Systematic Literature survey for Load Balancing Strategy in Cloud Computing

Hemant Kumar Singh[1], Dr. Midhun chakkaravarthy[2], Dr. Shashi Kant Gupta[3]
[1]Postdoctoral Research Fellow, Lincoln University College, Malaysia
[2]Lincoln University College, Malaysia,
[3]Centre for Research Impact & Outcome, Chitkara University, Punjab, India
[1]hemantbib@gmail.com, [2]midhun@lincoln.edu.my, [3]raj2008enator@gmail.com

Abstract:

Cloud computing is a rapidly evolving technology that integrates multiple essential components to establish a cohesive network of interconnected devices. These devices, including sensors, routers, smartphones, and smart appliances, form the backbone of the Internet of Everything (IoE). The massive amounts of data generated by IoE devices are processed and stored in the cloud, enabling real-time analysis and actionable insights. Consequently, there is a critical demand for effective load-balancing and task-scheduling mechanisms in cloud computing. The primary goal of these techniques is to distribute workloads evenly across available resources while addressing challenges such as minimizing execution and response times, enhancing throughput, and improving fault detection. This systematic literature review (SLR) focuses on examining various technologies, including optimization and machine learning algorithms, applied to load-balancing and task-scheduling challenges in cloud computing environments.
A diverse collection of 28 research articles has been compiled for an in-depth technology review, focusing on studies published over the past decade. The primary goal of load balancing (LB) is to efficiently distribute workloads across available resources, optimizing overall turnaround time. Prior to the last 10 years, traditional scheduling and load balancing methods, including First-Come-First-Serve (FCFS), Shortest Job First (SJF), Minimum-Minimum (Min-Min), Maximum-Minimum (Max-Min), and Round Robin (RR), were commonly employed. However, these techniques were frequently criticized for their inefficiency and slower processing speeds.

Keywords: Cloud computing, Load balancing, Machine learning, Virtual Machine, Optimization techniques

1.0 Introduction:

Cloud Load Balancing:

Load balancing is a process of task scheduling among virtual machines using a hypervisor (Eg. Virtual Machine Manager). (Buyya 2018) (Mishra and Majhi 2020) The cloud computing infrastructure has three substantial challenges: virtualization, distributed frameworks and load balancing. The load-balancing problem is defined as the distribution of workloads among the processing units. In a multi-node environment, it is pretty probable that certain nodes will experience extreme workload while others will remain inactive.
(Oduwole et al. 2022). Verma et al. (2024) given a load-balancing methodology, utilizing genetic algorithms (GA) to advance the quality of the telemedicine industry by efficiently adapting to changing workloads and network conditions at the fog level. The flexibility to adapt can enhance patient care and provide scalability for future healthcare systems.
Walia et al. (2023) cover several emerging technologies in their survey, including Software-Defned Networking (SDN), Blockchain, Digital Twins, Industrial IoT (IIoT), 5G, Serverless computing, and quantum computing. These technologies can be incorporated with the current fog/edge-of-things models for improved analysis and provide business intelligence for IoT platforms. Adaptive resource management

strategies are necessary for efficient scheduling and decision-ofoading due to the infrastructural efficiency of these computing paradigms. Following is the classification of load balancing algorithm

