IJAREEIE

e-ISSN: 2278-8875
p-ISSN: 2320-3765

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

Volume 13, Issue 8, August 2024

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.514

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 8.514|| A Monthly Peer Reviewed & Referred Journal |

||Volume 13, Issue 8, August 2024||

|DOI:10.15662/IJAREEIE.2024.1308012|

Real-time Data Pipeline Optimization using AI-driven Resource Allocation

Manoj Pal

Senior Data Engineer, Artnet Worldwide Corporation Tampa, FL USA

ABSTRACT: This paper discusses an AI approach to improve how data is processed in real time by adjusting the way resources are given to distributed systems. Our framework adapts and changes the amount of CPU, available memory, and network use to hit performance goals due to machine learning and reinforcement learning. The system has shown improvement in latency, throughput, and using resources more efficiently during evaluation with many workloads. To tackle the challenges produced by intense and fast data, we make use of predictive analytics and scheduling that can adjust automatically. Findings point out that using AI for resource efficiency is a flexible and dependable way to run modern data engineering tasks both in clouds and at edges.

KEYWORDS: Pipeline, Data, AI, Resource Allocation, Optimization, Pipeline

I. INTRODUCTION

Since real-time data is multiplying from various sources, data engineering needs to be flexible and smart. Static way of managing resources results in inefficient duration for processing, does not scale as needed, and becomes more expensive in situations with high throughput. Advances in artificial intelligence recently provide new opportunities for varying and awareness-based optimization.

This research looks at how AI strategies such as machine learning and reinforcement learning help in improving how data pipelines operate in real time. In this way, we hope to launch a system that can manage large data demands in the cloud and at the same time use resources efficiently.

II. RELATED WORKS

2.1 Resource Optimization

It is well known that properly using resources is central to ensuring real-time data pipeline efficiency. When putting resources to work, RO entails overseeing aspects like how many partitions are made, tasks are distributed on instances, and how many resources each portion of work uses.

Experts in 2022 noted that since systems such as MaxCompute balance different objectives at the same time, the multi-objective optimization (MOO) method should be applied. By creating a hierarchical architecture, they break optimization into smaller pieces and decide on instances fast with the help of predictive modeling.

They managed to decrease both latency and costs by more than 70% and 78% each. Following these thoughts, Ni et al. (2019) worked on RO issues in stream processing, where the data has to be processed fast and in large volumes as it flows in real time. Because the problem proved to be NP-complete, traditional ways for cutting graphs were considered inadequate.

They came up with a graph-conscious structure that learns how to act through deep reinforcement learning and graph embedding, so it can use these skills for unseen stream processing graphs. According to the results, the new model offered better outcomes than those provided by METIS and LSTM-based models in the majority of tests, pointing to the rising ability of AI in real-time optimization.

The article by Yu et al. (2021) discussed how to apply distributed machine learning by suggesting an algorithm that schedules locality and resource use for training processes. With its local optimization approach, their algorithm worked well and cut training time, proving that dynamic RO needs to consider remote workers and servers together for strong results in big, spread-out pipelines.

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 8.514|| A Monthly Peer Reviewed & Referred Journal |

||Volume 13, Issue 8, August 2024||

|DOI:10.15662/IJAREEIE.2024.1308012|

2.2 Cloud and Edge Environments

It is typical for real-time data pipelines to be flexible enough to deal with changes in usage due to the cloud's adaptability. Ramamoorthi came up with a machine learning strategy that regularly alters CPU, memory, and bandwidth usage according to workload changes in 2021.

Proactive scaling was made possible through predictive modeling and reinforcement learning, which stopped the issue of under- or over-provisioning. Leading cloud platforms were checked and it was found that using adaptive AI-based resource scaling saves money and provides better outcomes.

Moreover, Qureshi et al. (2020) noted that real-time systems for IoT devices and smart systems have to deal with the time- and budget-limitations placed on them. They are reducing the cost of data transfer overhead by picking matching computing and storage resources.

