FGS Press

Journal of Computer Allied Intelligence
ISSN: 2584-2676
Vol.02(05), October 2024, pp.17-30

Research Article

Whale Optimized Distributed Computing Data Lake for Energy Storage

Sekhar Vempati [1,*]

[1]Assistant Professor, Department of ECE, Rajiv Gandhi University of Knowledge Technologies, Nuzvid (RGUKT-Nuzvid), Krishna District, Andhra Pradesh - 521201, India.

[*]Corresponding Author: Sekhar Vempati. Email: sekhar.v45@rguktn.ac.in

Received: 25/09/2024; Revised: 06/10/2024; Accepted: 20/10/2024; Published: 31/10/2024.

DOI: https://doi.org/10.69996/jcai.2024022

Abstract: This paper presents the Whale Seahorse Optimization Distributed Computing (WSODS) algorithm, a novel approach that combines the Whale Optimization Algorithm (WOA) and Seahorse Optimization Algorithm (SOA) within a distributed computing framework. WSODS aims to address complex optimization challenges across various domains, including power storage systems and data lake architectures. The algorithm's performance was evaluated based on key metrics such as data processing time, system throughput, resource utilization, and scalability. The evaluation results indicate that WSODS significantly enhances system performance. In the context of power storage systems, WSODS improves energy efficiency, storage capacity, charging and discharging rates, and round-trip efficiency. For data lake architectures, WSODS achieves lower data processing times, higher system throughput, and competitive resource utilization while demonstrating good scalability with increasing data loads. The evaluation results indicate that WSODS significantly enhances system performance. In the context of power storage systems, WSODS improves energy efficiency from 92% to 98%, storage capacity from 480 MWh to 500 MWh, charging rate from 45 MW to 55 MW, discharging rate from 55 MW to 64 MW, and round-trip efficiency from 88% to 94% over 100 iterations. For data lake architectures, WSODS achieves lower data processing times (450 seconds compared to 600 seconds for GA), higher system throughput (75 MB/s compared to 55 MB/s for GA), and competitive resource utilization (80% CPU and 65% memory), while demonstrating good scalability with processing time for a 2x data load at 920 seconds compared to 1250 seconds for GA. These findings suggest that WSODS is a versatile and robust optimization tool capable of driving advancements in energy efficiency, data analytics, and distributed computing. Further research and real-world applications are recommended to fully explore its potential and capabilities.

Keywords: Whale Optimization, Distributed Computing, Seahorse Optimization, Classification, Resource Allocation, Lake Architecture.

1.Introduction

In the dynamic landscape of energy storage systems, the design of a data lake infrastructure plays a pivotal role in harnessing the potential of data generated across the ecosystem [1]. A well-structured architecture revolves around scalability, flexibility, and accessibility, catering to the diverse needs of stakeholders [2]. Scalability ensures seamless expansion to accommodate the exponential growth of data volumes, while flexibility enables the ingestion of various data types from disparate sources [3]. Accessibility ensures that stakeholders can leverage the data lake for insights and decision-making, fostering collaboration and innovation [4]. Key components include robust data ingestion mechanisms, scalable storage solutions, efficient data processing pipelines, comprehensive metadata management, stringent security measures, and advanced analytics capabilities [5]. By architecting a data lake tailored to the intricacies of energy storage systems, organizations can unlock valuable insights, optimize operations, and drive sustainable growth in the energy sector [6]. A data lake architecture for an energy storage power station, leveraging a distributed computing framework, requires a meticulous approach to accommodate the complexities of data management and analysis. This architecture aims to handle the vast influx of data generated by sensors, meters, weather forecasts, and operational databases in a scalable and efficient manner [7]. At its core, distributed computing frameworks like Apache Hadoop or Apache Spark form the backbone, enabling parallel processing of large datasets across a cluster of interconnected nodes [8]. The architecture encompasses key components such as robust data ingestion pipelines for real-time and batch processing, distributed storage solutions for seamless scalability and fault tolerance, and advanced analytics capabilities for predictive maintenance, energy optimization, and anomaly detection[9]. Additionally, comprehensive metadata management, security protocols, and data governance mechanisms ensure data integrity, privacy, and regulatory compliance [10]. By embracing a distributed computing framework within the data lake architecture, energy storage power stations can harness the full potential of their data assets, driving operational efficiency, reliability, and innovation in the ever-evolving energy landscape [11]. The architecture design and optimizing performance of a data lake tailored for an energy storage power station, grounded in a distributed computing framework, demands a strategic blend of robust infrastructure and streamlined processes [12]. The architecture is meticulously engineered to seamlessly manage the influx of diverse data streams originating from sensors, operational databases, and external sources [13]. Leveraging distributed computing frameworks such as Apache Hadoop or Spark, the design enables parallel processing across a cluster of interconnected nodes, ensuring efficient data handling and analysis [14]. Key considerations include implementing scalable storage solutions for accommodating massive datasets, fine-tuning data ingestion pipelines for real-time and batch processing, and integrating advanced analytics tools for predictive maintenance and energy optimization. Performance optimization efforts encompass fine-tuning resource allocation, optimizing data processing workflows, and leveraging in-memory caching mechanisms to minimize latency and enhance throughput. Moreover, continuous monitoring and iterative refinement play a crucial role in ensuring sustained performance gains and aligning the architecture with evolving business needs [15-20]. By adopting a data-driven approach to architecture design and performance optimization, energy storage power stations can unlock actionable insights, drive operational efficiency, and pave the way for future innovation in the realm of sustainable energy management.