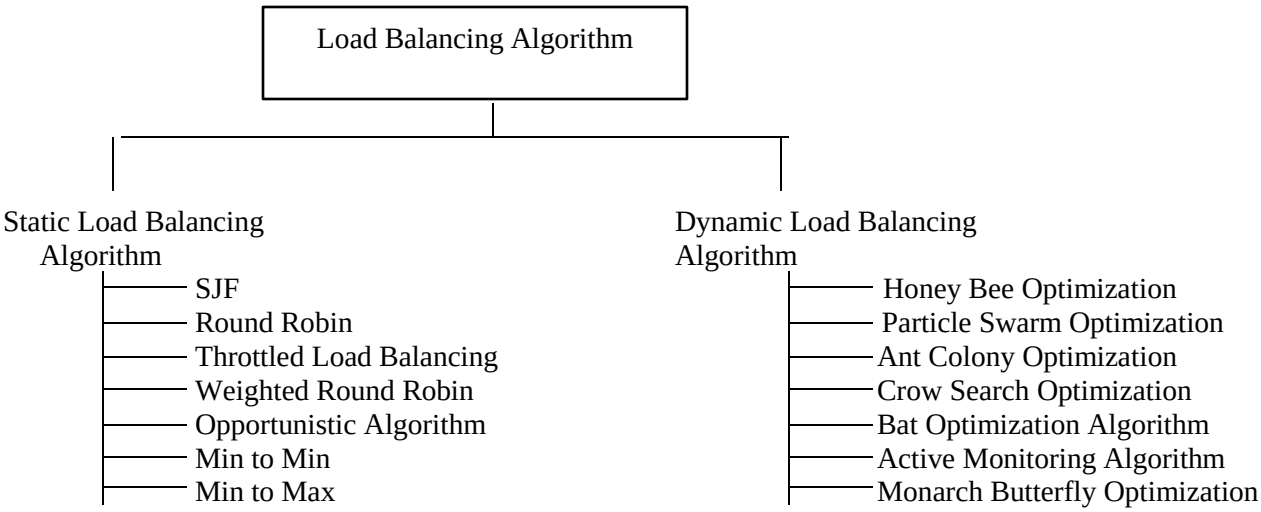

Figure-1: Classification of load balancing algorithm

1.1 Need for load balancing, factors affecting and associated challenges:

Smart Computing Resource Management is swiftly developing to meet the swelling needs of businesses in various sectors. It is driven by the propagation of Internet-based technologies, cloud computing and cyber-physical systems. With the growth of information intensive applications, artificial intelligence, cloud computing and IoT, intelligent computing monitoring and resource allocation have become crucial (Biswas et al. 2024). Cloud data centres naturally need to be optimized because they are built to handle hundreds of loads, which could result in low resource utilization and energy waste. The goals of load balancing include reduced job execution times, optimal resource utilization, and high system throughput. Load balancing reduces the overall resource waiting time and avoids resource overload (Apat et al. 2023).

The Industrial Internet of Things (IIoT) has experienced significant advancement and implementation due to the quick progress and use of artificial intelligence techniques. In Industry 5.0, the hyper-automation process involves the deployment of intelligent devices connected to the Industrial Internet of Things (IIoT), cloud computing, smart robots, agile software, and embedded components. These systems can leverage the Industry 5.0 concept, which generates massive amounts of data for hyper-automated communication across cloud computing, digital transformation, human sectors, intelligent robots, and industrial production. Big data management requires cloud and fog technology (Souri et al. 2024).

Similarly, telemedicine, facilitated by fog computing, has revolutionized the healthcare industry by providing remote access to medical treatments. However, ensuring minimal latency and efective resource utilization are essential for providing high-quality healthcare (Verma et al. 2024). Big data in the industrial sector is crucial for predictive maintenance, enabling informed decisions and enhancing task allocation in Industry 4.0, thus necessitating a profcient resource management system (Teoh et al. 2023). The growing demand for load balancing in various industries using cloud/fog. services prompted us to contemplate and inspired us to compose an evaluation of the escalating necessity for resource management technologies.

1.2 Objective of Systematic Literature survey:

- Systematically classify diverse load balancing algorithms used in cloud computing.
- To address significant research questions like effectiveness of different algorithmic methods, simulation tools and metrics evaluation.
- To analyse developments and patterns in the literature like occurrence of Meta-heuristic, Hybrid, and ML-centric approaches and detect any changes or evolving paradigms in algorithm design.

- To conduct a comparative analysis of the different algorithm categories, identifying strengths, weaknesses, research limitations and trade-offs between them.
- Finding the research gap by identifying areas where further research and advancement are needed.

1.3 Methodology of the systematic literature review

This section arranges the components of a systematic literature review that includes the search criteria, review methodology and research questions. This process involves defining research questions or objectives, identifying relevant databases and sources and systematically searching and screening for eligible studies. The search term constitutes a string encompassing all essential keywords in the research questions and their corresponding synonyms.
The various computer science publication libraries were manually searched. The Systematic Literature survey search was conducted using the Scopus database, IEEE Computer Society, Research Gate, Science Direct, Springer, and ACM Digital Archive. A comprehensive selection of 60 relevant research papers was finalized and considered for the technological survey.

1.4 Inclusion–exclusion criteria for filtering the relevant articles for Systematic Literature survey

Inclusion

- All papers including keywords "Load balancing and Task scheduling in cloud computing using Machine learning"
- The period chosen is 2014–2024
- The subject area chosen is "Computer Science."
- Document type taken is either "research article" or "Conference papers"
- The English language is chosen
- All those papers which have at least or above 0 citations

Exclusion:

- Papers not in the English language
- Papers that fall out of the time frame of 2014–2024
- All those research papers that do not contain keywords and emphasize the answers related to the research questions
- Papers have 0 citations Research papers that are commercial and require a purchase to access