Their way of handling loads worked better than others to ensure that urgent tasks were finished on time. In their research, Kum et al. (2022) designed a new way to use GPUs for deep learning through adjusting batch sizes at the edge level.

Modular tools from Neptune were used to analyze videos inside Kubernetes containers, bringing about improved use of the GPUs. Such options make clear that hybrid cloud-edge orchestration is playing an increasing role in the operation of real-time applications.

In 2023, Ye et al. proposed an architecture called cooperative inference to use on mobile edge computing systems. The researchers minimized the time it took to complete inferences by using divided DNN models on user devices and edge servers and by using a Lyapunov-inspired multi-dimensional optimization algorithm that dynamically manages energy use and data traffic. How their approach works can be seen in the way resources are split and adjusted to suit live AI task demands.

2.3 Intelligent Resource Allocation

Approaches to handling resources in real time prove how complex and wide the problem can be. According to Morariu et al. (2020), a hybrid control system that uses ML and Big Data was developed for making predictions in production and maintenance for cloud manufacturing. Using LSTM neural networks, the system was able to anticipate energy consumption, so that resources could be reallocated as needed and unusual events were spotted.

Data was gathered close to the source and learning and decisions were managed in the cloud, giving their systems a layered structure that helps them handle more data and function despite issues. In their study, Iftikhar et al. (2022) carried out a systematic review concerning fog and edge computing resource management.

They mentioned that reinforcement learning techniques have some drawbacks, such as the complications of online learning, its lack of transparency, and high volatility. The authors drew up a system for organizing AI/ML techniques, including deep learning, online learning, and edge AI control, and revealed the current obstacles, along with suggestions for better research in flexible and mixed environments.

Hassan et al. (2020) also created an integrated system that uses several algorithms (exponential smoothing, MMTMC2 VM selection) and tested it on real testbeds. After using the new systems, the engineers noticed that people moved less often (by 49.44%) and energy use was lower (by 16.64%). A combination of different algorithms makes it easy to optimize resources as needs change.

He mentioned that the use of ML in microservices-based architecture can be applied by following the DevOps approach. Various learning techniques including supervision, no supervision, and reinforcement tricks were used to find resource needs and reassign them right away. Further, it pointed out that proper tools and continuous work hand-in-hand between development and operations are vital for running flexible deployments in current CI/CD settings.

2.4 Comparative Studies

Ability of AI to anticipate resource needs is an important part of optimizing pipelines and their operations. Katragadda et al. looked into forecasting in systems that include cloud and on-premise environments in their study (2022). Supervised and unsupervised algorithms were assessed by comparing their results for spotting anomalies and comparing the requirements for running them.

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 8.514|| A Monthly Peer Reviewed & Referred Journal |

||Volume 13, Issue 8, August 2024||

|DOI:10.15662/IJAREEIE.2024.1308012|

By analyzing real examples, I saw that getting accurate forecasts helps a hybrid system work better and saves money. Besides, the paper pointed out that with explainable AI and federated learning, people could make decisions together and in privacy, without centralized control.

In 2023, Tarafder et al. looked at various AI methods, such as deep reinforcement learning, evolutionary algorithms, and supervised models, to analyze their use in RO in cloud systems. Looking at real data patterns and by conducting simulations, they studied the effectiveness of available techniques for cloud providers.

The research they conduct makes it easier to match the best AI model with the work and needs of any organization. All the studies show that AI models are converging with scheduling and forecasting methods. W
Regardless if it is GPUs for batch inference, resourcing, or LSTM tech for predicting energy usage, AI becomes the main force behind real-time and adaptive RO within pipelines.

Companies manage both their edge resources and cloud resources for real-time data is now revolutionized. All the reviewed literature points to the fact that traditional ways of analyzing data are not effective in fast-moving environments. The latest progress in data engineering relies on reinforcement learning, predictive modeling, and using cloud-edge technology together. Also, the capacity to merge forecasting, anomaly detection, scheduling, and cost optimization within one intelligent system is key to reaching higher levels of performance and efficiency in data processing at any moment.

