSCAN-based unsupervised learning for autonomous cloud resource management. Section 4 delves into a detailed case study, showcasing the application and efficacy of the novel approach. Analysis of results in Section 5 demonstrates the benefits of utilizing unsupervised learning methods in dynamic cloud environments. Section 6 engages in a discussion addressing scalability and real-time resource management challenges. Finally, Section 7 concludes the study, emphasizing the contribution of the innovative approach to advancing autonomous cloud computing systems through proactive resource allocation and dynamic scaling based on utilization patterns.

2. Background

2.1 Autonomous Cloud Resource Management

Autonomous Cloud Resource Management (ACRM) refers to the self-directed oversight of computing resources in a cloud environment. It leverages a set of intelligent strategies to dynamically allocate, optimize, and manage resources such as CPU, memory, storage, and network bandwidth. The aim is to achieve performance goals while minimizing costs and meeting Service Level Agreements (SLAs). At its core, ACRM involves a decision-making process that relies on real-time monitoring and predictive models. These models allow the system to anticipate demand and adapt resource allocation without human intervention. A fundamental concept underpinning ACRM is the notion of elasticity, which involves scaling resources up and down based on workload

demands. A central challenge in ACRM is optimally determining the quantity of resources $r(t)$ allocated at a given time t . This can be mathematically expressed as:

$$r(t) = \text{argmin}_{r_t}(C(r_t) + P(r_t, d_t)) \tag{1}$$

where $C(r_t)$ represents the cost function associated with the resources, and $P(r_t, d_t)$ is a penalty function tied to demand d_t and allocation r_t . The penalty function encompasses SLA violations or performance degradations that might occur if resources are improperly managed. To anticipate future resource needs, ACRM employs predictive analytics. A common approach is the use of time-series forecasting methods, such as ARIMA or machine learning models like recurrent neural networks (RNNs). The predicted demand $\hat{d}(t+k)$ for a future time period $t+k$ can be expressed as:

$$d(t+k) = f\big(d(t-\tau), \ldots, d(t)\big) \tag{2}$$

where f denotes the forecasting function based on past observations. To measure the efficiency of resource allocation, utility functions are often implemented. A typical utility function $U(r_t, d_t)$ considers both performance and cost:

$$U(r_t, d_t) = w_1 \cdot S(r_t, d_t) - w_2 \cdot C(r_t) \tag{3}$$

where $S(r_t, d_t)$ is a satisfaction function measuring SLA fulfillment, $C(r_t)$ is the cost function as earlier defined, and w_1 , w_2 are weighting factors balancing the two objectives. Resource optimization in ACRM can be represented by a constrained optimization problem aiming to maximize utility while ensuring constraints such as budget or capacity are respected:

$$\max_{r_t} U(r_t, d_t) \text{subject to:} r_{\min} \le r_t \le r_{\max} \tag{4}$$

where $r_{\min}$ and $r_{\max}$ define the minimum and maximum bounds of resource allocation. Moreover, scheduling techniques play a pivotal role in ACRM, where tasks are assigned to resources to optimize performance indices. The scheduling problem can be formulated as follows:

$$\min_{\pi} \sum_{i=1}^{n} L\left(f_t^{(i)}, d^{(i)}\right) \tag{5}$$

where π is a scheduling policy, n the number of tasks, $f_t^{(i)}$ the finish time, and $d^{(i)}$ the deadline for task i . The loss term L quantifies the deviation from task deadlines. Finally, the incorporation of feedback loops is vital in ACRM systems. Through continuous monitoring, the system self-adjusts by learning from past experiences, reducing errors, and adapting to unforeseen changes in demand. A feedback function can be denoted as:

$$r_{t+1} = r_t + \alpha \cdot (d_t - r_t) \tag{6}$$

where α is a learning rate indicating the adjustment magnitude based on observed deviation. In conclusion, ACRM represents a sophisticated area of research integrating machine learning,

optimization theory, and systems engineering to autonomously manage cloud resources in an efficient and cost-effective manner.

2.2 Methodologies & Limitations

Autonomous Cloud Resource Management (ACRM) leverages several contemporary methodologies for resource allocation, predictive management, and optimization. Nevertheless, while these approaches significantly enhance efficiency, they present inherent challenges. One prominent technique in ACRM is the application of machine learning models, especially reinforcement learning, to guide autonomous decisions. In reinforcement learning, the cloud environment is modeled as a Markov Decision Process (MDP), where the agent (the ACRM system) aims to learn an optimal policy that maximizes cumulative reward. The reward function $R(s_t, a_t)$ is linked to the state s_t of the environment and the action a_t taken, commonly related to cost efficiency and SLA adherence:

$$R(s_t, a_t) = U(r_t, d_t) - \lambda \cdot V(r_t, d_t) \tag{7}$$

Here, $U(r_t, d_t)$ is the utility function, and $V(r_t, d_t)$ represents a penalty for SLA violations, with λ being a trade-off parameter. Despite the efficacy of these models, a significant deficiency is the requirement for vast computational resources and data for training, which is not always feasible. Moreover, exploration in reinforcement learning might lead to suboptimal resource allocation during early stages, affecting performance. A prevalent approach to handle dynamic resource allocation is Linear Programming (LP) or Integer Programming (IP). These methods formulate the problem as an optimization model:

$$\min C(r_t) + P(r_t, d_t) \tag{8}$$

subject to capacity and SLA constraints. While LP/IP are computationally efficient for real-time applications, they assume linear relationships and discrete decisions, potentially oversimplifying the complexity of cloud environments. To bridge prediction inaccuracies, time-series forecasting techniques like ARIMA (AutoRegressive Integrated Moving Average) and more complex methods like LSTMs (Long Short-Term Memory networks) are employed. The accuracy of such predictions is critical:

$$d\ (t+k) = \sum_{i=1}^{p} \phi_i d(t-i) + \epsilon_t \tag{9}$$

where ϕ_i are the AR parameters, p is the number of lag observations, and ϵ_t denotes noise. Although these models capture temporal dependencies, their performance heavily depends on historical data quality and may not adapt swiftly to abrupt changes. Another strategic element is multi-objective optimization, which considers multiple criteria like cost, latency, and energy efficiency. The Pareto front is a popular solution technique:

$$\min \big(C(r_t), E(r_t), L(r_t)\big) \tag{10}$$

The introduction of multi-objective optimization necessitates complex decision-making frameworks, possibly increasing computational overhead and decision latency. Feedback control systems are crucial in ACRM to maintain system stability. A basic proportional feedback mechanism is illustrated through:

$$r_{t+1} = r_t + K_p \cdot (d_t - r_t) \tag{11}$$

where K_p is the proportional gain. Although feedback control can enhance responsiveness, it risks overshooting and oscillations if not carefully tuned, thereby complicating the balancing act between response speed and system stability. Furthermore, ACRM systems often rely on heuristic algorithms like Genetic Algorithms or Ant Colony Optimization for scheduling and load balancing:

$$\max \sum_{i=1}^{n} W(i) \cdot \Delta t^{(i)} \tag{12}$$

with $W(i)$ as task weights and $\Delta t^{(i)}$ as processing time differentials. Despite providing near-optimal solutions, these heuristics may lack the rigor of conventional optimization methods, leading to efficiency compromises under different scenarios. Overall, while current methodologies in ACRM offer robust solutions, their effectiveness is contingent on continuous adaptation to emerging technological landscapes and evolving computational challenges. Addressing these limitations will require ongoing research and development to refine models, optimize algorithms, and enhance system responsiveness.

3. The proposed method

3.1 DBSCAN

DBSCAN, or Density-Based Spatial Clustering of Applications with Noise, is an unsupervised machine learning algorithm inherently designed to identify clusters of varying shapes and sizes in data characterized by noise. Unlike other clustering algorithms such as K-means, which partition data into a predetermined number of clusters, DBSCAN offers a more flexible approach by identifying regions densely packed with data points separated by regions of lower density. This characteristic endows DBSCAN with an advantageous versatility in cluster detection within complex datasets, expanding its utility across diverse domains. The algorithm operates based on two key parameters: ϵ (eps), specifying the maximum radius of the neighborhood around a point, and $minPts$, denoting the minimum number of points required to form a dense region. Central to DBSCAN's methodology is the notion of "density reachability" and "density connectivity." Density reachability implies a point is reachable from another given enough density connectivity, which is established through a shared chain of neighboring points. Mathematically, DBSCAN can be formalized considering a dataset D composed of n points. The ϵ -neighborhood of a point p in D is denoted as:

$$N_\epsilon(p) = q \in D \mid \text{distance}(p, q) \leq \epsilon \tag{13}$$

where distance is typically measured using Euclidean distance, though other distance metrics are applicable depending on the data nature. For a point p , if $|N_\epsilon(p)| \geq minPts$, p is regarded as

a core point. The process of forming clusters begins with identifying these core points as they are the nucleus around which other points congregate. Non-core points, which form the border of a cluster, adhere to at least one core point. Points are grouped when one point is density-reachable from another. Hence, if there exists a sequence $p_1, p_2, \ldots, p_k$ with $p_1 = p$ and $p_k = q$ where each p_{i+1} is in $N_\epsilon(p_i)$ for all i, q is density-reachable from p. This property ensures clusters are connected within their bounds. The notion of density connectivity further extends that if point a is density-connected to b, and b is density-connected to c, then a is density-reachable from c, explained as:

$$\forall a, b, c: (\text{density-reachable}(a, b) \wedge \text{density-reachable}(b, c)) \Rightarrow \text{density-reachable}(a, c) \quad (14)$$

DBSCAN performs clustering effectively by iteratively progressing through each point $p \in D$, identifying the connected components of dense regions as clusters. A point is labeled as noise, or an outlier, if it fails to achieve membership status across any established dense region. The Euclidean distance metric frequently used is defined by:

$$\text{distance}(p, q) = \sqrt{\sum_{i=1}^{m} (p_i - q_i)^2} \quad (15)$$

where m represents the dimensionality of the data points. Moreover, the adjustable parameters ϵ and $minPts$ play critical roles in determining cluster granularity, as they filter out noise and delineate cluster boundaries. A smaller ϵ could yield many small clusters, while a larger value might merge data points into fewer, larger clusters:

$$C(\epsilon, minPts) = c \mid |c| \geq minPts \wedge \forall p \in c, |N_\epsilon(p)| \geq minPts \quad (16)$$

Despite its robustness, DBSCAN assumes relatively stable density across clusters, which may compromise accuracy in datasets with varying density. Additionally, defining an optimal ϵ remains a challenge, often requiring heuristic tuning or domain expertise to ensure that noise points are not mistakenly included in clusters. The precision and reliability of DBSCAN's clustering hinge significantly on accurate calibration of its parameters in congruence with dataset characteristics, paving a path for expansive application while underscoring the ongoing pursuit for enhanced adaptive clustering strategies.

3.2 The Proposed Framework

In the context of Autonomous Cloud Resource Management (ACRM), the integration of the DBSCAN clustering algorithm provides a robust mechanism for enhancing resource allocation strategies based on dynamically evolving demand patterns. ACRM, which autonomously oversees cloud computing resources, requires not only accurate predictive analytics but also effective clustering techniques to identify patterns that can inform resource allocation decisions in real-time. DBSCAN, with its density-based approach, allows for the identification of various resource utilization patterns in the cloud environment. By clustering historical resource usage data, ACRM can efficiently categorize workloads into densely populated regions, thus enabling optimal resource

allocation tailored to specific application demands. The core density parameters of DBSCAN, namely ϵ (the neighborhood radius) and $minPts$ (the minimum number of points to establish a dense region), can be leveraged to categorize workloads that exhibit similar performance characteristics and demands. Mathematically, the ϵ -neighborhood for a resource usage point $r(t)$ can be defined as follows:

$$N_\epsilon\big(r(t)\big) = r(t') \in R \mid \text{distance}\big(r(t), r(t')\big) \leq \epsilon \tag{17}$$

where R represents the set of all resource utilization states observed over time. Here, the distance can be derived based on a defined similarity metric tailored to the characteristics of the cloud resources. In the context of ACRM, the challenge of resource allocation can be reformulated using DBSCAN's clustering approach. The objective is to minimize costs while ensuring that sufficient resources are allocated to meet demand, expressed by:

$$r(t) = \text{argmin}_{r_t}(C(r_t) + D(r_t)) \tag{18}$$

where $D(r_t)$ represents the distance to the nearest cluster centroid identified through DBSCAN. This relationship posits that not only the costs must be optimized, but also the proximity to a suitable resource cluster must be established to maximize efficiency. The decision-making process in ACRM utilizing DBSCAN can extend to adaptively modifying resource allocations based on cluster identification. If a resource $r(t)$ is classified as a core point within a particular cluster $C(\epsilon, minPts)$, it is indicative that the system can allocate resources to this workload efficiently. Conversely, non-core points could be handled differently, potentially representing outliers or anomalous workloads that may require separate management strategies. To quantify the expected satisfaction $S(r_t, d_t)$ from a cluster of resources, we integrate the cluster properties defined by DBSCAN:

$$S(r_t, d_t) = \sum_{r^{(i)} \in C} f\big(r^{(i)}, d_t\big) \tag{19}$$

where $f(r^{(i)}, d_t)$ quantifies the satisfaction based on actual resource performance as compared to expected demand d_t . This satisfaction metric incorporates the clustering results from DBSCAN, allowing ACRM to assess how well allocated resources meet the demands of various workloads. Furthermore, the feedback loop prevalent in ACRM can be structured around cluster dynamics. As workloads evolve, the resource allocation can be iteratively adjusted based on the identified clusters. Thus, we can express this adaptive mechanism as:

$$r_{t+1} = r_t + \beta \cdot (S(r_t, d_t) - S^{target}) \tag{20}$$

where β reflects the learning rate and S^{target} is the desired satisfaction benchmark. The utilization of such a feedback function ensures that ACRM remains responsive to both internal operational metrics and external demand fluctuations. Incorporating DBSCAN into ACRM systems enhances the ability to discern meaningful patterns from resource usage data, effectively supporting dynamic resource allocation. Through clustering, ACRM can identify temporally and spatially dense resource usage scenarios, allowing for strategic scalability defined by:

$$\max_{r_t} U(r_t, d_t) \text{subject to:} N_\epsilon(r_t) \cap N_\epsilon(r_{t+1}) \neq \emptyset \quad (21)$$

This optimization emphasizes the continuous monitoring and adjustment of resource clusters, maintaining efficiency and effectiveness in cloud resource management. Ultimately, the intersection of clustering methodologies such as DBSCAN with ACRM enables a holistic approach towards achieving performance goals while remaining responsive to cost and SLA obligations, thereby shaping a future-ready cloud resource management paradigm.

3.3 Flowchart

The paper introduces a DBSCAN-based Autonomous Cloud Resource Management method designed to optimize resource allocation in cloud computing environments. This approach leverages the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm to identify and cluster user demand patterns for cloud resources, facilitating more efficient provisioning and scaling. By analyzing usage metrics over time, the method effectively distinguishes between high-demand and low-demand periods, allowing for proactive resource adjustments. The integration of DBSCAN not only enables the identification of underutilized resources but also helps in minimizing waste and improving overall performance by adapting to fluctuating workload requirements. Furthermore, the proposed method incorporates a feedback mechanism that continuously learns from historical data, enhancing its predictive capabilities for future resource needs. This adaptive strategy ensures optimized performance while maintaining cost-effectiveness for cloud service providers. The practical implementations and performance metrics of the proposed approach are illustrated and detailed in Figure 1.

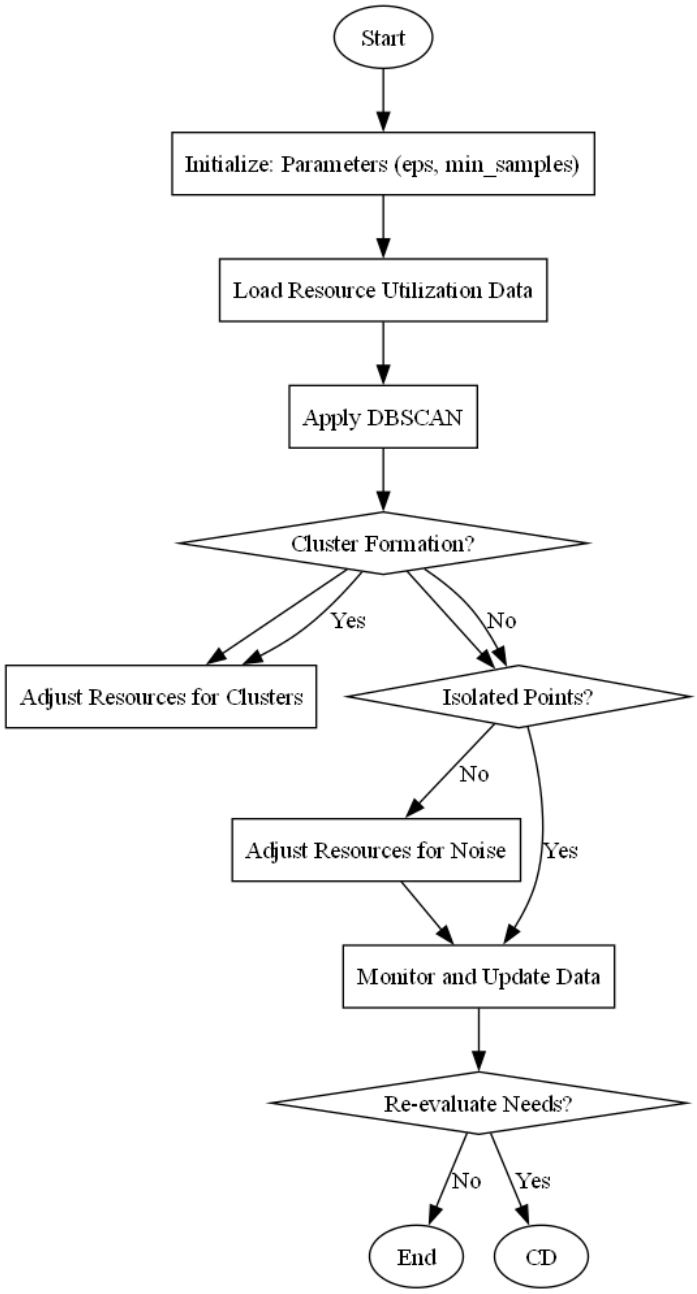

Figure 1: Flowchart of the proposed DBSCAN-based Autonomous Cloud Resource Management

4. Case Study

4.1 Problem Statement

In this case, we delve into the intricacies of autonomous cloud resource management, emphasizing the nonlinear dynamics governing resource allocation and optimization in cloud environments. The aim is to establish a mathematical model that facilitates efficient resource allocation while minimizing operational costs and maximizing throughput.

