Optimization of the Energy-Saving Data Storage Algorithm for Differentiated Cloud Computing Tasks

Optimization of the Energy-Saving Data Storage Algorithm

Peichen Zhao
School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China

***Abstract*—This study presents a novel energy-saving data storage algorithm designed to enhance data storage efficiency and reduce energy consumption in cloud computing environments. By intelligently discerning and categorizing various cloud computing tasks, the algorithm dynamically adapts data storage strategies, resulting in a targeted optimization methodology that is both devised and experimentally validated. The study findings demonstrate that the optimized model surpasses comparative models in accuracy, precision, recall, and F1-score, achieving peak values of 0.863, 0.812, 0.784, and 0.798, respectively, thereby affirming the efficacy of the optimized approach. In simulation experiments involving tasks with varying data volumes, the optimized model consistently exhibits lower latency compared to Attention-based Long Short-Term Memory Encoder-Decoder Network and Deep Reinforcement Learning Task Scheduling models. Furthermore, across tasks with differing data volumes, the optimized model maintains high throughput levels, with only marginal reductions in throughput as data volume increases, indicating sustained and stable performance. Consequently, this study is pertinent to cloud computing data storage and energy-saving optimization, offering valuable insights for future research and practical applications.**

***Keywords*—Energy-saving data storage algorithm; differentiated task recognition; cloud computing; intelligent storage strategy; data classification and distribution**

I. INTRODUCTION

With the rapid development of cloud computing, the demand for data storage has surged, rendering traditional centralized storage solutions inadequate for managing vast data volumes and diverse task requirements [1]. The data storage landscape in cloud computing is characterized by its large scale and the variety of data types, encompassing different task types and service models, such as big data analysis, real-time processing, and archival storage. Each task possesses unique resource requirements and storage characteristics, leading to significant variations in performance requirements across tasks [2-4]. To enhance overall performance, data storage algorithms in cloud computing must balance data accuracy, reliability, real-time performance, and energy consumption. However, existing storage strategies frequently overlook the distinctions between diverse tasks, resulting in wasted storage resources, performance degradation, and elevated energy consumption.

Previous studies have highlighted the substantial differences in resource requirements among various task types in cloud computing environments. For instance, Yang et al. demonstrated that these disparities significantly influence the selection and optimization of data storage strategies [5]. Saravanan et al. conducted an analysis of big data and real-time processing tasks, emphasizing fundamental differences in data access patterns and response time requirements, thereby underscoring the necessity for targeted algorithm design [6]. Additionally, Manukumar and Muthuswamy elucidated the variations in persistence and reliability requirements between archival storage and high-frequency access data, providing a theoretical foundation for the diversification of data storage strategies [7]. Furthermore, Hamid et al. proposed an energy-saving storage algorithm predicated on data access frequency, which markedly reduced energy consumption in data centers through intelligent data migration and caching strategies [8]. Zhang et al. developed a storage management system based on data hotness and task priority, effectively enhancing the utilization of storage resources and decreasing the energy consumption associated with redundant data [9]. Finally, Rahimikhanghah et al. conducted simulation experiments to compare the performance of various data storage algorithms under diverse cloud computing tasks, revealing that targeted optimization algorithms exhibit significant advantages in terms of accuracy and response time [10]. El-Menbawy et al. validated the improvements in throughput and storage efficiency of their proposed energy-saving storage algorithm through testing in a real cloud environment, providing robust data to support its practical application [11]. Dong et al. assessed the performance of various storage algorithms using metrics such as accuracy, recall, F1-score, and precision. Their findings indicated that algorithms that comprehensively accounted for task differences demonstrated superior performance across multiple indicators [12]. Al-Masri et al. applied machine learning techniques to predict and adjust the performance of storage systems, optimizing data distribution and replication strategies while minimizing energy consumption, all while maintaining data reliability [13].

Although previous studies have recognized the differences in cloud computing tasks, many have lacked in-depth analysis of their specific characteristics and requirements, leading to suboptimal performance of storage algorithms when faced with diverse task types. Moreover, most energy-saving storage algorithms are typically designed for specific tasks or scenarios, making it difficult for them to adapt flexibly to varied task environments, thereby diminishing overall performance and energy-saving effectiveness. By conducting a detailed analysis of the data characteristics and performance requirements of different cloud computing tasks, this study provides a clear optimization framework and decision-making basis for the design of energy-saving data storage algorithms. Additionally, the proposed algorithm can identify and adapt to diverse task environments, dynamically adjusting data storage strategies according to task types, thereby achieving an optimal balance between energy efficiency and performance under different task scenarios. This study first analyzes the differences in cloud computing tasks, revealing that a detailed analysis and understanding of these tasks can provide clearer directions for optimizing energy-saving data storage algorithms, enabling them to adapt to task variability and achieve a balance between energy efficiency and performance in diverse task environments. Secondly, the design of the energy-saving data storage algorithm is studied, emphasizing that through the comprehensive application of these strategies, the algorithm can effectively meet the demands of different cloud computing tasks, ensuring an optimal balance between performance, reliability, and energy consumption in data storage systems. Furthermore, by incorporating strategies for data distribution, load balancing, data replication, fault tolerance, data access, migration, and energy consumption optimization with performance evaluation, the model is refined. Finally, the effectiveness of the model is validated through experiments.

II. Optimization of the Energy-Saving Data Storage Algorithm for Differentiated Cloud Computing Tasks

A. Analysis of Differentiated Cloud Computing Tasks

Before designing and optimizing energy-saving data storage algorithms tailored to differentiated cloud computing tasks, a comprehensive analysis of the characteristics and requirements of each task is essential. Additionally, it is critical to establish a clear mapping between data storage models and task types, while identifying the key factors that influence the selection of storage algorithms [14-16]. The variety of cloud computing tasks is considerable, and the common task types are presented in Table I:

TABLE I. Types of Cloud Computing Tasks

Task types	Description
Real-time processing tasks	Real-time tasks, such as financial transactions and online games, require extremely low latency to guarantee a quick response to user requests. They often rely on caching strategies that ensure efficient reading and immediate updating of data.
Big data analysis tasks	They involve large-scale data processing and analysis, such as data mining, machine learning, and business intelligence. These tasks require high storage performance and require fast data transfer between distributed storage systems to meet computing and analysis requirements.
Archive storage tasks	For instance, long-term storage of logs, backups, and legal files requires high data persistence and security, but relatively low access speed. Data backup, fault tolerance, and tiered storage are often required to balance cost and performance.
Content delivery tasks	Examples include video streaming and large file downloads, emphasizing high throughput and efficiency of data distribution. Content Delivery Network (CDN) and multi-layer caching are needed to accelerate content distribution.