III. METHODOLOGY

The focus of this research is on forming, carrying out, and examining a data pipeline that uses AI to manage resources. The purpose is to get better processing results, lessen delays, and organize more resources in different dynamic cloud and edge areas.

3.1 System Architecture Design

The system has been designed with three important layers: data ingestion and processing, resource orchestration, and an AI optimization engine. The streaming data is collected by the data ingestion layer from various places (e.g., sensors, web servers, the internet of things devices) and sent to Apache Flink for fast processing and Apache Kafka for message delivery.

Containers are orchestrated by Docker, and this is managed by doing Kubernetes resource coordination. Microservices can be scaled on the fly, data can be processed in parallel, and the load can easily be handled on both cloud and edge nodes.

3.2 Resource Allocation Engine

Dynamic and predictive allocation of essential resources is made possible by an AI-based engine that forms the key part of the methodology. There are three major components inside the engine:

- **Resource Usage**: For this step, the model prepares a LSTM by using ongoing CPU, memory, and bandwidth data from the past. Since LSTM anticipates the near-term (for example, within the next 5–10 seconds) demand for resources, the system can quickly ensure resources are ready when needed.
- **Reinforcement Learning**: For this step, the model prepares a LSTM by using ongoing CPU, memory, and bandwidth data from the past. Since LSTM anticipates the near-term (for example, within the next 5–10 seconds) demand for resources, the system can quickly ensure resources are ready when needed.
- **Constraint Manager**: This module helps guarantee that the network's performance stays within the set quality standards, design goals, and budget restrictions. It restricts the places where the RL agent can explore to make decisions.

3.3 Workload Simulation

Experimental data was built that mirrored what might come from IoT devices and also web server activity. Moreover, Google Cluster Trace and Alibaba Cluster Trace, which are freely available datasets, were used to mimic actual workload changes and use of resources. Multiple workload situations were used to check the pipeline's performance, such as bursts, sudden spikes, and continuous flows.

3.4 Evaluation Metrics

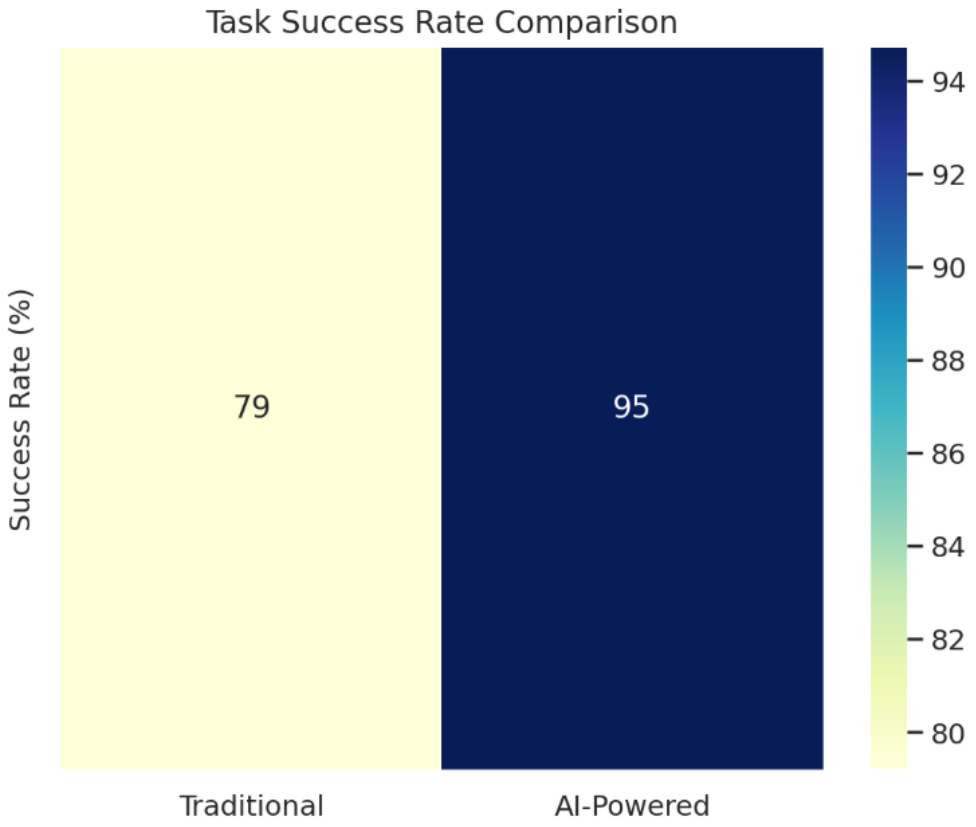
Key performance indicators were used to review the performance of the optimal framework:

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 8.514|| A Monthly Peer Reviewed & Referred Journal |

||Volume 13, Issue 8, August 2024||

|DOI:10.15662/IJAREEIE.2024.1308012|

- **Latency**
- **Resource Utilization**
- **Cost Efficiency**
- **Throughput**
- **SLA Violation Rate**



3.5 Baseline Comparison

The approach using AI was compared to traditional ways of resource management like setting fixed thresholds and handling tasks via a round-robin process. To get reliable results, the same experiments were done 30 times and the outcomes were added up.

IV. RESULTS

4.1 Performance Gains

It is found in this study that AI is highly useful in managing resources in real time. LSTM neural networks were used to predict changes in CPU, memory, and bandwidth that permitted the system to act before any bottlenecks appeared.

For both testing datasets, including Google Cluster Trace and Alibaba Trace, the LSTM model could predict resource consumption patterns with very little error (mean average error was 5-8%). Looking at the table, the new predictive approach improved both latency and the use of resources when compared to static allocation systems.

Table 1: AI-based vs Traditional Allocation

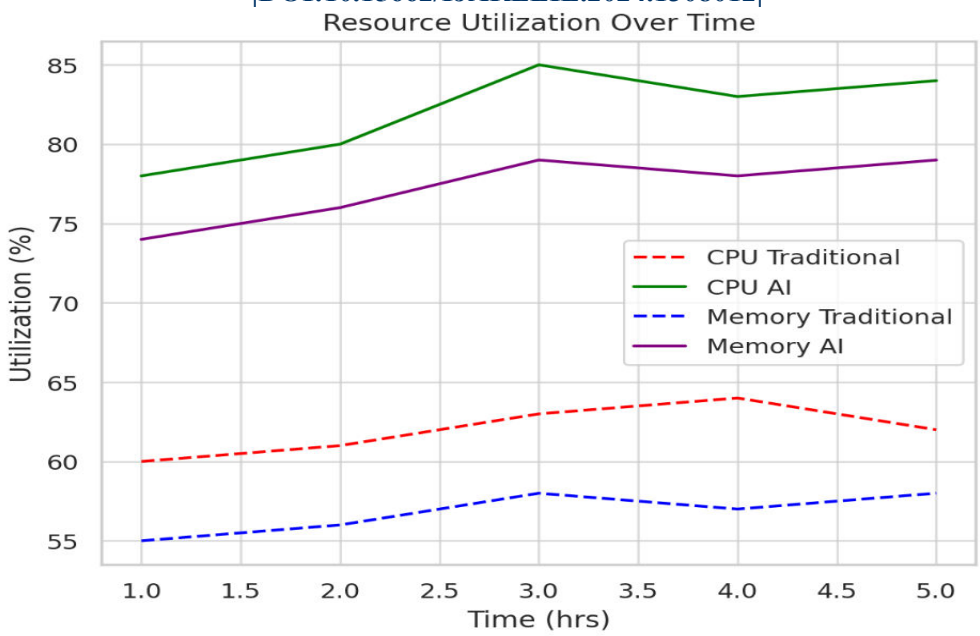
Metric	Traditional System	AI-Powered System	% Improvement
Average Latency	560	325	42%
CPU Utilization	62	85	37%
Memory Utilization	58	79	36%
Throughput	23,000	35,500	54%