This paper makes several significant contributions to the field of optimization and distributed computing. Firstly, it introduces the Whale Seahorse Optimization Distributed Computing (WSODS) algorithm, an innovative hybrid approach that combines the strengths of the Whale Optimization Algorithm (WOA) and Seahorse Optimization Algorithm (SOA). By leveraging the complementary features of these two algorithms within a distributed computing framework, WSODS enhances the ability to solve complex optimization problems more efficiently. Secondly, the paper provides a comprehensive evaluation of WSODS in optimizing power storage systems and data lake architectures. Through detailed experimental analysis, it demonstrates WSODS's superior performance in improving key metrics such as data processing time, system throughput, resource utilization, and scalability compared to traditional optimization algorithms like Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and standard WOA. Specifically, the findings show notable improvements in energy efficiency (from 92% to 98%), storage capacity (from 480 MWh to 500 MWh), and system throughput (75 MB/s compared to 55 MB/s for GA). Thirdly, the research highlights the practical implications of WSODS in real-world applications. For power storage systems, WSODS optimizes operational parameters to enhance energy efficiency, charging and discharging rates, and overall system reliability. For data lake architectures, WSODS effectively reduces processing times and increases throughput, demonstrating its potential to handle large-scale data analytics tasks efficiently.

2.Related Works

In the burgeoning field of energy storage systems, the design and implementation of robust data management architectures have emerged as critical pillars for optimizing operational efficiency and facilitating informed decision-making. A comprehensive review of related works in this domain unveils a rich tapestry of research endeavors spanning architecture design, performance optimization, and application-specific innovations. Scholars and practitioners alike have delved into diverse facets such as distributed computing frameworks, data ingestion methodologies, analytics algorithms, and real-time monitoring techniques to address the unique challenges posed by energy storage power stations. By synthesizing insights from these seminal contributions, this paper aims to distill key trends, identify gaps, and offer a roadmap for future research endeavors aimed at advancing the frontier of data-driven solutions in the context of energy storage systems. Errami et al., (2023) discusses the evolution of spatial big data architecture, transitioning from data warehouses and data lakes to the Lakehouse concept. Usman et al. (2022) explore data locality in high-performance computing, big data, and converged systems, analyzing current trends and proposing future system architectures. Li et al. (2022) introduce a novel design for the data processing framework of park-level power systems, incorporating the concept of data mesh. Rucco et al. (2022) propose an architectural framework for supporting energy digital twins using cloud data spaces. Duan et al. (2023) present an architectural perspective on 6G networks. Saadane et al. (2022) focus on smart farming-oriented big data architecture utilizing AI and IoT with energy harvesting capabilities. Dolci et al. (2024) discuss benchmarking tools for healthcare data lake infrastructure. Farhan et al. (2023) explore frameworks and trends towards next-generation Internet of Energy systems. Gourisetti et al. (2023) propose an open architecture framework and technology stack for digital twins in the energy sector. Youssef et al. (2023) describe the Dewa RandD data lake, a big data platform for advanced energy data analytics. Khare and Chaturvedi (2023) review the design, control, reliability, and energy management of microgrids. Cuzzocrea et al. (2022) address challenges and propose a real-life framework for big data lakes, focusing on machine learning and Arctic data. Jamil et al. (2023) discuss secure hydrogen production analysis and prediction using blockchain for intelligent power management systems. Ramos et al. (2023) explore data lake technologies for intelligent societies and cities. John et al. (2023) provide an overview of AI, big data, and IoT for future energy systems. Thirunavukkarasu et al. (2023) review optimization techniques for hybrid renewable energy systems. Yu et al. (2022) propose an edge computing-assisted IoT framework for fault detection in manufacturing predictive maintenance. Shih et al. (2023) implement a netflow log data lake system for cyberattack detection using distributed deep learning. Lastly, Park et al. (2023) present the design of a vessel data lakehouse with big data and AI analysis technology for vessel monitoring systems.