2.0 Literature survey on load balancing (LB) :

Diverse selection of 60 research articles is compiled for a comprehensive technology review, covering studies published since last 10 years. The primary objective of load balancing (LB) is to distribute workloads efficiently across available resources, thereby enhancing overall turnaround time. Before 10 years, conventional scheduling and load balancing techniques such as First-Come-First-Serve (FCFS), Shortest Job First (SJF), Minimum-Minimum (MIM-Min), Maximum-Minimum (Max-Min), and Round Robin (RR) were widely used but were often criticized for their inefficiency and slow processing speeds. Konjaang et al. explored the challenges associated with the traditional Max-Min algorithm and introduced the Expa-Max-Min method as a potential improvement .The algorithm efficiently schedules cloudlets by prioritizing those with the longest and shortest execution times. Workload distribution can be categorized based on factors such as memory capacity, CPU load, and network load. To enhance performance in cloud computing, load balancing techniques, along with virtual machine management (VMM), are employed to distribute workloads among virtual machines [1, 2].
Hung et al. introduced an improved Max-Min algorithm known as MMSIA. The primary goal of MMSIA is to enhance completion time in cloud computing by leveraging machine learning to cluster requests and optimize virtual machine utilization. This method assigns large tasks to virtual machines (VMs) with the lowest utilization rates, thereby boosting processing efficiency. The approach incorporates supervised learning into the Max-Min scheduling algorithm to refine clustering accuracy [3].

Kumar et al. highlight that the modified HEFT algorithm constructs a Directed Acyclic Graph (DAG) for all cloud-submitted tasks. Additionally, it determines computational costs and communication links between processing resources to optimize task execution [4].

Seth and Singh introduced the Dynamic Heterogeneous Shortest Job First (DHSJF) model as an advanced approach to task scheduling in cloud computing environments with diverse resource capabilities. This algorithm establishes a heterogeneous cloud infrastructure, dynamically generates cloudlet lists, and evaluates workload distribution and resource diversity to reduce the Makespan. By efficiently allocating dynamic requests to available resources, DHSJF enhances resource utilization. This method effectively addresses the shortcomings of the traditional Shortest Job First (SJF) approach.

Farrag et al. (2020) explored the implementation of the Ant-Lion Optimizer (ALO) and Grey Wolf Optimizer (GWO) for job scheduling in cloud computing. These algorithms aim to enhance task scheduling efficiency by optimizing the makespan through effective workload distribution. While ALO and GWO demonstrate superior performance compared to the Firefly Algorithm (FFA) in reducing makespan, their effectiveness relative to Particle Swarm Optimization (PSO) depends on specific conditions.

Reddy et al. (2022) proposed the AVS-PGWO-RDA framework, which integrates Probabilistic Grey Wolf Optimization (PGWO) in the load balancing unit to determine the optimal fitness value for task selection and resource allocation. This approach prioritizes tasks with lower complexity and execution time. The AVS technique is utilized to group related workloads, while the RDA-based scheduler assigns these clusters to appropriate virtual machines (VMs) within the cloud.

Likewise, Janakiraman and Priya (2023) introduced the Hybrid Grey Wolf and Improved Particle Swarm Optimization Algorithm with Adaptive Inertial Weight-based Multi-Dimensional Learning Strategy (HGWIPSOA). This method combines the Grey Wolf Optimization Algorithm (GWOA) with Particle Swarm Optimization (PSO) to enhance task allocation to virtual machines, ultimately improving scheduling accuracy.

The proposed system significantly improves task scheduling speed and resource allocation efficiency in cloud environments. It addresses the limitations of previous load-balancing methods by preventing premature convergence and enhancing global search capabilities. As a result, it offers several advantages, such as increased throughput, minimized makespan, reduced imbalance, lower latency, and shorter execution time. Behera and Sobhanayak (2024) demonstrated that integrating the Grey Wolf Optimizer (GWO) with the Genetic Algorithm (GA) leads to enhanced performance, ensuring faster convergence and minimized makespan in large-scale task scheduling scenarios.

Since early 2014, metaheuristic and hybrid-metaheuristic algorithms have been widely utilized to address optimization and load-balancing challenges in cloud computing. Zhan et al. (2014) introduced the Load-Aware Genetic Algorithm (LAGA), an enhanced version of the Genetic Algorithm (GA). LAGA integrates the TLB model to optimize makespan and load balancing, incorporating a novel fitness function to determine optimal schedules while maintaining an efficient balance.

Rekha and Dakshayini (2019) proposed a task allocation method based on the Genetic Algorithm for cloud environments, aiming to minimize job completion time and improve overall system performance. This approach considers multiple factors, including energy efficiency and response time, to optimize resource allocation. The evaluation results demonstrated superior throughput, validating the effectiveness of the proposed task allocation method.