A well-designed data storage model is crucial for meeting the performance requirements of different tasks. Cache modes are commonly employed for real-time processing and content delivery tasks, with the primary goal of minimizing data access latency while dynamically adjusting data distribution across various cache layers. Distributed file systems are typically utilized in large-scale data processing and transmission, ensuring reliable data distribution and fast access for big data analysis tasks [17-18]. For archival storage tasks, hierarchical storage strategies are applied, utilizing cold and hot data layers to manage data accessed at varying frequencies, thereby reducing overall storage costs while maintaining data persistence and security. Replication strategies are adapted based on task characteristics: real-time processing tasks often require leader-follower replication to ensure low-latency responses, while big data analysis tasks generally employ distributed replication with high redundancy to guarantee data reliability and persistence [19-21]. The selection and optimization of storage algorithms are influenced by a range of critical factors, depending on the task types and storage modes, as summarized in Table II:

TABLE II. Key Factors Influencing the Selection of Storage Algorithms

Factor	Analysis
Task priority	Various tasks have different priorities in the system. High-priority tasks such as financial transactions should be allocated more resources, while low-priority archiving tasks can use more energy-efficient storage strategies.
Data access mode	The data access mode of the task directly affects the design of the storage algorithm. The frequent read and write requirements of real-time processing tasks are different from the sequential batch processing of big data analysis tasks, and differentiated caching and migration strategies are required.
Data consistency	Different tasks have diverse requirements for data consistency. Real-time tasks require strong consistency, while analytical tasks can accept some degree of ultimate consistency.
Energy consumption and cost	Low energy consumption and storage costs are the focus of most missions. According to the performance and budget requirements of different tasks, a proper storage strategy can effectively reduce the cost of data migration and redundant copies.

A detailed analysis and understanding of various cloud computing tasks can provide clearer guidance for optimizing energy-saving data storage algorithms. This allows them to adapt to task variability and achieve a balance between energy efficiency and performance across diverse task environments [22].

B. Design of the Energy-Saving Data Storage Algorithm

The design of the energy-saving data storage algorithm adheres to the following principles and objectives. First, the algorithm must be adaptive, capable of recognizing and responding to the diverse requirements of different cloud computing tasks, and dynamically adjusting data storage and distribution strategies based on the tasks' characteristics and priorities. Second, reducing energy consumption in data centers through efficient data migration and replication strategies is a primary goal, focusing on minimizing hardware resource idleness and reducing redundant data replication. Third, the algorithm must ensure data availability and persistence across various task scenarios, maintaining data consistency even under high-load conditions. Lastly, it is essential to balance performance metrics such as response time, throughput, and accuracy while addressing the diverse demands of different tasks. This study proposes an energy-saving data storage algorithm designed to accommodate the diversity of cloud computing tasks while minimizing energy consumption. The framework of the proposed algorithm comprises several key modules, as illustrated in Fig. 1.

The identification and classification of differentiated task data form the foundational basis of the energy-saving storage algorithm. By analyzing task-specific characteristics, such as real-time requirements, priority levels, and data access modes, the data can be categorized into distinct types, as outlined in Table III:

TABLE III. DATA TYPES

Type	Analysis
High-priority real-time data	This type of data is used for real-time processing tasks, usually has high priority and low latency requirements, and is mainly stored in the cache layer to meet fast access demands.
Batch analysis data	It is suitable for big data analysis tasks that need to maintain the reliability and distribution of data in a distributed file system to ensure efficient computation and analysis.
Long-term archived data	The data access frequency is low and data is stored at the cold layer. The hierarchical storage strategy minimizes space and power consumption.

Building on the identification and classification of differentiated task data, appropriate energy-saving data storage strategies can be developed. First, the intelligent data migration strategy dynamically adjusts the distribution of data across various storage system tiers based on task characteristics and data access patterns. This ensures that high-frequency data is cached, while low-frequency data is archived, optimizing the hierarchical management of data. Second, the replication strategy is determined by task type and priority. High-priority tasks utilize a leader-follower replication model to meet low-latency response requirements, whereas batch analysis tasks implement a multi-replica distribution strategy to guarantee data reliability and consistency. In addition, the resource balancing strategy employs load-balancing algorithms to distribute access loads evenly across data nodes, mitigating resource idleness and avoiding bottlenecks caused by hot data, thereby enhancing resource utilization. Moreover, the dynamic optimization strategy continuously monitors the storage system's performance and energy consumption in real-time, using machine learning algorithms to optimize data storage strategies and adapt to evolving task demands. Through the comprehensive implementation of these energy-saving data storage strategies, the proposed algorithm effectively addresses the requirements of diverse cloud computing tasks, ensuring an optimal balance between performance, reliability, and energy efficiency in the data storage system.