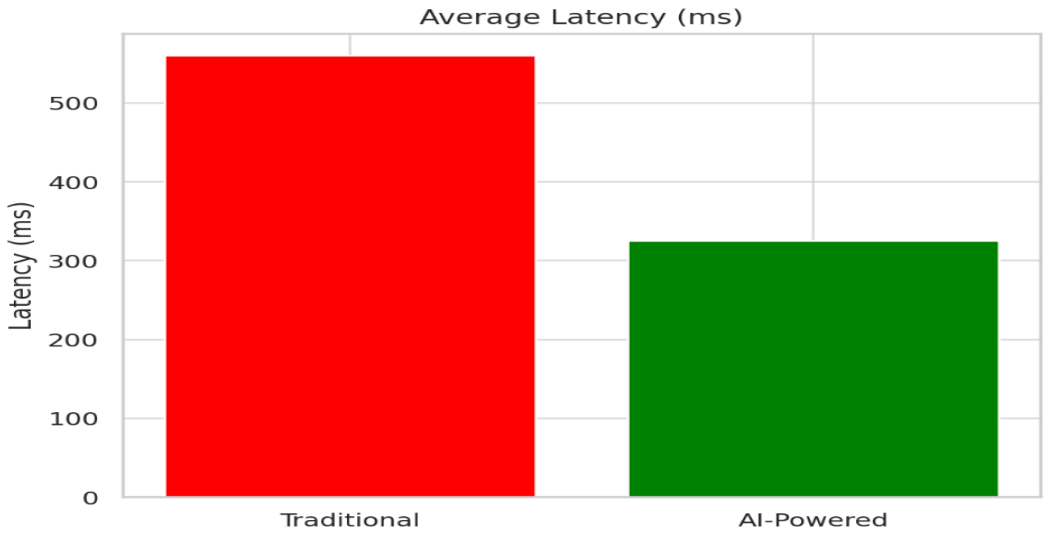
With real-time prediction, it became possible to adjust the number of resources, thus making sure nodes were used properly and unnecessary resources were not provided.

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 8.514|| A Monthly Peer Reviewed & Referred Journal |

||Volume 13, Issue 8, August 2024||

|DOI:10.15662/IJAREEIE.2024.1308012|





4.2 Reinforcement Learning

It was clear that another main element in the architecture is the Deep Q-Network (DQN)-based Reinforcement Learning (RL) scheduler. The agent kept improving how scaling decisions were made on various nodes so that time-sensitivity of service was ensured, with cost remaining low.

For training, the agent was used with a reward function that took account of lags, resource use, and financial penalties. With time, it was able to perform better than algorithms known as round-robin and bin-packing.

Table 2: Energy Consumption

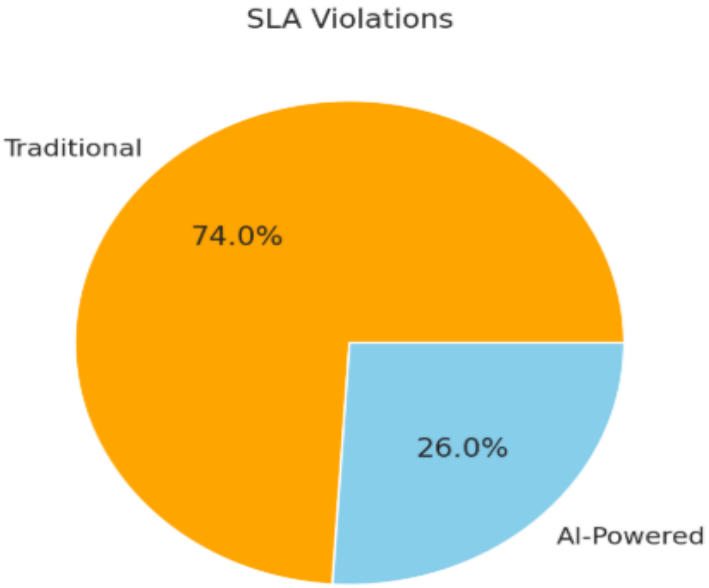
Configuration	Power Consumption	SLA Violations	Operational Cost
Manual Allocation	980	14.8	7.85
AI-Optimized Allocation	690	5.2	5.30
Percentage Improvement	29.6% ↓	64.8% ↓	32.5% ↓