Mishra and Majhi (2023) introduced a hybrid metaheuristic technique called GAYA, which combines the Genetic Algorithm (GA) and the JAYA algorithm to efficiently schedule dynamically independent biological data. GAYA enhances both exploration and exploitation capabilities, making it a reliable solution for scheduling medical data in cloud-based environments.

Brahmam and Vijay Anand (2024) developed the VMMISD model, which integrates the Genetic Algorithm (GA) with Ant Colony Optimization (ACO) for resource allocation. This system incorporates iterative optimization techniques, deep learning models, and security protocols to enhance load balancing efficiency during virtual machine migrations. It employs K-means clustering, Fuzzy Logic, Long Short-

Term Memory (LSTM) networks, and Graph Networks to predict workloads, support decision-making, and assess affinity between virtual machines (VMs) and physical machines.

Additionally, Behera and Sobhanayak (2024) introduced a hybrid approach that merges the Grey Wolf Optimizer (GWO) with the Genetic Algorithm (GA). This hybrid GWO-GA model effectively reduces makespan, energy consumption, and computing costs, outperforming conventional algorithms in large-scale scheduling scenarios. Its ability to accelerate convergence provides a significant advantage over previous techniques.

Furthermore, the combination of autoscaling and reinforcement learning (RL) has gained considerable attention for its capability to dynamically allocate resources in an adaptive and efficient manner (Joshi et al. 2024). Deep Reinforcement Learning (DRL) is an emerging technique that automates workload prediction and enables real-time decision-making for resource allocation. By continuously monitoring workload fluctuations and performance metrics, DRL ensures efficient resource distribution to meet system demands effectively.

Ran et al. (2019) proposed a task scheduling approach utilizing deep reinforcement learning (DRL). This DRL-based load balancer dynamically allocates tasks to virtual machines (VMs), effectively reducing average response time while maintaining load balance. The method was tested on a tower server with specific hardware and software configurations, demonstrating its effectiveness in distributing workloads across VMs while ensuring compliance with service level agreement (SLA) constraints. By leveraging deep reinforcement learning (DRL) and deep deterministic policy gradient (DDPG) networks, the approach facilitates optimal scheduling decisions.

Ran et al. (2019) introduced a task scheduling approach based on deep reinforcement learning (DRL), enabling the system to learn from experience without requiring prior knowledge. This DRL-based load balancer dynamically distributes tasks among virtual machines (VMs), reducing average response time and ensuring balanced workload distribution. The method was tested on a tower server with specific hardware and software configurations, demonstrating its effectiveness in workload management while maintaining compliance with service-level agreements (SLAs). By utilizing DRL and deep deterministic policy gradient (DDPG) networks, the approach facilitates optimal scheduling decisions.

Similarly, Jyoti and Shrimali (2020) applied DRL in their study and developed a technique known as Multi-Agent Deep Reinforcement Learning-Dynamic Resource Allocation (MADRL-DRA). This method integrates the Local User Agent (LUA) and Dynamic Optimal Load-Aware Service Broker (DOLASB) within the Global User Agent (GUA) to dynamically allocate resources and improve quality of service (QoS) metrics. The approach enhances various performance indicators, including execution time, waiting time, energy efficiency, throughput, resource utilization, and makespan, surpassing conventional techniques.

Furthermore, Tong et al. (2021) proposed a DRL-based task scheduling method designed to minimize virtual machine (VM) load imbalance and job rejection rates while ensuring adherence to SLA constraints. The proposed Deep Deterministic Multi-Agent Task Scheduling (DDMTS) algorithm demonstrates improved stability and outperforms existing algorithms in balancing the Degree of Imbalance (DI) and reducing job rejection rates. The algorithm's effectiveness in addressing task scheduling challenges relies on the precise configuration of state, action, and reward parameters, leveraging the Deep Q-Network (DQN) algorithm for optimal decision-making.

Double Deep Q-learning has been effectively applied to address load-balancing issues in various applications. Swarup et al. (2021) introduced a Deep Reinforcement Learning (DRL)-based approach to optimize job scheduling in cloud computing. Their method utilizes a **Clipped Double Deep Q-learning** algorithm designed to reduce computational costs while ensuring compliance with resource limitations and deadline requirements. The algorithm incorporates **target networks** and **experience replay** mechanisms to enhance the optimization of its objective function. To maintain a balance between exploration and exploitation, the algorithm employs an **ε-greedy policy**, which guides action selection by weighing the trade-off between exploring new actions and exploiting known strategies. Specifically, the system either chooses actions randomly (to encourage exploration) or based on Q-values (to prioritize

exploitation), ensuring a dynamic equilibrium between testing new options and leveraging existing knowledge.