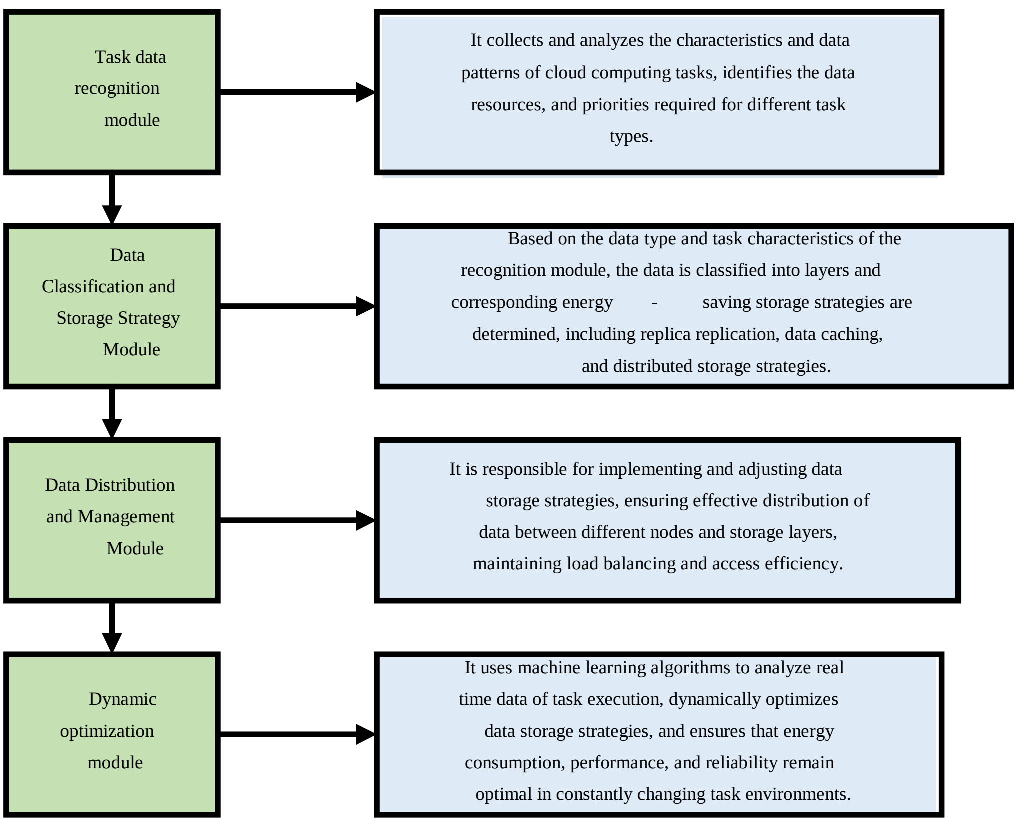


Fig. 1. The framework of energy-saving data storage algorithms.

C. Algorithm Optimization and Implementation

The optimization and implementation of the energy-saving data storage algorithm are designed to enhance the efficiency and reliability of the data storage system while minimizing energy consumption, all without compromising performance. Central to cloud computing data storage are effective data distribution strategies, which significantly influence the system's performance and stability. The algorithm proposed here is founded on three key principles: task priority, data hotness, and load balancing. Task data is allocated to various storage nodes based on the real-time nature and priority of the tasks, facilitating hierarchical management of data access. High-priority tasks with stringent real-time requirements are directed to nodes that offer faster response times, while lower-priority tasks are assigned to nodes with slower response capabilities. By analyzing the frequency and hotness of data access, high-demand (hot) data is stored in the high-speed cache layer to ensure efficient access, whereas low-demand (cold) data is migrated to the cold storage layer to alleviate pressure on the cache. The load balancing algorithm dynamically adjusts the data distribution across nodes to achieve an equitable distribution of access loads among all storage nodes, thus preventing resource idleness and mitigating the occurrence of bottlenecks due to hot data. Furthermore, replication and fault tolerance mechanisms within the data storage framework ensure data persistence and reliability, as illustrated in Table IV.

TABLE IV. DATA REPLICAS AND FAULT TOLERANCE MECHANISMS

Dimension	Analysis
Copy replication strategy	According to the requirements of different tasks, various copy replication strategies are designed. Leader-follower replication is adopted for high-priority tasks to ensure real-time performance and fast recovery. Multiple replicas are used for batch processing and archiving tasks to ensure reliable data recovery in case of faults.
Fault tolerance	Regular data integrity checks and replica status monitoring should be implemented to promptly detect and handle storage node failures, restore damaged data to healthy nodes, and ensure data availability and consistency.

Data access and migration strategies significantly influence the performance and flexibility of the data repository. Access optimization employs caching strategies and data layering to refine access pathways in accordance with task type and data popularity, ensuring that frequently accessed data resides in the cache layer while infrequently accessed data is relegated to secondary storage. Dynamic migration adjusts data storage locations in response to evolving access patterns, transferring data between cold and hot layers to meet shifting task requirements and sustain optimal system performance. In pursuit of a comprehensive reduction in energy consumption and an evaluation of storage system performance, this study delineates the following strategies, as outlined in Table V:

TABLE V. OPTIMIZATION STRATEGY

Strategy	Analysis
Energy consumption optimization	By adjusting data distribution, and reducing node idle rates and redundant copies, unnecessary energy consumption is reduced. Meanwhile, energy-saving storage hardware and resource hibernation mechanisms are employed to mitigate system energy consumption while maintaining high performance.
Performance evaluation	Comprehensive performance evaluation indicators are established to identify system bottlenecks and optimize storage strategies, thereby achieving a balance between performance and energy consumption.

By integrating a comprehensive array of data distribution strategies, load balancing, data duplication, fault tolerance mechanisms, data access, migration strategies, energy consumption optimization, and performance assessment, the algorithm proposed herein ensures that the data storage system achieves an optimal amalgamation of high performance and low energy consumption across various task environments.

III. ANALYSIS OF PERFORMANCE AND SIMULATION RESULTS OF ENERGY-SAVING STORAGE ALGORITHMS

A. Analysis of Performance Comparison Results of Energy-Saving Storage Algorithms

The dataset utilized for this experiment is the Alibaba Cluster dataset, a comprehensive resource derived from Alibaba's production cluster, specifically designed to facilitate research on cluster management. The dataset spans multiple versions from 2017 to 2023, offering valuable insights into various facets of cloud computing tasks. It is publicly accessible via the official repository at https://github.com/alibaba/clusterdata. Details of the experimental environment are presented in Table VI.

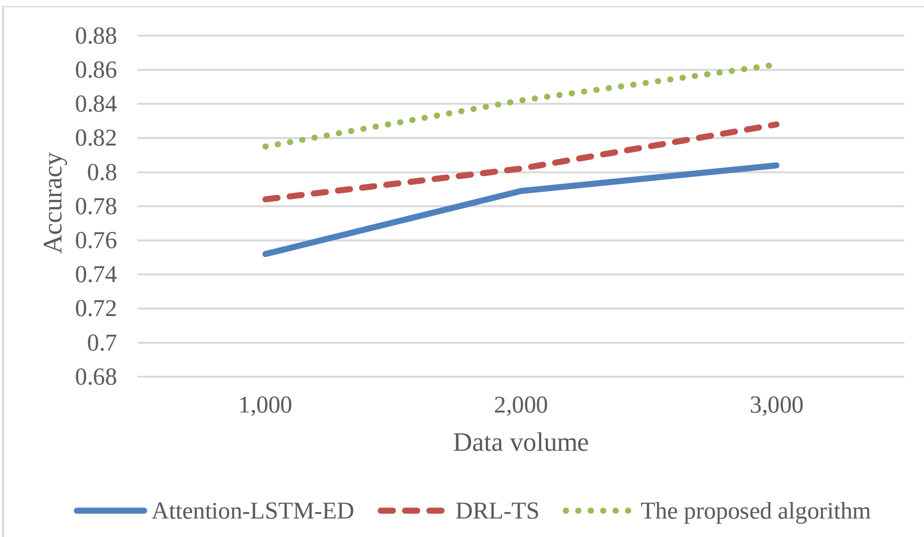
TABLE VI. EXPERIMENTAL ENVIRONMENT

Equipment type	Parameter configuration
Processor	Inter(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz
Graphics card	NVIDIA Titan Xp 12GB
Memory	128 GB
Operating system	Ubuntu 16.04 LTS
Programming language	Python 3.6