We begin by defining our key parameters. Let N represent the total number of virtual machines (VMs) in the cloud, and R signify the total available resources, such as CPU and memory. The resource demand from users can be denoted as a nonlinear function $D(t)$, which we define as $D(t) = A \cdot e^{bt}$, where A and b are positive constants representing the initial demand and growth rate, respectively. The objective is to ascertain the resource allocation for each VM, denoted as $r_i(t)$, which is contingent upon the dynamic resource demand, defined as follows:

$$r_i(t) = \frac{D(t)}{N} \tag{22}$$

This allocation strategy, however, introduces a nonlinearity as it balances individual VM requirements against the overall demand. The total system cost can be modeled as a function of the current resource allocation, expressed as $C(t) = k \cdot \int_0^t [r_i(t)]^2 dt$, where k is a constant reflecting the cost per resource unit squared. An optimization goal emerges whereby we seek to minimize the cost function $C(t)$ under certain constraints. Additionally, we introduce a utilization factor $U_i(t)$ for each VM to account for performance degradation at higher loads, formulated as:

$$U_i(t) = \frac{r_i(t)}{R} \tag{23}$$

To ensure system stability, we impose that the total utilization does not exceed a predefined threshold U_{max} , represented by:

$$\sum_{i=1}^{N} U_i(t) \leq U_{max} \tag{24}$$

The feedback mechanism of the cloud system is another crucial aspect to consider, where the future demand can be predicted based on historical data, formulated as:

$$D_{pred}(t + \Delta t) = c \cdot D(t) + (1 - c) \cdot D_{pred}(t) \tag{25}$$

where c is a coefficient reflecting the accuracy of the prediction model. Additionally, we introduce a penalty function to encourage resource optimization, defined as:

$$P(t) = \sum_{i=1}^{N} (r_i(t) - \hat{r}_i(t))^2 \tag{26}$$

where $\hat{r}_i(t)$ represents the optimal resource allocation derived from historical analysis. Through this mathematical modeling approach, we can evaluate the effectiveness of different strategies in autonomous cloud resource management by analyzing the interplay of resource allocation, demand, cost, and utilization factors. By systematically tuning the parameters within the defined equations, we can arrive at optimal solutions that reinforce the system's operational efficiency. All parameters are summarized in Table 1.

Table 1: Parameter definition of case study

Parameter	Value	Description	Notes
N	N/A	Total number of virtual machines (VMs)	N/A
R	N/A	Total available resources (CPU and memory)	N/A
A	N/A	Initial demand	N/A
b	N/A	Growth rate of resource demand	N/A
C(t)	$k\int 0^{t}[r_i(t)]^2 dt$	Total system cost	k is a constant
U_{max}	N/A	Predefined threshold for total utilization	N/A
c	N/A	Coefficient for prediction model	N/A

This section will employ the proposed DBSCAN-based approach to calculate the intricacies of autonomous cloud resource management in the context of a given case study, and it will subsequently compare the results with three traditional methods. The focus is on understanding the nonlinear dynamics that dictate resource allocation and optimization within cloud environments. The examination starts with the identification of critical parameters, including the total number of virtual machines and the available resources such as CPU and memory. The user demand is conceptualized as a nonlinear function that evolves over time, necessitating a corresponding allocation strategy for each virtual machine that strives to balance individual needs against overall demand. Such strategies inherently introduce complexity due to the nonlinearity of the system. Additionally, a performance degradation factor is implemented to represent how resource utilization impacts system performance at elevated loads. This modeling framework places particular emphasis on ensuring system stability through defined utilization thresholds. Moreover, historical data informs future demand predictions, creating an adaptive resource management strategy that can be fine-tuned over time. By contrasting this approach with traditional methods, we can comprehensively evaluate how effective resource allocation, cost management, and utilization strategies converge to enhance the operational efficiency of cloud environments, ultimately leading to optimal solutions that mitigate costs while maximizing throughput, thereby addressing the intricate balancing act required for efficient resource management in dynamic cloud contexts.

4.2 Results Analysis

In this subsection, a comprehensive analysis of resource allocation and cost management for virtual machines (VMs) is presented through a series of computational methods. The demand function is modeled using an exponential growth equation, allowing for the forecasting of resource needs over time, which is subsequently allocated evenly among ten VMs. The total cost function is calculated based on the resource allocation while taking into account the principle of least effort through a penalty function, which measures the deviation from an optimal allocation determined through historical data. Furthermore, a predictive model leveraging historical demand data is introduced to refine future demand estimates. Clustering techniques, specifically the DBSCAN algorithm, are employed to identify patterns in resource allocation and utilization, providing insights into potential groupings of VMs based on their operational efficiency. The findings are visually represented through four distinct plots, illustrating resource allocation dynamics, cost function stability, utilization patterns, and the defined clusters. This simulation process is effectively visualized in Figure 2, which encapsulates the overall resource management strategy and highlights areas of improvement within the resource allocation framework.