In a similar vein, Kruekaew et al. (Mao et al., 2014) applied Q-learning to improve job scheduling and resource allocation. Their proposed solution, **Multi-Objective ABCQ (MOABCQ)**, combines the **Artificial Bee Colony Algorithm** with Q-learning to optimize task scheduling, resource utilization, and load balancing in cloud environments. The MOABCQ framework demonstrated exceptional performance in achieving these goals, highlighting the advantages of integrating metaheuristic algorithms with reinforcement learning techniques for cloud computing challenges.

The integration of Q-learning with the Artificial Bee Colony (ABC) algorithm significantly enhances its efficiency, leading to improved throughput and a higher **Average Resource Utilization Ratio (ARUR)** compared to other algorithms. Figure 5 illustrates the growing trend of hybrid techniques observed in recent literature, highlighting the fusion of diverse methodologies to address complex challenges.

Among these, **Particle Swarm Optimization (PSO)**, a swarm-based technique, has gained considerable attention for solving load-balancing issues in cloud computing. By combining PSO with other advanced methods, researchers aim to achieve optimal solutions through thorough exploration and investigation of the search space. For instance, Panwar et al. (2019) proposed a **TOPSIS-PSO** method for non-preemptive task scheduling in cloud systems. This approach uses the **TOPSIS method** to evaluate tasks based on execution time, transmission time, and cost, followed by optimization using PSO. The method effectively optimizes key metrics such as **Makespan**, execution time, transmission time, and cost.

In 2020, Agarwal et al. introduced a **Mutation-based PSO algorithm** to address challenges like premature convergence, reduced convergence speed, and local optima entrapment. This approach aims to minimize performance metrics such as **Makespan time** while enhancing the fitness function in cloud environments. Similarly, in 2021, Negi et al. developed a hybrid load-balancing algorithm called **CMODLB**, which integrates machine learning and soft computing techniques. The CMODLB method leverages **artificial neural networks**, **fuzzy logic**, and **clustering techniques** to achieve efficient load balancing in cloud computing systems. These advancements demonstrate the effectiveness of combining traditional optimization methods with modern computational techniques to address complex cloud computing challenges.

To achieve even workload distribution, the system employs **Bayesian optimization-augmented K-means** for virtual machine (VM) clustering and the **TOPSIS-PSO method** for task scheduling. VM migration decisions are made using an **interval type-2 fuzzy logic system** that considers load conditions. While these algorithms have shown strong performance, they do not account for the specific types of content used by users. Adil et al. (2022) highlighted that incorporating knowledge about task content types can significantly improve scheduling efficiency and reduce VM workload. Their **PSO-CALBA system** categorizes user tasks into content types such as video, audio, image, and text using a **Support Vector Machine (SVM) classifier**. The classification process begins by selecting file fragments, which are tasks composed of diverse content types. The initial classification stage uses the **Radial Basis Function (RBF) kernel** to handle high-dimensional data, addressing a significant challenge in the process.

Pradhan et al. (2022) tackled the issue of managing complex, high-dimensional data in cloud environments by combining **Deep Reinforcement Learning (DRL)** with **parallel Particle Swarm Optimization (PSO)**. Their proposed method synergistically integrates PSO and DRL to optimize rewards by minimizing both **makespan time** and **energy consumption**, while ensuring high accuracy and fast execution. The algorithm iteratively improves accuracy, demonstrating superior performance in dynamic environments and effectively handling diverse tasks in cloud settings. Similarly, Jena et al. (2022) introduced the **QMPSO algorithm**, which evenly distributes workloads across VMs, improving metrics such as **makespan**, **throughput**, **energy utilization**, and **task waiting time**. The QMPSO algorithm enhances performance by modifying velocity based on the best action generated through **Q-learning**, employing dynamic resource allocation to distribute tasks among VMs with varying priorities. This approach minimizes task waiting time and maximizes VM throughput, making it highly efficient for independent tasks.

In **Fog computing**, load balancing is particularly challenging due to limited resources. Talaat et al. (2022) proposed an **Effective Dynamic Load Balancing (EDLB)** method that uses **Convolutional Neural Networks (CNN)** and **Multi-Objective Particle Swarm Optimization (MPSO)** to optimize resource allocation and maximize resource utilization. The EDLB system consists of three main modules: the **Fog Resource Monitor (FRM)**, the **CNN-based Classifier (CBC)**, and the **Optimized Dynamic Scheduler (ODS)**. The FRM monitors server resource utilization, the CBC classifies fog servers, and the ODS allocates incoming tasks to the most suitable server, reducing response time and improving resource utilization. This approach effectively minimizes response time while enhancing system efficiency.