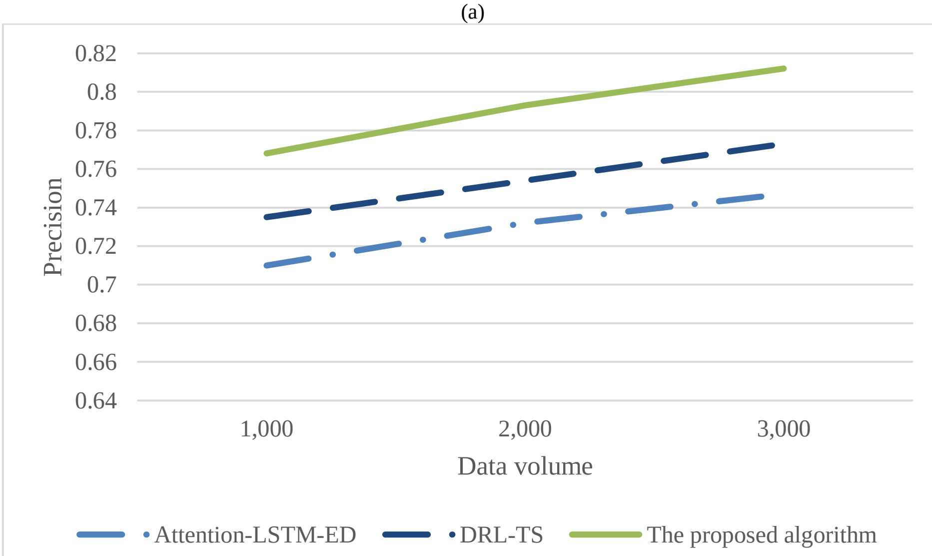
The model parameters are uniformly configured to ensure experimental accuracy. Specifically, the learning rate is set to 0.01, with a batch size of 64, two hidden layers, each containing 128 hidden units. The Adam optimizer is employed, and training is conducted over 50 epochs with a dropout probability of 0.02. For comparative analysis, the Attention-based Long Short-Term Memory Encoder-Decoder (Attention-LSTM-ED) and Deep Reinforcement Learning Task Scheduling (DRL-TS) models are selected due to their representative and practical applications within the cloud computing domain. The Attention-LSTM-ED model leverages the LSTM network and attention mechanism to effectively handle time series data, making it suitable for applications such as task load prediction. In contrast, the DRL-TS model utilizes deep reinforcement learning for task scheduling, offering an efficient solution to challenges related to resource allocation and task scheduling. These models provide a robust comparison, enabling a thorough evaluation of the proposed model's improvements in terms of performance, efficiency, and energy consumption. The performance metrics compared in this study include accuracy, recall, precision, and F1-score, with the results illustrated in Fig. 2.

Fig. 2 demonstrates that the accuracy of the optimized model reaches 0.815, 0.842, and 0.863 for data volumes of 1000, 2000, and 3000, respectively, consistently surpassing the performance of the Attention-LSTM-ED and DRL-TS models. This suggests that the optimized model maintains high accuracy across varying data sizes, highlighting its superior generalization ability and stability. While the Attention-LSTM-ED and DRL-TS models exhibit relatively strong performance with smaller data volumes, their accuracy decreases as data volume increases, underscoring the optimized model's advantage in handling large-scale datasets.

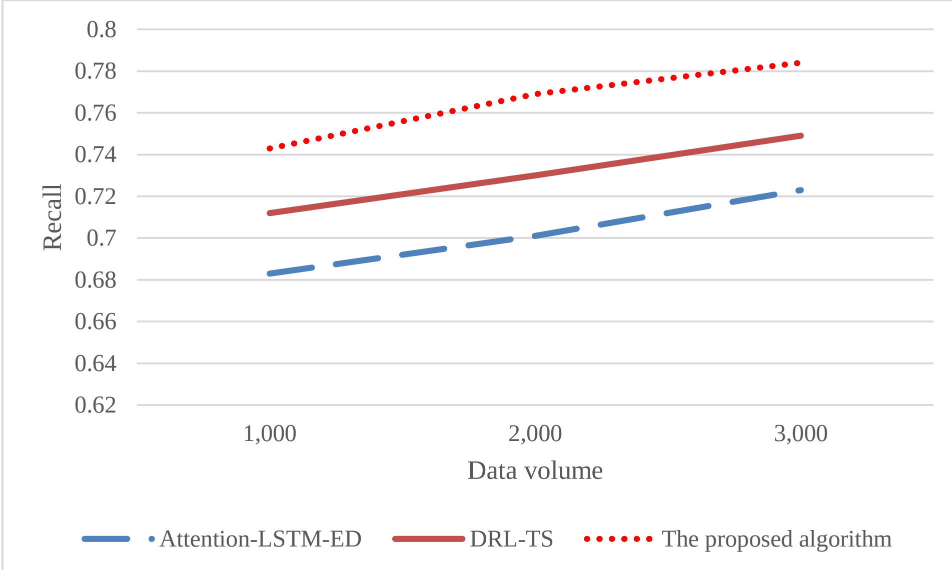
In terms of precision, the optimized model consistently achieves stable and high precision levels of 0.768, 0.793, and 0.812 across different data volumes. In comparison, the precision of the Attention-LSTM-ED and DRL-TS models is 0.710 and 0.735, respectively, at a data volume of 1000, and 0.732 and 0.754, respectively, at a data volume of 2000. Overall, the optimized model outperforms the other two models across all data volumes. This performance advantage likely stems from the optimized model's enhanced data recognition and classification capabilities, which allow it to adapt more effectively to varying data volumes and task requirements. Even at smaller data volumes, the optimized model demonstrates efficient precision, which improves further as data volume increases. In contrast, the precision improvement of the Attention-LSTM-ED and DRL-TS models is relatively slow, suggesting that they may struggle to maintain stable performance as data size grows. These comparative results highlight the significant advantage of the optimized model in terms of precision, particularly in large-scale data environments, where it can more accurately process cloud computing tasks.