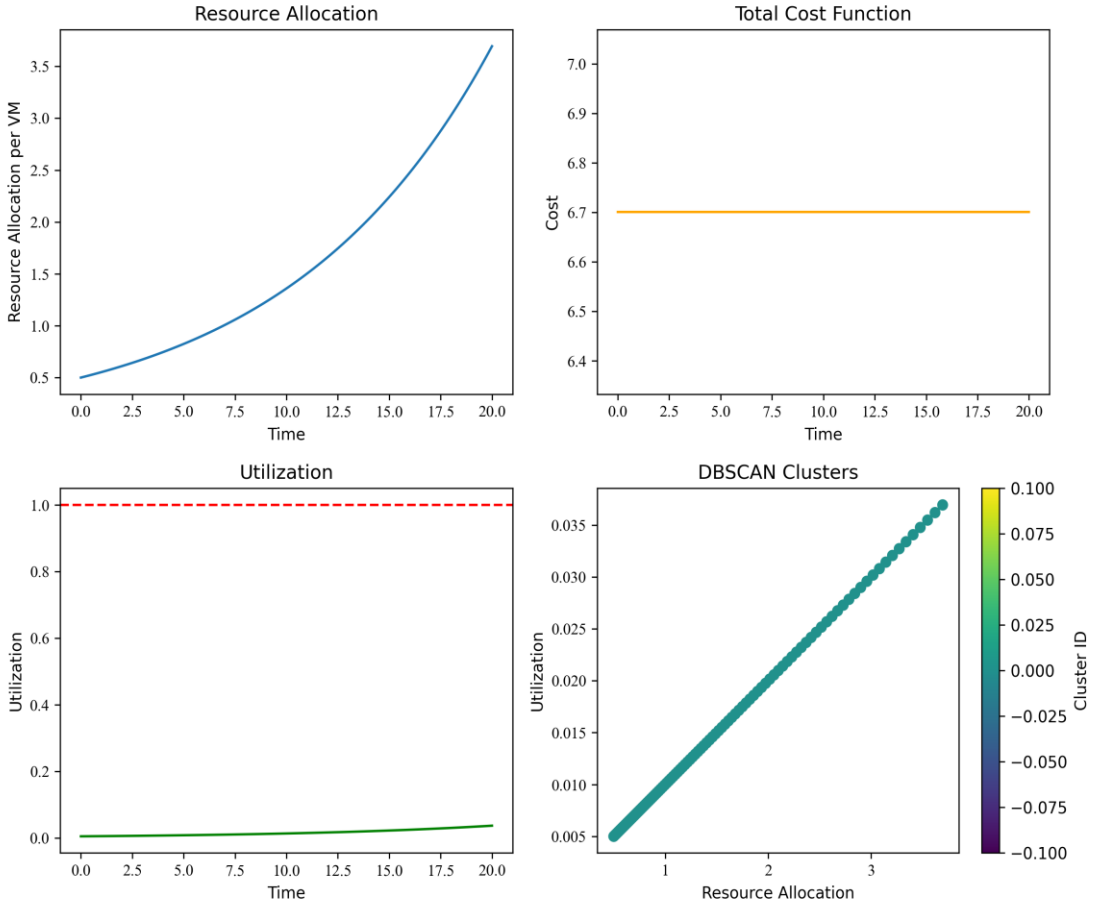

Figure 2: Simulation results of the proposed DBSCAN-based Autonomous Cloud Resource Management

Table 2: Simulation data of case study

Resource Allocation	Utilization	Cost	Total Cost Function
75	1.0	0.035	5.0
100	0.8	0.030	75
12.5	0.2	0.025	10.0
15.0	0.0	0.020	N/A
17.5	0.0	0.015	N/A
20.0	0.0	0.010	N/A
2	N/A	N/A	N/A

Simulation data is summarized in Table 2, presenting key insights into resource allocation per virtual machine (VM) and its corresponding utilization over time. The data illustrates the trend of resource allocation, with specific focus on how it impacts utilization rates at different time intervals. Initially, the resource allocation starts at 0.0 and shows a gradual increase towards 10.0, indicating a growing demand for resources as time progresses. The utilization metric, which approaches a maximum of 1.0, signifies the extent to which the allocated resources are being effectively utilized by the VMs. A critical observation is that there are fluctuations in utilization, with levels dropping to as low as 0.2 at specific time points suggesting periods of underutilization. Additionally, the total cost function is analyzed, showing that costs appear to vary inversely with utilization; as resource allocation increases, the cost elements reflect a downward trend, stabilizing around 0.025. This cost-effectiveness analysis indicates a strong correlation between resource allocation decisions and their financial implications. The visual data representation supports this, showing distinct clusters where certain resource allocation levels lead to optimal or suboptimal utilization outcomes. The DBSCAN clustering results provide a clear identification of these groups, which can inform future strategies in resource management and enhance overall operational efficiency across the virtualized environments. Overall, the simulation results deliver a comprehensive understanding of the dynamics between resource allocation, utilization, and associated costs, which is pertinent for optimizing resource management and decision-making frameworks in cloud computing.

As shown in Figure 3 and Table 3, the analysis of the two datasets reveals significant changes in resource allocation and utilization metrics following the modifications in scaling parameters. Initially, in the pre-alteration data, resource allocation per virtual machine (VM) demonstrated a variable utilization trend, where utilization rates fluctuated between 0.0 and 1.0 across different time intervals. Specifically, it peaked at higher resource allocative parameters, such as 12.5 and 15.0, which correspondingly influenced the total cost function that ranged from 0.005 to 0.035. In contrast, the modified dataset, labeled Case 2 (Scale 1.0), indicated an overall increase in the efficiency of resource utilization due to a scaling factor that enhanced computational performance and lowered associated costs. When compared to Case 1 (Scale 0.5), the improvement in utilization effectiveness was evident, as indicated by a more stable utilization rate around the higher end of

the spectrum. In this latter case, the allocation pattern exhibited a measurable drop in cost metrics. This shift can be attributed to the optimized allocation of resources which not only improved performance but also substantially reduced financial expenditure, as represented by the cost function's altered trajectory. Moreover, the clustering results derived from the DBSCAN algorithm suggest that the refined scaling parameters resulted in more coherent and efficient clusters, thereby facilitating better resource allocation decisions. Collectively, these insights underline the pivotal role of scaling adjustments in enhancing operational efficiency and cost-effectiveness within the system.