Nabi et al. (2022) introduced an **Adaptive PSO-Based Task Scheduling Approach** for cloud computing, focusing on achieving load balance and optimization. Their solution incorporates a **Linearly Descending and Adaptive Inertia Weight (LDAIW)** technique to improve job scheduling efficiency. Inspired by swarm intelligence, the method uses a population-based scheduling system where particles represent solutions, and their updates are influenced by factors such as **inertia weight**, **personal best**, and **global best**. This approach reduces task execution time, increases throughput, and balances local and global search effectively.

Systematic Literature Survey Review Summary Table-1

Year	Authors	Technique/Algorithm	Key Contribution	Outcome/Advantages
2014	Zhan et al.	Load-Aware Genetic Algorithm (LAGA)	Enhanced GA with TLB model for makespan optimization and load balancing.	Improved fitness function for optimal scheduling and load balance.
2019	Rekha and Dakshayini	Genetic Algorithm (GA) for task allocation	Task allocation method to minimize job completion time and improve system performance.	Superior throughput, energy efficiency, and response time optimization.
2019	Ran et al.	Deep Reinforcement Learning (DRL)	DRL-based task scheduling for dynamic resource allocation.	Reduced average response time, balanced workload distribution, and SLA compliance.
2020	Farrag et al.	Ant-Lion Optimizer (ALO) & Grey Wolf Optimizer (GWO)	Job scheduling optimization for makespan reduction.	Outperformed Firefly Algorithm (FFA) in makespan reduction.
2020	Jyoti and Shrimali	Multi-Agent DRL-Dynamic Resource Allocation (MADRL-DRA)	Integrated LUA and DOLASB for dynamic resource allocation.	Improved QoS metrics: execution time, waiting time, energy efficiency, throughput, and resource utilization.
2021	Swarup et al.	Clipped Double Deep Q-learning	DRL-based job scheduling for computational cost reduction.	Balanced exploration-exploitation using ε-greedy policy, optimized resource allocation.
2021	Tong et al.	Deep Deterministic Multi-Agent Task Scheduling (DDMTS)	DRL-based task scheduling to minimize VM load imbalance and job rejection rates.	Improved stability, reduced Degree of Imbalance (DI), and lower job rejection rates.
2021	Negi et al.	CMODLB (Hybrid Load-Balancing Algorithm)	Combined machine learning and soft	Efficient load balancing using ANN, fuzzy logic, and

			computing for load balancing.	clustering techniques.
2022	Reddy et al.	AVS-PGWO-RDA Framework	Integrated Probabilistic GWO for task selection and resource allocation.	Optimized fitness value, reduced task complexity, and improved execution time.
2022	Adil et al.	PSO-CALBA System	SVM-based task categorization for improved scheduling efficiency.	Enhanced scheduling efficiency by considering task content types (video, audio, image, text).
2022	Pradhan et al.	Parallel PSO with DRL	Combined PSO and DRL for high-dimensional data management.	Minimized makespan time and energy consumption, improved accuracy and execution speed.
2022	Jena et al.	QMPSO Algorithm	Hybrid of Modified PSO and Q-learning for task scheduling.	Improved makespan, throughput, energy utilization, and reduced task waiting time.
2022	Talaat et al.	Effective Dynamic Load Balancing (EDLB)	CNN and MPSO for resource allocation in Fog computing.	Reduced response time, improved resource utilization, and efficient task allocation.
2022	Nabi et al.	Adaptive PSO with LDAIW	Adaptive PSO for load balancing and task scheduling.	Reduced task execution time, increased throughput, and balanced local-global search.
2023	Janakiraman and Priya	HGWIPSOA (Hybrid GWO and PSO)	Combined GWO and PSO for task allocation.	Improved scheduling accuracy, faster convergence, and minimized makespan.
2023	Mishra and Majhi	GAYA (Hybrid GA and JAYA)	Hybrid metaheuristic for dynamic task scheduling.	Enhanced exploration and exploitation for scheduling medical data in cloud environments.
2024	Behera and Sobhanayak	Hybrid GWO-GA	Combined GWO and GA for large-scale task scheduling.	Faster convergence, minimized makespan, and reduced energy consumption.
2024	Brahmam and Vijay Anand	VMMISD Model	Integrated GA and ACO for resource allocation and load balancing.	Improved load balancing efficiency using K-means, Fuzzy Logic, LSTM, and Graph Networks.
2024	Joshi et al.	Autoscaling with Reinforcement Learning (RL)	Dynamic resource allocation using DRL for workload prediction.	Real-time decision-making, adaptive resource distribution, and improved system performance.

Research Gap:

In cloud computing, fault tolerance and load balancing are critical for ensuring system reliability and performance. While existing research has explored these areas individually, there is a notable gap in studies that integrate fault tolerance mechanisms with load balancing strategies, particularly concerning security implications.