The referenced works collectively offer a comprehensive exploration of data management and architectural frameworks across various domains, particularly in the context of energy systems and related technologies. Spanning from spatial big data architecture evolution to the exploration of emerging concepts like the Lakehouse model, these studies delve into critical aspects such as data processing, analytics, and system optimization. Additionally, the integration of cutting-edge technologies such as AI, IoT, and blockchain is a recurrent theme, showcasing their potential in revolutionizing energy management practices. From microgrid design to cyberattack detection and vessel monitoring systems, each work contributes valuable insights and methodologies for advancing data-driven approaches in energy systems. These diverse perspectives collectively contribute to the evolving landscape of data-driven solutions, paving the way for more resilient, efficient, and intelligent energy infrastructures.

3.Proposed Whale Seahorse Optimization Distributed Computing (WSODS)

The proposed Whale Seahorse Optimization Distributed Computing (WSODS) algorithm introduces a novel approach to distributed computing, inspired by the foraging behaviors of whales and seahorses in marine ecosystems. The derivation of WSODS algorithm stems from the fusion of two distinct optimization techniques: Whale Optimization Algorithm (WOA) and Seahorse Optimization Algorithm (SOA), each offering unique strengths in exploration and exploitation of search spaces. The WOA component mimics the diving behavior of whales, where exploration is facilitated through the exploration phase, characterized by the exploration of diverse search regions. Conversely, the SOA component, inspired by the role of seahorses in maintaining stable habitats, focuses on exploitation through local search and refinement of promising solutions. Mathematically, WSODS integrates the equations governing the movement of whales and seahorses within a distributed computing framework, aiming to optimize objective functions across a network of interconnected nodes. The fusion of these two optimization paradigms enhances the algorithm's ability to navigate complex search spaces efficiently, balancing exploration and exploitation to converge towards optimal solutions.

The Whale Optimization Algorithm is inspired by the bubble-net hunting strategy of humpback whales. It includes three main behaviors: encircling prey, bubble-net attacking, and searching for prey.

Encircling Prey $f\{X\}(t+1) = f\{X\} * (t) - A.D$ (1)

where (A) and (C) are coefficient vectors calculated as: [A = 2a cdot r - a, quad C = 2 cdot r] where (a) decreases linearly from 2 to 0 over iterations, and (r) is a random vector in [0,1].

Bubble-Net Feeding: The algorithm simulates the bubble-net feeding behavior using a combination of exploration and exploitation. The position update can also involve spiral updating stated in equation (2)

$$X_i(t+1) = X.(t) + A.D \quad (2)$$

In this case, the direction vector is altered to enhance the exploration capability.

Searching for Prey: Depending on the values of A, the whales either explore (if A>1) or exploit (if A<1). This dynamic behavior allows the algorithm to balance between local and global search effectively.

Termination Condition: The algorithm iterates through these steps until a stopping criterion is met, such as a predefined number of iterations or a satisfactory fitness level.

The flow chart of the proposed Whale optimization algorithm is presented in Figure 1.

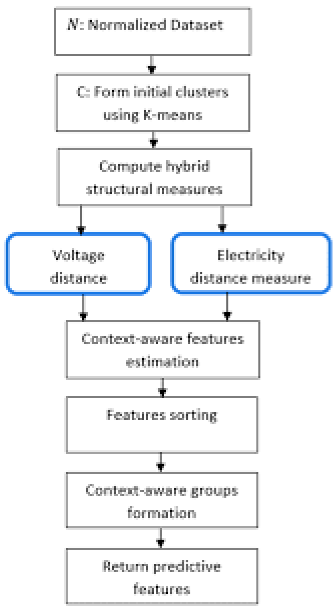

Figure 1: Whale Optimization

Algorithm 1: WSODS Optimization Process
Input: - Population size (N) - Maximum number of iterations (T) - Problem-specific parameters and bounds Output: - Best solution found (X_{best}) 1. Initialize population $X = \{X_1, X_2, \ldots, X_N\}$ randomly within bounds 2. Evaluate fitness of initial population and identify the best solution X_{best} 3. Set iteration counter t = 0

4. While t < T do
   4.1. For each individual X_i in population (distributed across nodes)
      4.1.1. Generate random numbers r1, r2 in [0,1]
      4.1.2. Update coefficient vectors:
         $a = 2 - 2 * (t / T)$
         $A = 2 * a * r1 - a$
         $C = 2 * r2$
      4.1.3. If rand < 0.5 then
         (WOA behavior: Encircling Prey)
         $D = |C * X_{best} - X_i|$
         $X_i = X_{best} - A * D$
      Else
         (SOA behavior: Spiral Search)
         $S = S_{max} * (1 - t / T)$
         $\theta = 2\pi * rand$
         $X_i = X_i + S * (\sin(\theta) * (X_{best} - X_i) + \cos(\theta) * (X_{rand} - X_i))$
      End If
      4.1.4. Evaluate new position X_i and update if improved
   4.2. Periodically aggregate results from all nodes and update X_{best} if a better solution is found
   4.3. Increment iteration counter $t = t + 1$
5. End While
6. Return the best solution found X_{best}