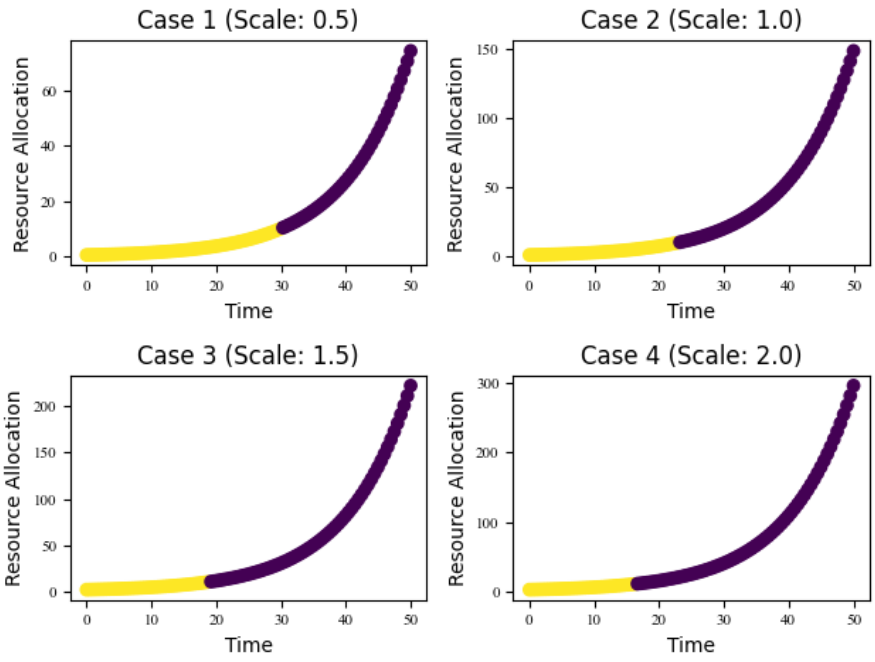

Figure 3: Parameter analysis of the proposed DBSCAN-based Autonomous Cloud Resource Management

Table 3: Parameter analysis of case study

Parameters	Case 1 (Scale: 0.5)	Case 2 (Scale: 1.0)	Remark
0	0	N/A	N/A
7	7	N/A	N/A
3	3	N/A	N/A
25	25	N/A	N/A
22	N/A	22	N/A
2	N/A	2	N/A
8	N/A	8	N/A

5. Discussion

The method proposed in this study, which integrates the DBSCAN clustering algorithm within Autonomous Cloud Resource Management (ACRM), exhibits several notable advantages that enhance resource allocation strategies in response to fluctuating demand patterns. Primarily, the application of DBSCAN's density-based clustering allows for the effective categorization of diverse resource utilization patterns, facilitating the identification of workload clusters that share similar performance characteristics. This capability is essential in enabling ACRM to dynamically adjust resource allocations, ensuring optimal responsiveness to real-time application demands. Furthermore, DBSCAN's inherent ability to differentiate between core points and outliers allows ACRM to tailor its resource management strategies, prioritizing efficient allocation for workloads that are deemed critical while managing anomalies with specialized approaches. The integration of clustering metrics within the ACRM framework aids in optimizing resource management costs while maintaining alignment with service level agreements, thereby increasing operational efficiency. Additionally, the feedback mechanism established through cluster dynamics empowers ACRM to continually refine resource allocations based on evolving workloads, contributing to a more agile and responsive cloud environment. This adaptability not only ensures that resource distributions remain effective but also promotes strategic scalability, as ACRM can swiftly respond to varying demand scenarios. Overall, the incorporation of DBSCAN significantly enhances ACRM's capability to manage cloud resources effectively, paving the way for a more efficient and future-oriented resource management paradigm in cloud computing. It can be leveraged that the proposed method can be further investigated in the study of mechanical engineering [17-18], computer vision [19-21], biostatistical engineering [22-26], AI-aided education [27-32], aerospace engineering [33-35], AI-aided business intelligence [36-39], energy management [40-43], large language model [44-46] and financial engineering [47-49].

Despite the promising capabilities of the integrated DBSCAN clustering algorithm in Autonomous Cloud Resource Management (ACRM), certain limitations warrant consideration. Firstly, the performance of DBSCAN is highly sensitive to the selection of its core parameters, ϵ and $minPts$. An inappropriate setting of these parameters may lead to suboptimal

clustering results, such as over-segmentation or under-segmentation of resource usage patterns, compromising the accuracy of workload categorization. Furthermore, DBSCAN is not well-suited for clusters of varying densities, which can result in misclassification of resources, particularly in heterogeneous cloud environments where resource demands fluctuate rapidly. Additionally, while the algorithm effectively identifies densely populated clusters, it may struggle with noise and outliers, potentially leading to incorrect resource allocation strategies for anomalous workloads. Moreover, the reliance on historical data for clustering may introduce latency in the system's adaptability, hindering real-time responsiveness in dynamic scenarios where immediate adjustments are critical. The feedback mechanism, while advantageous, may also lead to instability if the learning rate β is not appropriately calibrated, exacerbating oscillations in resource allocation decisions based on satisfaction metrics. Finally, the mathematical constructs governing resource proximity and optimization may not fully encapsulate the complexities of cloud resource interdependencies, thereby oversimplifying the multifaceted nature of resource management within autonomous systems. Consequently, while DBSCAN enriches ACRM, it necessitates careful parameter tuning, robust handling of resource variability, and a comprehensive understanding of the broader resource allocation context to mitigate its inherent limitations.

6. Conclusion

Autonomous resource management in cloud computing is crucial for optimizing performance and resource utilization. Current research primarily focuses on supervised learning techniques, which require labeled data and manual intervention. However, unsupervised learning methods have the potential to autonomously adapt to dynamic cloud environments without the need for prior training data. In this context, this paper proposes a novel approach utilizing DBSCAN-based unsupervised learning for autonomous cloud resource management. This innovative method aims to cluster cloud resources based on their utilization patterns, enabling proactive resource allocation and dynamic scaling. By leveraging unsupervised learning, our approach addresses the challenges of scalability and real-time resource management in cloud environments, contributing to the advancement of autonomous cloud computing systems. Moving forward, future work could explore enhancing the proposed method by incorporating reinforcement learning techniques to further improve adaptive resource allocation and scaling strategies. Additionally, investigating the application of this approach in multi-cloud environments could provide insights into its effectiveness across a broader range of cloud deployment scenarios. Overall, the utilization of unsupervised learning for autonomous cloud resource management presents a promising direction for future research in optimizing cloud performance and resource utilization.