A systematic literature review by Behera and Sobhanyayak (2024) highlights the importance of fault-tolerant load balancing but does not delve into the security aspects of this integration. Similarly, Tawfeeg et al. (2022) and Brahmam and Vijay Anand (2024) discuss load balancing and fault tolerance techniques without extensively examining their interconnected security concerns.

Recent studies have begun to address this intersection. For instance, a 2023 paper presents a novel fault-tolerant load balancing technique that adaptively monitors system health to detect and handle faults, aiming to prevent network congestion. This approach combines Ant Colony Optimization with active clustering to distribute loads evenly across data centers, enhancing both performance and fault tolerance.

Despite these advancements, there remains a scarcity of research focusing on the security vulnerabilities that may arise from integrating fault tolerance with load balancing. For example, a 2024 study revealed a vulnerability in Amazon Web Service's Application Load Balancer that could allow attackers to bypass access controls due to user implementation issues. This highlights the need for comprehensive strategies that address both fault tolerance and security in load balancing mechanisms.

In summary, while fault tolerance and load balancing are well-studied individually, there is a significant research gap in exploring their integration, especially concerning security implications. Future research should focus on developing comprehensive frameworks that seamlessly combine these aspects to enhance the resilience and security of cloud computing services.

Conclusion:

Our comprehensive literature review has uncovered significant insights and emerging trends that are essential for driving advancements in cloud computing technology. This discussion synthesizes the research findings, addressing the initial research questions and drawing conclusions based on an in-depth analysis of selected studies conducted from 2014 to 2024.

References:

[1] Konjaang JK, Ayob FH, Muhammed A (2018) Cost effective Expa-Max-Min scientific workflow allocation and load balancing strategy in cloud computing. J Comput Sci 14(5):623–638. https:// doi. org/ 10. 3844/jcssp. 2018. 623. 638

[2] Velpula P, Pamula R, Jain PK, Shaik A (2022) Heterogeneous load balancing using predictive load summarization. Wirel Pers Commun 125(2):1075–1093. https:// doi. org/ 10. 1007/ s11277- 022- 09589-y

[3] Hung TC, Hy PT, Hieu LN, Phi NX (2019) MMSIA: improved max-min scheduling algorithm for load balancing on cloud computing. In: presented at the ACM International Conference Proceeding Series, pp 60–64 https:// doi. org/ 10. 1145/ 33109 86. 33110 17

[4] Farrag AAS, Mohamad SA, El-Horbaty ESM (2020) Swarm optimization for solving load balancing in cloud computing. In: presented at the advances in intelligent systems and computing, pp 102–113 https:// doi. org/ 10. 1007/ 978-3- 030- 14118-9_ 11

[5] Seth S, Singh N (2019) Dynamic heterogeneous shortest job first (DHSJF): a task scheduling approach for heterogeneous cloud computing systems. Int J Inf Technol Singap 11(4):653–657. https:// doi. org/ 10. 1007/s41870- 018- 0156-6

[6] Reddy KL, Lathigara A, Aluvalu R, Viswanadhula UM (2022) PGWO-AVS-RDA: An intelligent optimization and clustering based load balancing model in cloud. Concurr Comput Pract Exp. https:// doi. org/ 10. 1002/cpe. 7136

[7] Janakiraman S, Priya MD (2023) Hybrid grey wolf and improved particle swarm optimization with adaptive intertial weight-based multi-dimensional learning strategy for load balancing in cloud environments.Sustain Comput Inform Syst. https:// doi. org/ 10. 1016/j. suscom. 2023. 100875

[8] Behera I, Sobhanayak S (2024) Task scheduling optimization in heterogeneous cloud computing environments: a hybrid GA-GWO approach. J Parallel Distrib Comput. https:// doi. org/ 10. 1016/j. jpdc. 2023.104766

[9] Joshi S, Panday N, Mishra A (2024) Reinforcement learning based auto scaling strategy used in cloud environment:tate of Art, p 736 https:// doi. org/ 10. 1109/ CSNT6 0213. 2024. 10545 922