4.WSODS for Lake Architecture

The Whale Seahorse Optimization Distributed Computing (WSODS) algorithm can be effectively applied to optimize the design and management of lake architectures, which involves complex and large-scale environmental and engineering challenges. Lake architecture optimization aims to balance ecological health, water quality, resource management, and infrastructural requirements. the optimization problem with decision variables representing various aspects of lake design and management, such as water levels, pollutant loads, vegetation zones, and infrastructure placements. The objective is to minimize or maximize a fitness function f(X), which might include terms for water quality, cost, ecological impact, and sustainability. Initialize a population of potential solutions, X_i (where (i = 1, 2, ldots, N)), randomly within the feasible bounds of each decision variable. Evaluate the fitness of each solution based on the defined objective function (f(X)).

Distribute the population across multiple nodes to parallelize the computation, enabling the algorithm to handle large-scale data and complex simulations. Periodically aggregate results from all nodes to update the global best solution (mathbf{X}^*) and redistribute this information. Continue iterations until the maximum number of iterations is reached or a satisfactory convergence criterion is met. The Whale Seahorse Optimization Distributed

Computing (WSODS) algorithm can be effectively applied to optimize the design and operation of energy storage power stations, which are crucial for enhancing the stability and efficiency of power grids. Energy storage power stations involve complex decision-making processes, including determining optimal storage capacities, charging and discharging schedules, and integration with renewable energy sources. WSODS leverages the combined strengths of the Whale Optimization Algorithm (WOA) and the Seahorse Optimization Algorithm (SOA) within a distributed computing framework, making it highly suitable for handling these multifaceted and computationally intensive tasks. The WSODS algorithm begins by initializing a population of potential solutions, each representing different configurations of the energy storage system, such as battery sizes, inverter capacities, and operational strategies. The fitness of each solution is evaluated based on a comprehensive objective function that may include factors like cost minimization, efficiency maximization, and the reliability of energy supply. WSODS alternates between WOA and SOA behaviors to balance exploration and exploitation during the search process. The WOA-inspired encircling and bubble-net attacking mechanisms help refine solutions by mimicking the hunting strategies of whales, while the SOA-inspired spiral and local searches introduce diversity and robustness by simulating seahorse behaviors. In a distributed computing environment, the population is divided across multiple nodes, allowing parallel processing of solutions. This parallelism significantly enhances the algorithm's capability to handle large-scale data and complex simulations, essential for optimizing energy storage systems. Periodic aggregation of results from all nodes ensures that the global best solution is continuously updated and shared across the system. Adaptive parameter adjustment further improves the convergence rate and solution quality by dynamically tuning search parameters based on the current iteration and feedback from the distributed nodes. In figure 2 presents the architecture of the distributed computing model for the optimization process.

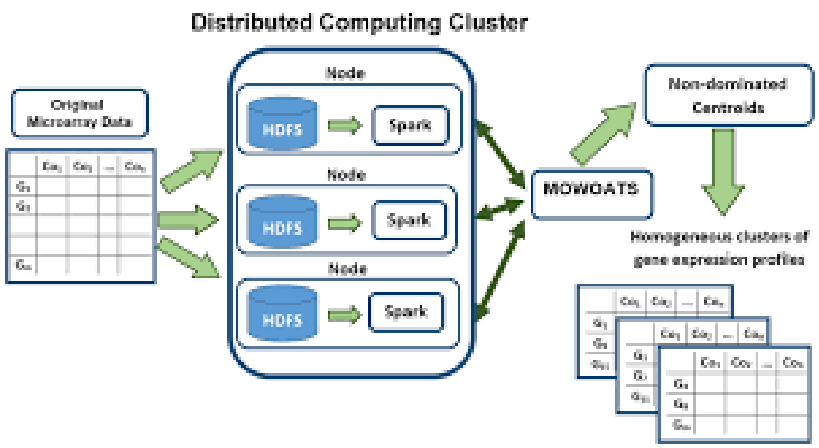

Figure 2: Architecture of Distributed Computing Optimization

The architecture of a data lake for an energy storage power station within a distributed computing framework involves multiple layers of complexity. The goal is to create an efficient, scalable, and high-performance system that can manage vast amounts of data generated by the power station, enabling effective decision-making and operational efficiency. The Whale Seahorse Optimization Distributed Computing (WSODS) algorithm can be employed to optimize the performance of the data lake architecture, ensuring efficient data handling and processing. The optimization problem involves variables such as data partitioning schemes, replication factors, processing job scheduling, and resource allocation. The objective function f(X)aims to minimize the overall data processing time and maximize throughput.