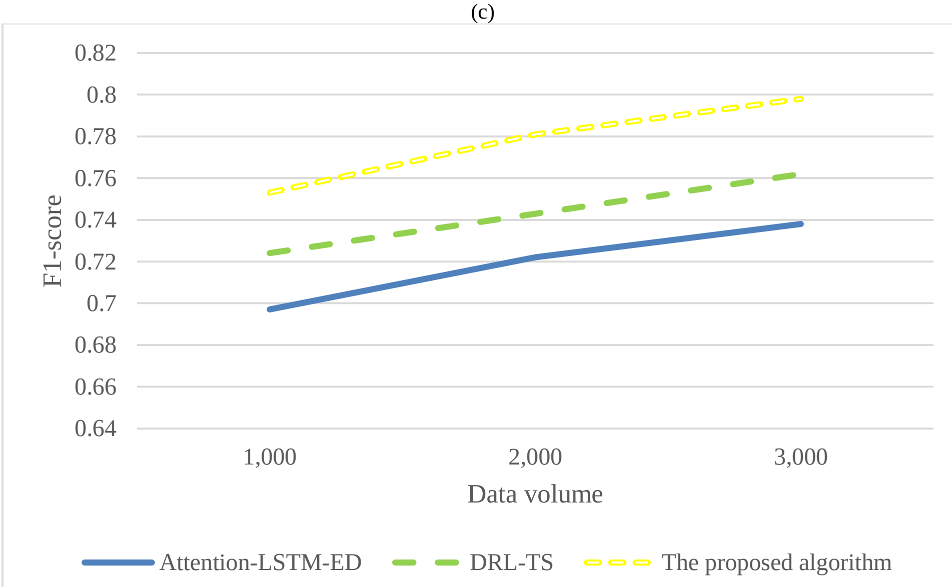
(a)



(b)



(c)



(d)

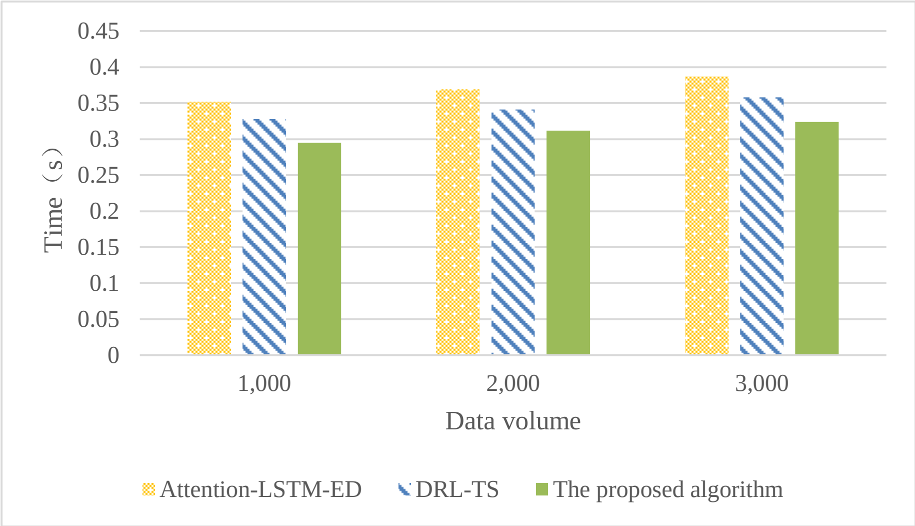
Fig. 2. Performance comparison results (a): Accuracy; (b): Precision; (c): Recall; (d): F1-score.

In terms of recall, the optimized model achieves recall values of 0.743, 0.769, and 0.784 for data volumes of 1000, 2000, and 3000, respectively, consistently outperforming the other models at each data level. For a data volume of 1000, the recall for the Attention-LSTM-ED model is 0.683, while the DRL-TS model achieves 0.712. At a data volume of 2000, their recall values are 0.701 and 0.730, respectively. This consistency indicates that the optimized model exhibits lower sensitivity to varying data scales in terms of recall, allowing it to effectively capture relevant information and enhance data processing performance across a range of task conditions. As data volume increases, the recall of the optimized model improves steadily, demonstrating its robustness and reliability in handling large-scale datasets.

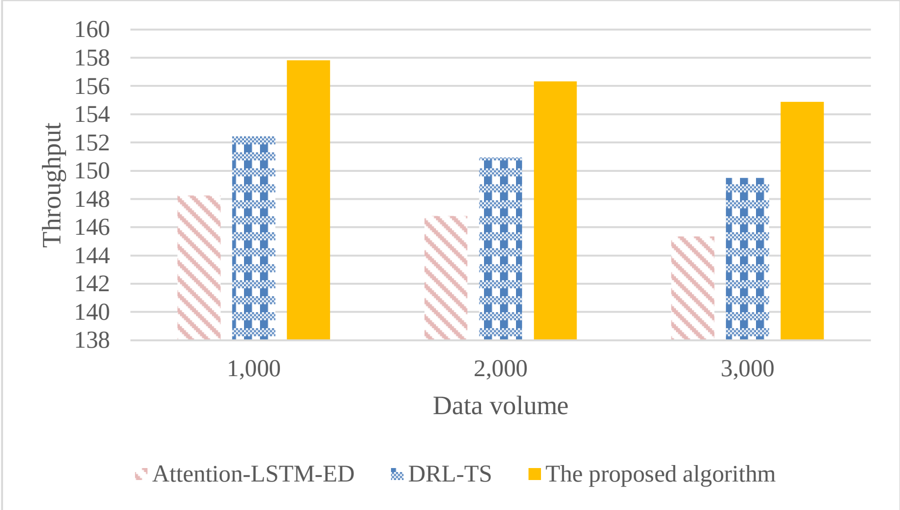
With respect to the F1-score, the optimized model achieves values of 0.753, 0.781, and 0.798 for data volumes of 1000, 2000, and 3000, respectively, outperforming both the Attention-LSTM-ED and DRL-TS models. At a data volume of 1000, the F1-scores for Attention-LSTM-ED and DRL-TS are 0.697 and 0.724, respectively, while at a data volume of 2000, these values are 0.722 and 0.743. The optimized model consistently maintains higher F1-scores across various data volumes, indicating superior performance and stability in task recognition and classification. As data volume increases, the F1-score of the optimized model steadily improves, underscoring its consistency and superiority in managing large-scale data. In contrast, the F1-scores of the Attention-LSTM-ED and DRL-TS models are comparatively lower, particularly at larger data volumes, and their rate of improvement is relatively slow. This suggests that their generalization ability is not as strong as that of the optimized model, and they struggle to maintain the same level of accuracy and stability in environments with expanding data scales and evolving task requirements.