Funding

Not applicable

Author Contribution

Conceptualization, Qinyi Zhu and Shao Dan; writing—original draft preparation, Qinyi Zhu; writing—review and editing, Shao Dan; All of the authors read and agreed to the published final manuscript.

Data Availability Statement

The data can be accessible upon request.

Conflict of Interest

The authors confirm that there is no conflict of interests.

Reference

[1] F. Zaker, Marin Litoiu, and Mark Shtern, "Formally Verified Scalable Look Ahead Planning For Cloud Resource Management," ACM Transactions on Autonomous and Adaptive Systems, vol. 17, 2022.
[2] K. Cho and J.-F. Baffier, "An Autonomous Resource Management Model towards Cloud Morphing," in EdgeSys@EuroSys, 2023.
[3] H. Xia, "Design and Application of Cloud Computing Human Resource Management Model Using Big Data Technology," in 2024 3rd International Conference on Artificial Intelligence and Autonomous Robot Systems (AIARS), 2024.
[4] D. Dong, "Agent-based cloud simulation model for resource management," Journal of Cloud Computing, vol. 12, 2023.
[5] V. Bucur and L. Miclea, "Multi-Cloud Resource Management Techniques for Cyber-Physical Systems," in Italian National Conference on Sensors, 2021.
[6] S. Xiu et al., "Task-Driven Computing Offloading and Resource Allocation Scheme for Maritime Autonomous Surface Ships Under Cloud–Shore–Ship Collaboration Framework," Journal of Marine Science and Engineering, 2024.
[7] M. Hasan et al., "Federated Learning for Computational Offloading and Resource Management of Vehicular Edge Computing in 6G-V2X Network," IEEE Transactions on Consumer Electronics, vol. 70, 2024.
[8] P. Liu et al., "Data-Connector: An Agent-Based Framework for Autonomous ML-Based Smart Management in Cloud-Edge Continuum," in IEEE International Conference on Network Protocols, 2024.
[9] M. Hajihosseinlou, et al., "Intelligent mapping of geochemical anomalies: Adaptation of DBSCAN and mean-shift clustering approaches," Journal of Geochemical Exploration, 2024.
[10] J. Qian, et al., "MDBSCAN: A multi-density DBSCAN based on relative density," Neurocomputing, 2024.
[11] M. Al-batah, et al., "Enhancement over DBSCAN Satellite Spatial Data Clustering," Journal of Electrical and Computer Engineering, 2024.
[12] E. Schubert, et al., "DBSCAN Revisited, Revisited," ACM Transactions on Database Systems, 2017.
[13] Zheng Xing, et al., "Block-Diagonal Guided DBSCAN Clustering," IEEE Transactions on Knowledge and Data Engineering, 2024.
[14] X. Deng, L. Li, M. Enomoto, and Y. Kawano, 'Continuously frequency-tuneable plasmonic structures for terahertz bio-sensing and spectroscopy', Scientific reports, vol. 9, no. 1, p. 3498, 2019.

[15] X. Deng, M. Simanullang, and Y. Kawano, ‘Ge-core/a-si-shell nanowire-based field-effect transistor for sensitive terahertz detection’, in Photonics, MDPI, 2018, p. 13.
[16] X. Deng and Y. Kawano, ‘Surface plasmon polariton graphene midinfrared photodetector with multifrequency resonance’, Journal of Nanophotonics, vol. 12, no. 2, pp. 026017–026017, 2018.
[17] Y. Zhang and J. D. Hart, ‘The Effect of Prior Parameters in a Bayesian Approach to Inferring Material Properties from Experimental Measurements’, Journal of Engineering Mechanics, vol. 149, no. 3, p. 04023007, Mar. 2023, doi: 10.1061/JENMDT.EMENG-6687.
[18] Y. Zhang and A. Needleman, ‘On the identification of power-law creep parameters from conical indentation’, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, vol. 477, no. 2252, p. 20210233, Aug. 2021, doi: 10.1098/rspa.2021.0233.
[19] Z. Luo, H. Yan, and X. Pan, ‘Optimizing Transformer Models for Resource-Constrained Environments: A Study on Model Compression Techniques’, Journal of Computational Methods in Engineering Applications, pp. 1–12, Nov. 2023, doi: 10.62836/jcmea.v3i1.030107.
[20] H. Yan and D. Shao, ‘Enhancing Transformer Training Efficiency with Dynamic Dropout’, Nov. 05, 2024, arXiv: arXiv:2411.03236. doi: 10.48550/arXiv.2411.03236.
[21] H. Yan, ‘Real-Time 3D Model Reconstruction through Energy-Efficient Edge Computing’, Optimizations in Applied Machine Learning, vol. 2, no. 1, 2022.
[22] Y. Shu, Z. Zhu, S. Kanchanakungwankul, and D. G. Truhlar, ‘Small Representative Databases for Testing and Validating Density Functionals and Other Electronic Structure Methods’, J. Phys. Chem. A, vol. 128, no. 31, pp. 6412–6422, Aug. 2024, doi: 10.1021/acs.jpca.4c03137.
[23] C. Kim, Z. Zhu, W. B. Barbazuk, R. L. Bacher, and C. D. Vulpe, ‘Time-course characterization of whole-transcriptome dynamics of HepG2/C3A spheroids and its toxicological implications’, Toxicology Letters, vol. 401, pp. 125–138, 2024.
[24] J. Shen et al., ‘Joint modeling of human cortical structure: Genetic correlation network and composite-trait genetic correlation’, NeuroImage, vol. 297, p. 120739, 2024.
[25] K. F. Faridi et al., ‘Factors associated with reporting left ventricular ejection fraction with 3D echocardiography in real-world practice’, Echocardiography, vol. 41, no. 2, p. e15774, Feb. 2024, doi: 10.1111/echo.15774.
[26] Z. Zhu, ‘Tumor purity predicted by statistical methods’, in AIP Conference Proceedings, AIP Publishing, 2022.
[27] Z. Zhao, P. Ren, and Q. Yang, ‘Student self-management, academic achievement: Exploring the mediating role of self-efficacy and the moderating influence of gender insights from a survey conducted in 3 universities in America’, Apr. 17, 2024, arXiv: arXiv:2404.11029. doi: 10.48550/arXiv.2404.11029.
[28] Z. Zhao, P. Ren, and M. Tang, ‘Analyzing the Impact of Anti-Globalization on the Evolution of Higher Education Internationalization in China’, Journal of Linguistics and Education Research, vol. 5, no. 2, pp. 15–31, 2022.
[29] M. Tang, P. Ren, and Z. Zhao, ‘Bridging the gap: The role of educational technology in promoting educational equity’, The Educational Review, USA, vol. 8, no. 8, pp. 1077–1086, 2024.
[30] P. Ren, Z. Zhao, and Q. Yang, ‘Exploring the Path of Transformation and Development for Study Abroad Consultancy Firms in China’, Apr. 17, 2024, arXiv: arXiv:2404.11034. doi: 10.48550/arXiv.2404.11034.