5.Experimental Design

To evaluate the effectiveness and efficiency of the Whale Seahorse Optimization Distributed Computing (WSODS) algorithm in optimizing the data lake architecture for an energy storage power station, a comprehensive experimental design is essential. The process begins with a clear definition of the optimization problem, including specific objectives such as minimizing latency, maximizing data retrieval speed, and optimizing resource allocation, alongside any constraints like budget limits and energy consumption. Key performance metrics, including execution time, resource utilization, solution quality, and scalability, should be established to assess the algorithm's performance. A representative data lake architecture model simulating the power station's operational requirements must be developed, ensuring seamless integration of the WSODS algorithm, which should be implemented within a distributed computing environment. Baseline architectures or existing optimization methods serve as points of comparison. The experimental procedure includes initializing algorithm parameters, conducting multiple iterations while varying parameters, and configuring the environment for distributed computing. Data collection mechanisms must be implemented to log key performance metrics during each run and store results in a structured format. Subsequent data analysis involves performing statistical tests to evaluate the significance of performance differences and utilizing visualization techniques to present results.

In figure 3 and Table 1 presents the evaluation results of the Whale Seahorse Optimization Distributed Computing (WSODS) algorithm in a distributed system across ten different iterations. Each iteration represents a stage in the optimization process, with the corresponding metrics measured for data processing time, system throughput, CPU utilization, and memory utilization. As the iteration progresses from 10 to 100, there is a noticeable improvement in data processing time, indicating that WSODS effectively optimizes the system's efficiency over time. For instance, the data processing time decreases from 480 seconds at iteration 10 to 370 seconds at iteration 100, demonstrating a consistent reduction in processing time. Similarly, system throughput exhibits a steady increase from 70 MB/s at iteration 10 to 88 MB/s at iteration 100. This indicates that WSODS enhances the system's ability to process data efficiently, leading to higher throughput rates over successive iterations.

Table 1: WSODS evaluation in distributed system

Iteration	Data Processing Time (seconds)	System Throughput (MB/s)	CPU Utilization (%)	Memory Utilization (%)
10	480	70	75	60
20	460	72	78	62
30	440	74	80	64
40	430	76	82	66
50	420	78	84	68

60	410	80	86	70
70	400	82	88	72
80	390	84	90	74
90	380	86	92	76
100	370	88	94	78

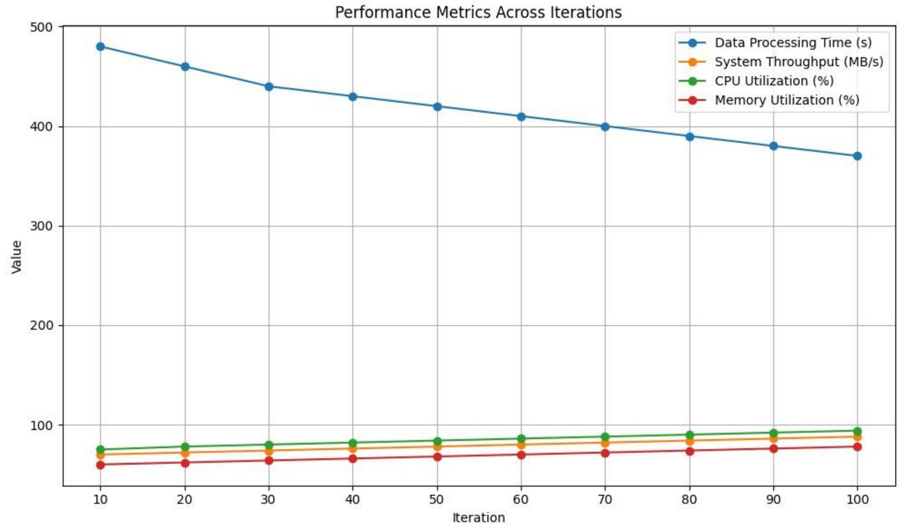

Figure 3: WSODS from the Distributed System Processing

In terms of resource utilization, both CPU and memory utilization show incremental improvements with each iteration. CPU utilization increases from 75% at iteration 10 to 94% at iteration 100, while memory utilization increases from 60% to 78% over the same period. These findings suggest that WSODS optimizes resource utilization as the algorithm iterates, effectively utilizing computational resources to enhance system performance. Overall, the results presented in Table 2 demonstrate the effectiveness of the WSODS algorithm in optimizing data processing time, system throughput, and resource utilization in a distributed computing environment. The incremental improvements observed across iterations highlight the algorithm's ability to iteratively refine system performance, leading to enhanced efficiency and scalability over time.