B. Analysis of Simulation Results of Energy-Saving Storage Algorithms

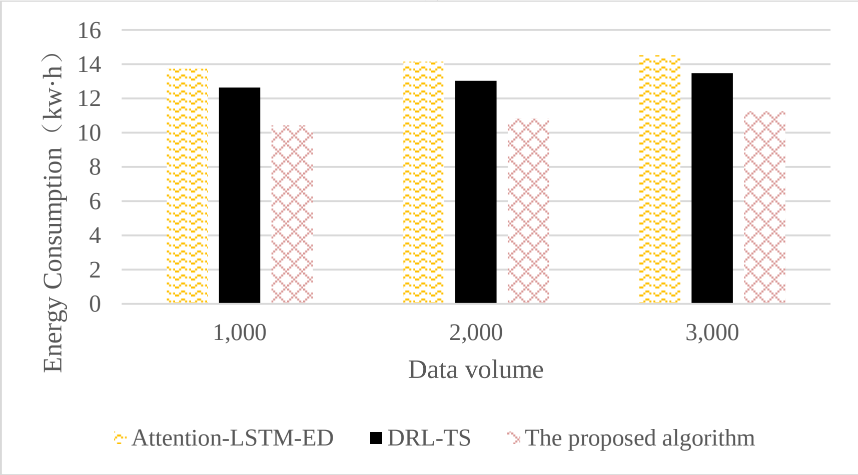
To further validate the effectiveness of the optimized model, simulation experiments are conducted to compare key performance indicators, including delay, throughput, energy consumption, and storage efficiency. The results of these experiments are presented in Fig. 3.



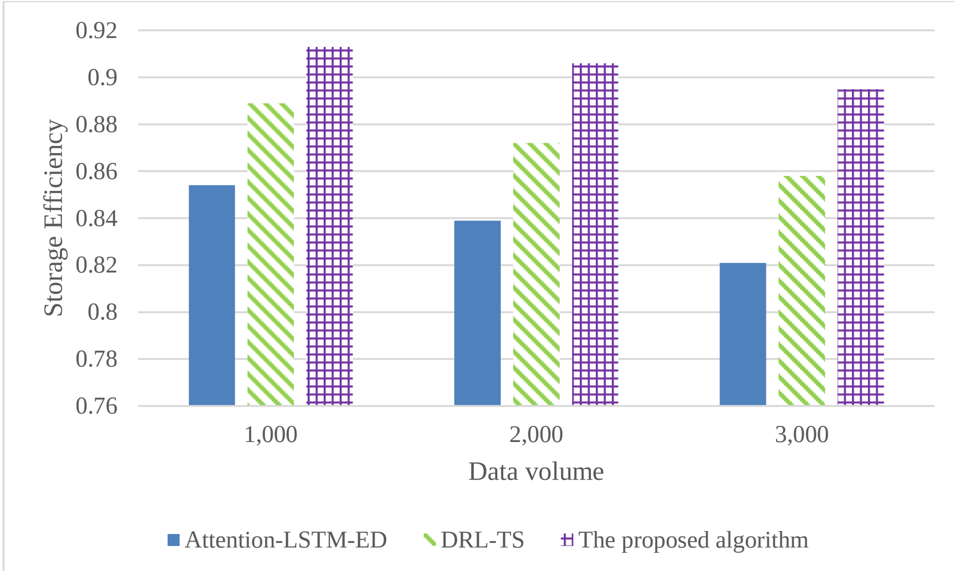
(a)



(b)



(c)



(d)

Fig. 3. Analysis of simulation experiment results (a): Latency; (b): Throughput; (c): Energy Consumption; (d): Storage Efficiency.

Fig. 3 illustrates the delay comparison across different data volumes, demonstrating the superior performance of the optimized model. Specifically, the optimized model exhibits delays of 0.295 seconds, 0.312 seconds, and 0.324 seconds for data volumes of 1000, 2000, and 3000, respectively. At each data volume, its delay is notably lower than that of the Attention-LSTM-ED and DRL-TS models. For instance, at 1000 data volumes, the Attention-LSTM-ED and DRL-TS models report delays of 0.352 and 0.328 seconds, respectively, and at 2000 data volumes, the delays increase to 0.369 and 0.341 seconds. These results underscore the optimized model's significantly lower processing delay, reflecting a marked advantage in task completion speed. Furthermore, as the data size increases, the delay of the optimized model shows only a modest increase, illustrating its strong scalability and adaptability to large-scale data processing. In contrast to the other two models, the optimized model consistently demonstrates superior stability and efficiency in terms of latency. This performance advantage can be attributed to the model's enhanced ability to recognize and classify task data, coupled with its flexible data storage strategies across varying data scales.

In terms of throughput, the optimized model consistently maintains relatively high values across all data volumes. At 1000, 2000, and 3000 data volumes, it achieves throughput values of 157.832, 156.321, and 154.888, respectively, surpassing the Attention-LSTM-ED and DRL-TS models. For instance, at a data volume of 1000, the throughputs of the Attention-LSTM-ED and DRL-TS models are 148.237 and 152.435, and at 2000, they decrease to 146.789 and 150.942, respectively. This analysis reveals that, even as the data volume increases, the optimized model experiences only a slight decline in throughput, maintaining relatively stable performance. Such stability highlights the model's efficiency and ability to adapt to diverse data sizes. By contrast, the Attention-LSTM-ED and DRL-TS models exhibit a more pronounced reduction in throughput, suggesting potential bottlenecks as data volume increases. This further emphasizes the optimized model's distinct advantages in handling large-scale data, attributed to its ability to effectively identify and classify task-specific data while employing energy-efficient storage strategies. As a result, the optimized model meets the demands for high data processing efficiency across varying task requirements.