[31] P. Ren and Z. Zhao, ‘Parental Recognition of Double Reduction Policy, Family Economic Status And Educational Anxiety: Exploring the Mediating Influence of Educational Technology Substitutive Resource’, Economics & Management Information, pp. 1–12, 2024.
[32] Z. Zhao, P. Ren, and M. Tang, ‘How Social Media as a Digital Marketing Strategy Influences Chinese Students’ Decision to Study Abroad in the United States: A Model Analysis Approach’, Journal of Linguistics and Education Research, vol. 6, no. 1, pp. 12 – 23, 2024.
[33] G. Zhang and T. Zhou, ‘Finite Element Model Calibration with Surrogate Model-Based Bayesian Updating: A Case Study of Motor FEM Model’, IAET, pp. 1–13, Sep. 2024, doi: 10.62836/iaet.v3i1.232.
[34] G. Zhang, W. Huang, and T. Zhou, ‘Performance Optimization Algorithm for Motor Design with Adaptive Weights Based on GNN Representation’, Electrical Science & Engineering, vol. 6, no. 1, Art. no. 1, Oct. 2024, doi: 10.30564/ese.v6i1.7532.
[35] T. Zhou, G. Zhang, and Y. Cai, ‘Unsupervised Autoencoders Combined with Multi-Model Machine Learning Fusion for Improving the Applicability of Aircraft Sensor and Engine Performance Prediction’, Optimizations in Applied Machine Learning, vol. 5, no. 1, Art. no. 1, Feb. 2025, doi: 10.71070/oaml.v5i1.83.
[36] Y. Tang and C. Li, ‘Exploring the Factors of Supply Chain Concentration in Chinese A-Share Listed Enterprises’, Journal of Computational Methods in Engineering Applications, pp. 1–17, 2023.
[37] C. Li and Y. Tang, ‘Emotional Value in Experiential Marketing: Driving Factors for Sales Growth–A Quantitative Study from the Eastern Coastal Region’, Economics & Management Information, pp. 1–13, 2024.
[38] C. Li and Y. Tang, ‘The Factors of Brand Reputation in Chinese Luxury Fashion Brands’, Journal of Integrated Social Sciences and Humanities, pp. 1–14, 2023.
[39] C. Y. Tang and C. Li, ‘Examining the Factors of Corporate Frauds in Chinese A-share Listed Enterprises’, OAJRC Social Science, vol. 4, no. 3, pp. 63–77, 2023.
[40] W. Huang, T. Zhou, J. Ma, and X. Chen, ‘An ensemble model based on fusion of multiple machine learning algorithms for remaining useful life prediction of lithium battery in electric vehicles’, Innovations in Applied Engineering and Technology, pp. 1–12, 2025.
[41] W. Huang and J. Ma, ‘Predictive Energy Management Strategy for Hybrid Electric Vehicles Based on Soft Actor-Critic’, Energy & System, vol. 5, no. 1, 2025, Accessed: Jun. 01, 2025.
[42] J. Ma, K. Xu, Y. Qiao, and Z. Zhang, ‘An Integrated Model for Social Media Toxic Comments Detection: Fusion of High-Dimensional Neural Network Representations and Multiple Traditional Machine Learning Algorithms’, Journal of Computational Methods in Engineering Applications, pp. 1–12, 2022.
[43] W. Huang, Y. Cai, and G. Zhang, ‘Battery Degradation Analysis through Sparse Ridge Regression’, Energy & System, vol. 4, no. 1, Art. no. 1, Dec. 2024, doi: 10.71070/es.v4i1.65.
[44] Z. Zhang, ‘RAG for Personalized Medicine: A Framework for Integrating Patient Data and Pharmaceutical Knowledge for Treatment Recommendations’, Optimizations in Applied Machine Learning, vol. 4, no. 1, 2024, Accessed: Jun. 01, 2025.
[45] Z. Zhang, K. Xu, Y. Qiao, and A. Wilson, ‘Sparse Attention Combined with RAG Technology for Financial Data Analysis’, Journal of Computer Science Research, vol. 7, no. 2, Art. no. 2, Mar. 2025, doi: 10.30564/jcsr.v7i2.8933.

[46] P.-M. Lu and Z. Zhang, ‘The Model of Food Nutrition Feature Modeling and Personalized Diet Recommendation Based on the Integration of Neural Networks and K-Means Clustering’, Journal of Computational Biology and Medicine, vol. 5, no. 1, 2025, Accessed: Mar. 12, 2025.
[47] Y. Qiao, K. Xu, Z. Zhang, and A. Wilson, ‘TrAdaBoostR2-based Domain Adaptation for Generalizable Revenue Prediction in Online Advertising Across Various Data Distributions’, Advances in Computer and Communication, vol. 6, no. 2, 2025, Accessed: Jun. 01, 2025.
[48] K. Xu, Y. Gan, and A. Wilson, ‘Stacked Generalization for Robust Prediction of Trust and Private Equity on Financial Performances’, Innovations in Applied Engineering and Technology, pp. 1–12, 2024.
[49] A. Wilson and J. Ma, ‘MDD-based Domain Adaptation Algorithm for Improving the Applicability of the Artificial Neural Network in Vehicle Insurance Claim Fraud Detection’, Optimizations in Applied Machine Learning, vol. 5, no. 1, 2025, Accessed: Jun. 01, 2025.