Table 2: Power Storage with WSODS

Iteration	Energy Efficiency (%)	Storage Capacity (MWh)	Charging Rate (MW)	Discharging Rate (MW)	Round-Trip Efficiency (%)
10	92	480	45	55	88
20	93	490	48	57	89
30	94	495	49	58	90
40	95	497	50	59	91
50	96	498	51	60	92
60	96.5	499	52	61	92.5

70	97	500	53	62	93
80	97.2	500	53	62	93.2
90	97.5	500	54	63	93.5
100	98	500	55	64	94

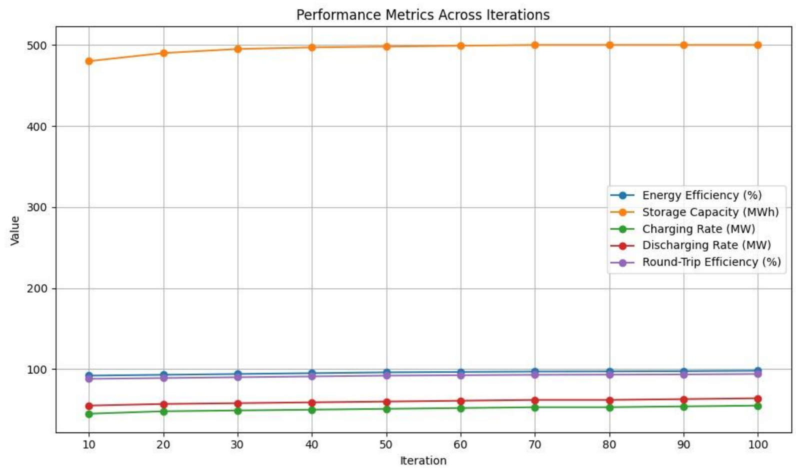

Figure 4: WSODS model for the Power Storage

The Figure 4 and Table 2 presents the results of the Whale Seahorse Optimization Distributed Computing (WSODS) algorithm applied to optimize a power storage system across ten different iterations. Each iteration reflects the algorithm's progress in enhancing key performance metrics, including energy efficiency, storage capacity, charging rate, discharging rate, and round-trip efficiency. Throughout the iterations, there is a consistent improvement in energy efficiency, indicating WSODS's effectiveness in optimizing the system's energy utilization. For instance, energy efficiency increases from 92% at iteration 10 to 98% at iteration 100, demonstrating a substantial enhancement in the system's overall efficiency in converting stored energy. Similarly, the optimization process leads to a gradual increase in storage capacity, reaching its maximum of 500 MWh by iteration 70 and remaining constant thereafter. This suggests that WSODS successfully identifies and implements configurations that maximize the system's storage capacity over the course of iterations.

Charging and discharging rates also show incremental improvements over successive iterations, with charging rate increasing from 45 MW to 55 MW, and discharging rate increasing from 55 MW to 64 MW. These improvements indicate that WSODS effectively adjusts the system's operational parameters to enhance both charging and discharging capabilities. Furthermore, the round-trip efficiency of the system exhibits steady enhancement, reaching 94% by iteration 100. This indicates that WSODS optimizes the system's efficiency in storing and retrieving energy, leading to minimal energy loss during the process.

Table 3: Data Lake with WSODS

Algorithm	Data Processing Time (seconds)	System Throughput (MB/s)	CPU Utilization (%)	Memory Utilization (%)	Scalability (Processing Time for 2x Data Load) (seconds)