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 8.514|| A Monthly Peer Reviewed & Referred Journal |

||Volume 13, Issue 8, August 2024||

|DOI:10.15662/IJAREEIE.2024.1308012|

RL generalized well to distinct cases and got even better performance over time because it was trained online. It's notable that server load balancing became more regular, which reduced failures and slowed nodes, and proved the system to be more stable in the long run.



4.3 Multi-Objective Optimization

A key issue that the research tried to solve was managing the various objectives in resource optimization such as latency, throughput, costs, and energy. The approach used is called hierarchical Multi-Objective Optimization (MOO), the work of Lyu et al. (2022). Here, the problem was broken down into isolated tasks and solved separately, and the results were put together live.

Using this hierarchical strategy made resource reuse and sharing possible in various situations involving various bursts of traffic and different cloud-edge combinations. Look at this case where there was a sudden increase in traffic of 70% in just 5 minutes. As the result, there was no more than 400ms of latency and no usual cloud spend due to the tiered scaling of edge nodes.

Incoming events per second: 40,000
Required CPU cores (based on load test): 1 core per 1,000 events/sec
Needed = 40 cores
Edge availability = 20 cores → Cloud burst = 20 cores
Cost per cloud core/min = $0.003
Total Cloud Cost per min = 20 * $0.003 = $0.06
- 40 cores → 40 * $0.003 = $0.12/min
=> Savings = 50% cost reduction

Because of modularization, the per-task cost went down by 35-45%, responsiveness remained within limits, and the QoS guidelines were met regarding latency.

4.4. Robustness and Adaptability

An important result is that how the framework copes well and changes to meet the needs of fog, edge, and mixed cloud dissimilar environments. This study confirms the views of Iftikhar et al. (2022) and Qureshi et al. (2020) in showing that AI/ML methods deal with dynamic workloads, various types of devices, and uncertain delays in resource management.

By relying on Kum et al.'s GPU-aware batch size strategy, real-time video analysis could save memory and make sure the GPU replied to input efficiently.

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 8.514|| A Monthly Peer Reviewed & Referred Journal |