In the comparison of energy consumption, the optimized model consistently demonstrates lower energy usage across data volumes of 1000, 2000, and 3000, with consumption rates of 10.428 KWH, 10.836 KWH, and 11.257 KWH, respectively. This highlights its significant energy-saving potential. In contrast, the Attention-LSTM-ED model consumes 13.752 KWH at a data volume of 1000, while the DRL-TS model consumes 12.639 KWH. As data volumes increase, both models experience a substantial rise in energy consumption, revealing their inefficiency in managing large datasets. The optimized model, however, exhibits only a modest increase in energy consumption, maintaining consistently low levels across all data volumes. This underscores its advantage and innovation in energy-efficient data storage algorithms. By leveraging advanced storage strategies, the model effectively reduces overall energy consumption while handling complex tasks, striking an optimal balance between performance and energy efficiency. In terms of storage efficiency, the optimized model achieves rates of 0.913, 0.906, and 0.895 for data volumes of 1000, 2000, and 3000, respectively. Its efficiency remains consistently superior to other models under all data volume conditions, demonstrating the clear advantages of the optimization. In comparison, the Attention-LSTM-ED and DRL-TS models exhibit slightly lower efficiency. At a data volume of 1000, the Attention-LSTM-ED model records an efficiency of 0.854, while the DRL-TS model achieves 0.889. As data volumes increase, their efficiency declines further to 0.839 and 0.872, respectively. This comparison highlights the optimized model's exceptional performance in maximizing storage space utilization, allowing it to maintain higher storage efficiency. By classifying task-specific data and applying storage strategies tailored to diverse task characteristics, the optimized model fully capitalizes on available resources, meeting the storage efficiency demands of various tasks. Its stability and superior efficiency render it particularly well-suited for large-scale data environments.

IV. DISCUSSION

In cloud computing environments, the characteristics of different tasks can significantly impact data storage efficiency and energy consumption performance. These tasks often vary in terms of data volume, computational complexity, I/O requirements, and latency demands. To address these variations, this study proposes an energy-efficient data storage algorithm that intelligently identifies and classifies tasks, dynamically adjusting data storage strategies to maximize storage efficiency and reduce energy consumption. Experimental results show that the proposed algorithm maintains high accuracy, precision, recall, and F1 scores when processing large-scale data, with low latency and stable throughput performance. Notably, the algorithm exhibits minimal throughput decline under varying data volumes, while energy consumption shows a steady growth, demonstrating its strong adaptability—especially suited for cloud computing scenarios with diverse task characteristics and fluctuating data volumes. Additionally, the algorithm's optimized approach to storage strategy adjustment significantly reduces energy consumption during large-scale data processing. For instance, with data volumes ranging from 1,000 to 3,000, the algorithm consistently consumes less energy than the comparative models. The energy-saving effect is particularly evident, as large data processing tasks typically involve extensive data read/write operations and substantial computational resource consumption. By intelligently identifying task characteristics and optimizing data storage strategies, the proposed algorithm reduces energy consumption and enhances processing efficiency, making it especially applicable to platforms with growing data analysis demands. Cloud storage systems often need to handle diverse storage requirements, where fluctuations in data volume make storage efficiency and energy consumption critical issues. The proposed algorithm significantly reduces energy consumption while ensuring storage efficiency, making it suitable for long-term, continuous cloud storage services, such as CDN or enterprise cloud storage systems. Compared to the research by Shi et al., this study focuses more on optimizing energy savings when handling large-scale heterogeneous tasks. Their model, based on static storage strategies, demonstrates some energy-saving effects in small-scale tasks, but as data volume increases, their model's energy consumption rises significantly. In contrast, the proposed algorithm dynamically adjusts data storage strategies, further improving energy efficiency in large-scale task processing, particularly when data volumes fluctuate sharply, with much lower energy consumption growth than their model. Therefore, this study demonstrates stronger adaptability in big data environments, addressing the shortcomings of their research in large-scale task processing [23]. In comparison with Zhang et al.'s research, this study not only emphasizes storage efficiency optimization but also introduces a more detailed mechanism for identifying heterogeneous tasks. Zhang et al. primarily focused on improving storage efficiency by simplifying the task identification process to reduce computational overhead. However, simplified task identification strategies may lead to fluctuations in storage efficiency when dealing with complex and dynamic tasks. The proposed algorithm, by intelligently identifying heterogeneous tasks and dynamically adjusting storage strategies, strikes a balance between task processing accuracy and storage efficiency. As a result, this study not only compensates for the deficiencies in their research related to task identification and data storage but also offers a more generalized solution [24].

Through experimental analysis and application scenarios, the proposed energy-efficient storage algorithm for heterogeneous cloud computing tasks demonstrates significant advantages across multiple dimensions, particularly in meeting the storage and processing needs of large-scale heterogeneous tasks. Future research directions could focus on further enhancing the scalability and dynamic adaptability of the algorithm to better handle increasingly complex cloud computing task environments.

V. CONCLUSION

This study proposes and implements an energy-efficient data storage algorithm that intelligently identifies and classifies data characteristics for differentiated cloud computing tasks. Based on this, the dynamic adjustment of data storage strategies has optimized storage performance, reduced energy consumption, and enhanced overall system efficiency. Compared to existing models, the proposed optimized model demonstrates distinct advantages, not only in storage efficiency but also in throughput, latency, and energy consumption. Despite the significant advantages in performance and energy savings, this study has several limitations. First, the model shows limitations in adapting to certain anomalous data features specific to particular task types, indicating that the current task classification approach requires further refinement to ensure high accuracy and efficiency across a broader range of data characteristics. Second, the algorithm's performance optimization under high concurrency conditions requires further testing and improvement. It remains to be verified whether the algorithm can maintain its superior performance in more complex task environments, such as high-concurrency task processing scenarios. Future research will focus on several key areas for improvement. First, a multi-task concurrency mechanism will be introduced to enhance the efficiency of data allocation and resource utilization across different task types. By improving parallel processing capabilities, the optimized model will be able to maintain stability in high-concurrency environments. Second, machine learning methods will be incorporated to develop more refined adaptive storage strategies that address dynamically changing data requirements, thereby enhancing the model's adaptability in complex cloud computing tasks. Additionally, the study will explore the synergies between edge computing and cloud computing, investigating more efficient edge-cloud data storage and task allocation strategies to achieve superior performance and energy efficiency. With these improvements, the optimized model will be better suited to large-scale, dynamic cloud computing environments, enhancing its efficiency and energy-saving capacity in multi-task processing.