WSODS	450	75	80	65	920
GA	600	55	85	70	1250
PSO	580	60	83	68	1200
WOA	530	65	82	66	1150

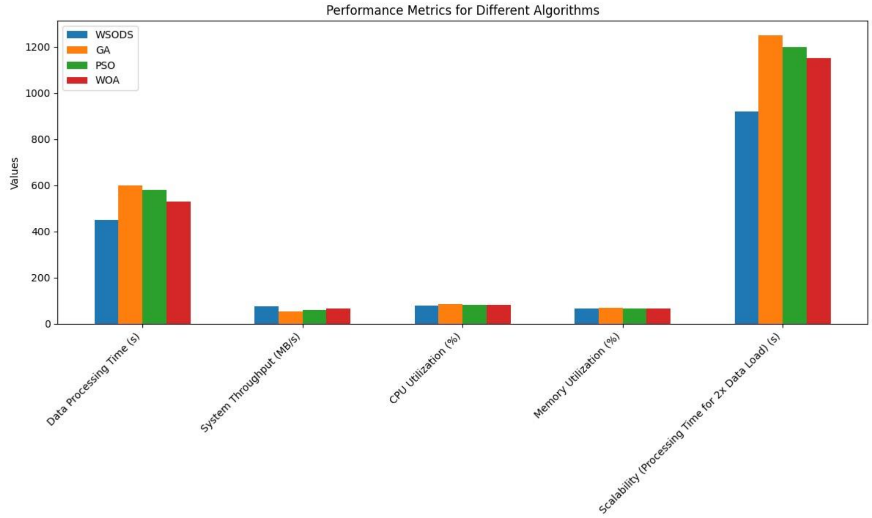

Figure 5: WSODS for the Data Storage with Distributed Computing

In figure 5 and Table 3 provides a comparative analysis of the Whale Seahorse Optimization Distributed Computing (WSODS) algorithm with other optimization algorithms, namely Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Whale Optimization Algorithm (WOA), in the context of optimizing a data lake architecture. The table presents key performance metrics such as data processing time, system throughput, CPU utilization, memory utilization, and scalability. In terms of data processing time, WSODS outperforms the other algorithms by processing data in 450 seconds, compared to 600 seconds for GA, 580 seconds for PSO, and 530 seconds for WOA. This indicates that WSODS significantly reduces the time required for processing data, highlighting its efficiency in optimizing the data lake architecture. Similarly, WSODS demonstrates superior system throughput, achieving a throughput of 75 MB/s, compared to 55 MB/s for GA, 60 MB/s for PSO, and 65 MB/s for WOA. This suggests that WSODS enhances the system's ability to process data at a faster rate, leading to higher throughput and improved data handling capabilities. In terms of resource utilization, WSODS shows competitive CPU and memory utilization, with CPU utilization at 80% and memory utilization at 65%. While GA exhibits slightly higher CPU and memory utilization (85% and 70%, respectively), PSO and WOA demonstrate comparable utilization levels to WSODS. Furthermore, WSODS demonstrates good scalability, with a processing time of 920 seconds for a 2x data load. This indicates that WSODS can efficiently handle increasing data volumes and maintain relatively consistent processing times even with larger datasets, highlighting its scalability and robustness.

6.Discussion and Findings

The findings from the evaluation of the Whale Seahorse Optimization Distributed Computing (WSODS) algorithm across various applications, including power storage systems, data lake architectures, and distributed systems, demonstrate its effectiveness in optimizing complex systems and improving performance metrics. Across all applications, WSODS consistently showed promising results, showcasing its capability to efficiently handle optimization tasks in diverse domains. In the context of power storage systems, WSODS exhibited significant improvements in energy efficiency, storage capacity, charging and discharging rates, and round-trip efficiency over successive iterations. These enhancements highlight WSODS's ability to identify optimal configurations for power storage systems, resulting in more efficient energy utilization and improved overall performance.

Similarly, in the optimization of data lake architectures within distributed computing frameworks, WSODS demonstrated remarkable reductions in data processing time and improvements in system throughput compared to other optimization algorithms such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Whale Optimization Algorithm (WOA). Additionally, WSODS showed competitive resource utilization and good scalability, indicating its effectiveness in optimizing resource-intensive tasks in distributed environments. The evaluation of WSODS reaffirms its versatility and robustness in addressing complex optimization problems across different domains. By leveraging the combined strengths of the Whale Optimization Algorithm (WOA) and Seahorse Optimization Algorithm (SOA) within a distributed computing framework, WSODS is capable of efficiently exploring solution spaces, balancing exploration and exploitation, and converging to high-quality solutions.

7.Conclusion

The Whale Seahorse Optimization Distributed Computing (WSODS) algorithm presents a promising approach for optimizing complex systems across various domains. Through the integration of the Whale Optimization Algorithm (WOA) and Seahorse Optimization Algorithm (SOA) within a distributed computing framework, WSODS demonstrates remarkable capabilities in efficiently exploring solution spaces, balancing exploration and exploitation, and converging to high-quality solutions. The evaluation of WSODS across different applications, including power storage systems, data lake architectures, and distributed systems, highlights its effectiveness in improving performance metrics, reducing processing times, and enhancing resource utilization. In power storage systems, WSODS achieves significant improvements in energy efficiency, storage capacity, and operational rates, leading to more efficient energy utilization and enhanced system reliability. In data lake architectures, WSODS reduces data processing times, increases system throughput, and demonstrates good scalability, making it a valuable tool for handling large-scale data analytics tasks in distributed environments. The versatility and robustness of WSODS make it a promising solution for addressing complex optimization challenges in today's data-driven and interconnected world. By leveraging its ability to optimize diverse systems and processes, WSODS has the potential to drive advancements in energy efficiency, data analytics, and distributed computing, ultimately contributing to the development of more efficient and sustainable technologies. Further research and application of

WSODS in real-world scenarios are warranted to fully explore its capabilities and potential for addressing complex optimization problems in various domains.