[10] Rekha PM, Dakshayini M (2019) Efficient task allocation approach using genetic algorithm for cloud environment.Clust Comput 22(4):1241–1251. https:// doi. org/ 10. 1007/ s10586- 019- 02909-1
[11] Mishra K, Majhi SK (2023) A novel improved hybrid optimization algorithm for efficient dynamic medical data scheduling in cloud-based systems for biomedical applications. Multimed Tools Appl 82(18):27087–27121. https:// doi. org/ 10. 1007/ s11042- 023- 14448-4
[12] Brahmam MG, Vijay Anand R (2024) VMMISD: an efficient load balancing model for virtual machine migrations via fused metaheuristics with iterative security measures and deep learning optimizations.IEEE Access. https:// doi. org/ 10. 1109/ ACCESS. 2024. 33734 65
[13] Ran L, Shi X, Shang M (2019) SLAs-aware online task scheduling based on deep reinforcement learning method in cloud environment. In: presented at the Proceedings - 21st IEEE International Conference on High Performance Computing and Communications, 17th IEEE International Conference on Smart City and 5th IEEE International Conference on Data Science and Systems, HPCC/SmartCity/DSS 2019, pp 1518–1525 https:// doi. org/ 10. 1109/ HPCC/ Smart City/ DSS. 2019. 00209
[14] Jyoti A, Shrimali M (2020) Dynamic provisioning of resources based on load balancing and service broker policy in cloud computing. Clust Comput 23(1):377–395. https:// doi. org/ 10. 1007/ s10586- 019- 02928-y
[15] Tong Z, Deng X, Chen H, Mei J (2021) DDMTS: a novel dynamic load balancing scheduling scheme under SLA constraints in cloud computing. J Parallel Distrib Comput 149:138–148. https:// doi. org/ 10. 1016/j.jpdc. 2020. 11. 007
[16] Swarup S, Shakshuki EM, Yasar A (2021) Task scheduling in cloud using deep reinforcement learning. In: presented at the Procedia Computer Science, pp 42–51 https:// doi. org/ 10. 1016/j. procs. 2021. 03. 016
[17] Mao Y, Chen X, Li X (2014) Max–min task scheduling algorithm for load balance in cloud computing. Adv Intell Syst Comput 255:457–465. https:// doi. org/ 10. 1007/ 978- 81- 322- 1759- 6_ 53
[18] Panwar N, Negi S, Rauthan MMS, Vaisla KS (2019) TOPSIS–PSO inspired non-preemptive tasks scheduling algorithm in cloud environment. Clust Comput 22(4):1379–1396. https:// doi. org/ 10. 1007/s10586- 019- 02915-3
[19] Agarwal R, Baghel N, Khan MA (2020) Load balancing in cloud computing using mutation based particle swarm optimization. In: presented at the 2020 International Conference on Contemporary Computing and Applications, IC3A 2020, pp 191–195 https:// doi. org/ 10. 1109/ IC3A4 8958. 2020. 233295
[20] Ni L, Sun X, Li X, Zhang J (2021) GCWOAS2: multiobjective task scheduling strategy based on Gaussian cloud-whale optimization in cloud computing. Comput Intell Neurosci. https:// doi. org/ 10. 1155/2021/ 55467 58
[21] Adil M, Nabi S, Raza S (2022) PSO-CALBA: Particle swarm optimization based content-aware load balancing algorithm in cloud computing environment. Comput Inform 41(5):1157–1185. https:// doi. org/10. 31577/ cai_ 2022_5_ 1157
[22] Pradhan A, Bisoy SK, Kautish S, Jasser MB, Mohamed AW (2022) Intelligent decision-making of load balancing using deep reinforcement learning and parallel PSO in cloud environment. IEEE Access 10:76939–76952. https:// doi. org/ 10. 1109/ ACCESS. 2022. 31926 28
[23] Jena UK, Das PK, Kabat MR (2022) Hybridization of meta-heuristic algorithm for load balancing in cloud computing environment. J King Saud Univ Comput Inf Sci 34(6):2332–2342. https:// doi. org/ 10. 1016/j.jksuci. 2020. 01. 012
[24] Talaat FM, Ali HA, Saraya MS, Saleh AI (2022) Effective scheduling algorithm for load balancing in fog environment using CNN and MPSO. Knowl Inf Syst 64(3):773–797. https:// doi. org/ 10. 1007/s10115- 021- 01649-2
[25] Nabi S, Ahmad M, Ibrahim M, Hamam H (2022) AdPSO: adaptive PSO-based task scheduling approach for cloud computing. Sensors. https:// doi. org/ 10. 3390/ s2203 0920

[26] Rostami S, Broumandnia A, Khademzadeh A (2024) An energy-efficient task scheduling method for heterogeneous cloud computing systems using capuchin search and inverted ant colony optimization algorithm. J Supercomput 80(6):7812–7848. https:// doi. org/ 10. 1007/ s11227- 023- 05725-y

[27] Saba T, Rehman A, Haseeb K, Alam T, Jeon G (2023) Cloud-edge load balancing distributed protocol for IoE services using swarm intelligence. Clust Comput 26(5):2921–2931. https:// doi. org/ 10. 1007/s10586- 022- 03916-5