REFERENCES

[1] K. Y. Tai, F. Y. S. Lin, and C. H. Hsiao, "An integrated optimization-based algorithm for energy efficiency and resource allocation in heterogeneous cloud computing centers," IEEE Access, vol. 56, no. 13, pp. 556-557, Mar. 2023.

[2] Y. Dong, H. Sui, and L. Zhu, "Application of cloud computing combined with GIS virtual reality in construction process of building steel structure," Math. Probl. Eng., vol. 2, no. 2, pp. 11-13, Feb. 2022.

[3] A. Lakhan, M. A. Mohammed, A. N. Rashid, S. Kadry, & K. H. Abdulkareem, "Deadline aware and energy-efficient scheduling algorithm for fine-grained tasks in mobile edge computing," Int. J. Web Grid Serv., vol. 18, no. 2, pp. 168-193, Feb. 2022.

[4] F. S. Prity, M. H. Gazi, and K. M. A. Uddin, "A review of task scheduling in cloud computing based on nature-inspired optimization algorithm," Cluster Comput., vol. 26, no. 5, pp. 3037-3067, May 2023.

[5] H. Yang, H. Zhou, Z. Liu, X. Deng, "Energy optimization of wireless sensor embedded cloud computing data monitoring system in 6G environment," Sensors, vol. 23, no. 2, pp. 1013, Feb. 2023.

[6] G. Saravanan, S. Neelakandan, P. Ezhumalai, S. Maurya, "Improved wild horse optimization with levy flight algorithm for effective task scheduling in cloud computing," J. Cloud Comput., vol. 12, no. 1, pp. 24, Jan. 2023.

[7] S. T. Manukumar and V. Muthuswamy, "A novel data size-aware offloading technique for resource provisioning in mobile cloud computing," Int. J. Commun. Syst., vol. 36, no. 2, pp. 5378, Feb. 2023.

[8] L. Hamid, A. Jadoon, and H. Asghar, "Comparative analysis of task level heuristic scheduling algorithms in cloud computing," J. Supercomput., vol. 78, no. 11, pp. 12931-12949, Nov. 2022.

[9] W. Zhang, R. Yadav, Y. C. Tian, S. K. S. Tyagi, I. A. Elgendy, & S. Kaiwartya, "Two-phase industrial manufacturing service management for energy efficiency of data centers," IEEE Trans. Ind. Informat., vol. 18, no. 11, pp. 7525-7536, Nov. 2022.

[10] A. Rahimikhanghah, M. Tajkey, B. Rezazadeh, A. M. Rahmani, "Resource scheduling methods in cloud and fog computing environments: a systematic literature review," Cluster Comput., vol. 118, no. 42, pp. 1-35, Oct. 2022.

[11] N. El-Menbawy, H. A. Ali, M. S. Saraya, "Energy-efficient computation offloading using hybrid GA with PSO in internet of robotic things environment," J. Supercomput., vol. 79, no. 17, pp. 20076-20115, Sept. 2023.

[12] Y. Dong, H. Sui, and L. Zhu, "Application of cloud computing combined with GIS virtual reality in construction process of building steel structure," Math. Probl. Eng., vol. 5, no. 3, pp. 9812-9814, Mar. 2022.

[13] E. Al-Masri, A. Souri, H. Mohamed, W. Yang, J. Olmsted, & O. Kotevska, "Energy-efficient cooperative resource allocation and task scheduling for Internet of Things environments," Internet Things, vol. 23, no. 3, pp. 100832, Mar. 2023.

[14] K. Rajalakshmi, M. Sambath, L. Joseph, K. Ramesh, R. Surendiran, "An effective approach for improving data access time using intelligent node selection model (INSM) in cloud computing environment," SSRG Int. J. Electr. Electron. Eng., vol. 10, no. 5, pp. 174-184, May 2023.

[15] K. Li, J. Zhao, J. Hu, Y. Chen, "Dynamic energy efficient task offloading and resource allocation for NOMA-enabled IoT in smart buildings and environment," Build. Environ, vol. 22, no. 6, pp. 109513, June 2022.

[16] Y. Wang, W. Shafik, J. T. Seong, M. S. A. Mustafa, M. R. Mouhamed, "Service delay and optimization of the energy efficiency of a system in fog-enabled smart cities," Alexandria Eng. J., vol. 8, no. 4, pp. 112-125, Apr. 2023.

[17] S. Alhelaly, A. Muthanna, and I. A. Elgendy, "Optimizing task offloading energy in multi-user multi-UAV-enabled mobile edge-cloud computing systems," Appl. Sci., vol. 12, no. 13, pp. 6566, July 2022.

[18] H. J. Muhasin, M. A. Jabar, S. Abdullah, S. Kasim, "Managing sensitive data in cloud computing for effective information system' decisions," Acta Informatica Malaysia. vol. 1, no. 2, pp. 01-02. February. 2017.

[19] R. Agrawal, S. Singhal, and A. Sharma, "Blockchain and fog computing model for secure data access control mechanisms for distributed data storage and authentication using hybrid encryption algorithm," Cluster Comput., vol. 1, no. 1, pp. 1-16, Jan. 2024.

[20] R. Thatikonda, A. Padthe, S. A. Vaddadi, P. R. R. Arnepal, "Effective secure data agreement approach-based cloud storage for a healthcare organization," Int. J. Smart Sens. Adhoc Netw., vol. 3, no. 4, pp. 19-20, Apr. 2023.

[21] S. Zhao, "Energy efficient resource allocation method for 5G access network based on reinforcement learning algorithm," Sustain. Energy Technol. Assess., vol. 56, no. 20, pp. 103020, Mar. 2023.

[22] S. Zhang, Z. Wang, Z. Zhou, "Blockchain and federated deep reinforcement learning-based secure cloud-edge-end collaboration in power IoT," IEEE Wireless Commun., vol. 29, no. 2, pp. 84-91, Feb. 2022.

[23] W. Shi, H. Li, J. Guan, M. S. A. Mustafa, M. R. Mouhamed, "Energy-efficient scheduling algorithms based on task clustering in heterogeneous Spark clusters," Parallel Comput., vol. 11, no. 2, pp. 102947, Feb. 2022.

[24] P. Zhang, N. Chen, G. Xu, M. Guizani, Y. Duan, K. Yu, "Multi-target-aware dynamic resource scheduling for cloud-fog-edge multi-tier computing network," IEEE Trans. Intell. Transp. Syst., vol. 5, no. 3, pp. 20, Mar. 2023.