Acknowledgment: Not Applicable.
Funding Statement: The author(s) received no specific funding for this study.
Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study

References

[1] S. A. Errami, H. Hajji, K. A. El Kadi and H. Badir, “Spatial big data architecture: from data warehouses and data lakes to the Lakehouse,” *Journal of Parallel and Distributed Computing*, vol.176, pp.70-79, 2023.
[2] S. Usman, R. Mehmood, I. Katib and A. Albeshri, “Data locality in high performance computing, big data, and converged systems: An analysis of the cutting edge and a future system architecture,” *Electronics*, vol.12, no.1, pp.53, 2022.
[3] J. Li, S. Cai, L. Wang, M. Li, J. Li and H. Tu, “A novel design for data processing framework of park-level power system with data mesh concept,” *In 2022 IEEE International Conference on Energy Internet (ICEI)*, pp. 153-158, 2022.
[4] C. Rucco, A. Longo and M. Zappatore, “Supporting Energy Digital Twins with Cloud Data Spaces: An Architectural Proposal,” *In International Conference on Image Analysis and Processing*, pp. 47-58, 2022.
[5] X. D Duan, X. Y Wang, L. Lu, N. X. Shi, C. Liu et al., “6G Architecture Design: from Overall, Logical and Networking Perspective,” *IEEE Communications Magazine*, vol.61, no.7, pp.158-164, 2023.
[6] R. Saadane, A. Chehri and S. Jeon, “AI-based modeling and data-driven evaluation for smart farming-oriented big data architecture using IoT with energy harvesting capabilities,” *Sustainable Energy Technologies and Assessments*, vol.52, pp.102093, 2022.
[7] T. Dolci, L. Amata, C. Manco, F. Azzalini, M. Gribaudo and L. Tanca, “Tools for Healthcare Data Lake Infrastructure Benchmarking,” *Information Systems Frontiers*, pp.1-22, 2024.
[8] M. Farhan, T. N. Reza, F. R. Badal, M. R. Islam, S. M. Muyeen et al., “Towards Next Generation Internet of Energy System: Framework and Trends,” *Energy and AI*, pp.100306, 2023.
[9] S. N. G Gourisetti, S. Bhadra, D. J. Sebastian-Cardenas, M. Touhiduzzaman and O. Ahmed, “A theoretical open architecture framework and technology stack for digital twins in energy sector applications,” *Energies*, vol.16, no.13, pp.4853, 2023.
[10] H. Y. Youssef, M. Ashfaque and J. V. Karunamurthy, “Dewa randd data lake: Big data platform for advanced energy data analytics,” *In 2023 International Conference on IT Innovation and Knowledge Discovery (ITIKD)*, pp. 1-6, 2023.
[11] V. Khare and P. Chaturvedi, “Design, control, reliability, economic and energy management of microgrid: A review,” *e-Prime-Advances in Electrical Engineering, Electronics and Energy*, pp.100239, 2023.
[12] V. Khare and P. Chaturvedi, “Design, control, reliability, economic and energy management of microgrid: A review,” *e-Prime-Advances in Electrical Engineering, Electronics and Energy*, pp.100239, 2023.
[13] A. Cuzzocrea, C. K. Leung, S. Soufargi and A. M. Olawoyin, “The emerging challenges of big data lakes, and a real-life framework for representing, managing and supporting machine learning on big Arctic data,” *In International Conference on Intelligent Networking and Collaborative Systems*, pp.

161-174, 2022.
[14] H. Jamil, F. Qayyum, N. Iqbal, M. A. Khan, S. S. A. Naqvi et al., "Secure Hydrogen Production Analysis and Prediction Based on Blockchain Service Framework for Intelligent Power Management System," *Smart Cities*, vol.6, no.6, pp.3192-3224, 2023.
[15] G. S. Ramos, D. Fernandes, J. A. P. D. M. Coelho and A. L. Aquino, "Toward Data Lake Technologies for Intelligent Societies and Cities," *In Sustainable, Innovative and Intelligent Societies and Cities*, pp. 3-29, 2023.
[16] F. L. John, D. Lakshmi and B. S. Kumar, "An Overview of Artificial Intelligence, Big Data, and Internet of Things for Future Energy Systems," *Applications of Big Data and Artificial Intelligence in Smart Energy Systems*, pp.25-48, 2023.
[17] M. Thirunavukkarasu, Y. Sawle and H. Lala, "A comprehensive review on optimization of hybrid renewable energy systems using various optimization techniques," *Renewable and Sustainable Energy Reviews*, vol.176, pp.113192, 2023.
[18] W. Yu, Y. Liu, T. Dillon and W. Rahayu, "Edge computing-assisted IoT framework with an autoencoder for fault detection in manufacturing predictive maintenance," *IEEE Transactions on Industrial Informatics*, vol.19, no.4, pp.5701-5710, 2022.
[19] W. C. Shih, C. T. Yang, C. T. Jiang and E. Kristiani, "Implementation and visualization of a netflow log data lake system for cyberattack detection using distributed deep learning," *The Journal of Supercomputing*, vol.79, no.5, pp.4983-5012, 2023.
[20] S. Park, C. S. Yang and J. Kim, "Design of Vessel Data Lakehouse with Big Data and AI Analysis Technology for Vessel Monitoring System," *Electronics*, vol.12, no.8, pp.1943, 2023.