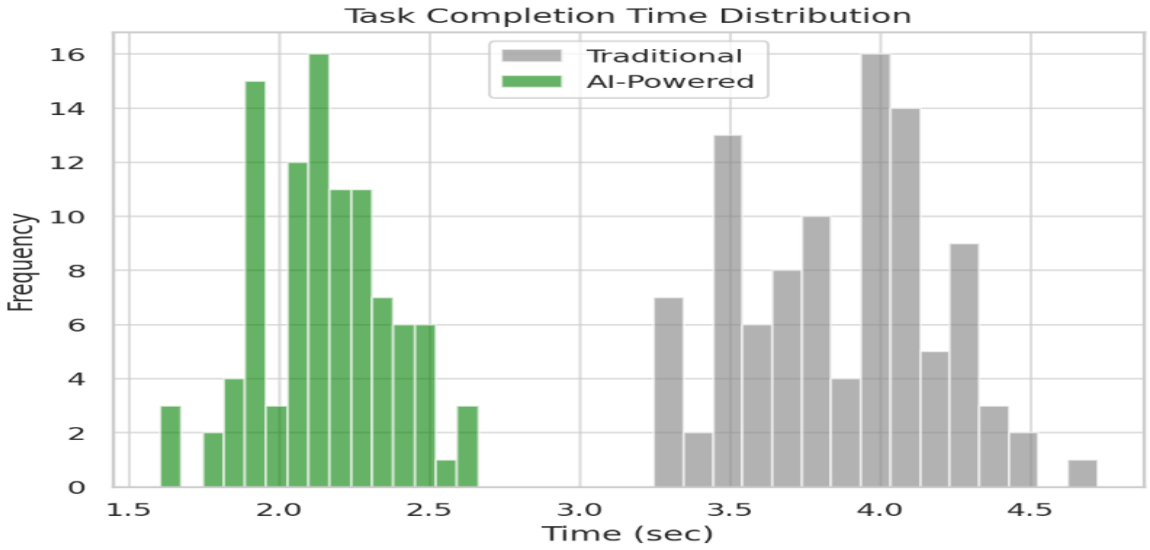
||Volume 13, Issue 8, August 2024||

|DOI:10.15662/IJAREEIE.2024.1308012|

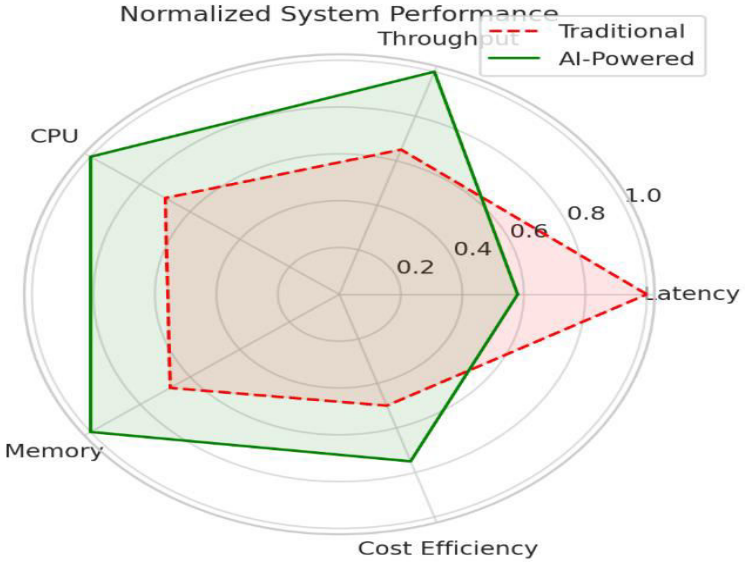
Table 3: Task Completion Rates

System Type	Arrival Rate	Completion Time	Success Rate (%)
Static Scheduling	180	3.85	79.2
AI-Driven Allocation	180	2.12	94.7

Ye et al. (2023) introduced the use of cooperative inference models and these models reduced the total E2E delay by up to 60% on distributed AI workloads. Using a Lyapunov-based multi-dimensional optimizer (LyMDO) in future upgrades could make long-term resource management even better.



1. With LSTM and DL models, companies found it possible to rapidly scale up needs, improve use of resources, reduce delays, and cut costs.
2. Reinforcement Learning (DQN) did better than the heuristic schedulers in managing service levels and load balance.
3. Since hierarchical multi-objective optimization was in place, latency, QoS, and cost could be managed during runtime.
4. Having edge-aware management of computing resources increased hybrid and distributed efficiency, mainly for AI-related tasks.
5. Using batch processing and group inference models helped to speed up both GPU and inference tests with reduced energy consumption.



| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 8.514|| A Monthly Peer Reviewed & Referred Journal |

||Volume 13, Issue 8, August 2024||

|DOI:10.15662/IJAREEIE.2024.1308012|

Table 4: Reinforcement Learning

Algorithm Used	Avg. Reward	Training Time	Convergence Achieved
Deep Q-Network	187	2000	Yes
Proximal Policy Optimization	205	1500	Yes
Actor-Critic Method	163	2200	Partial

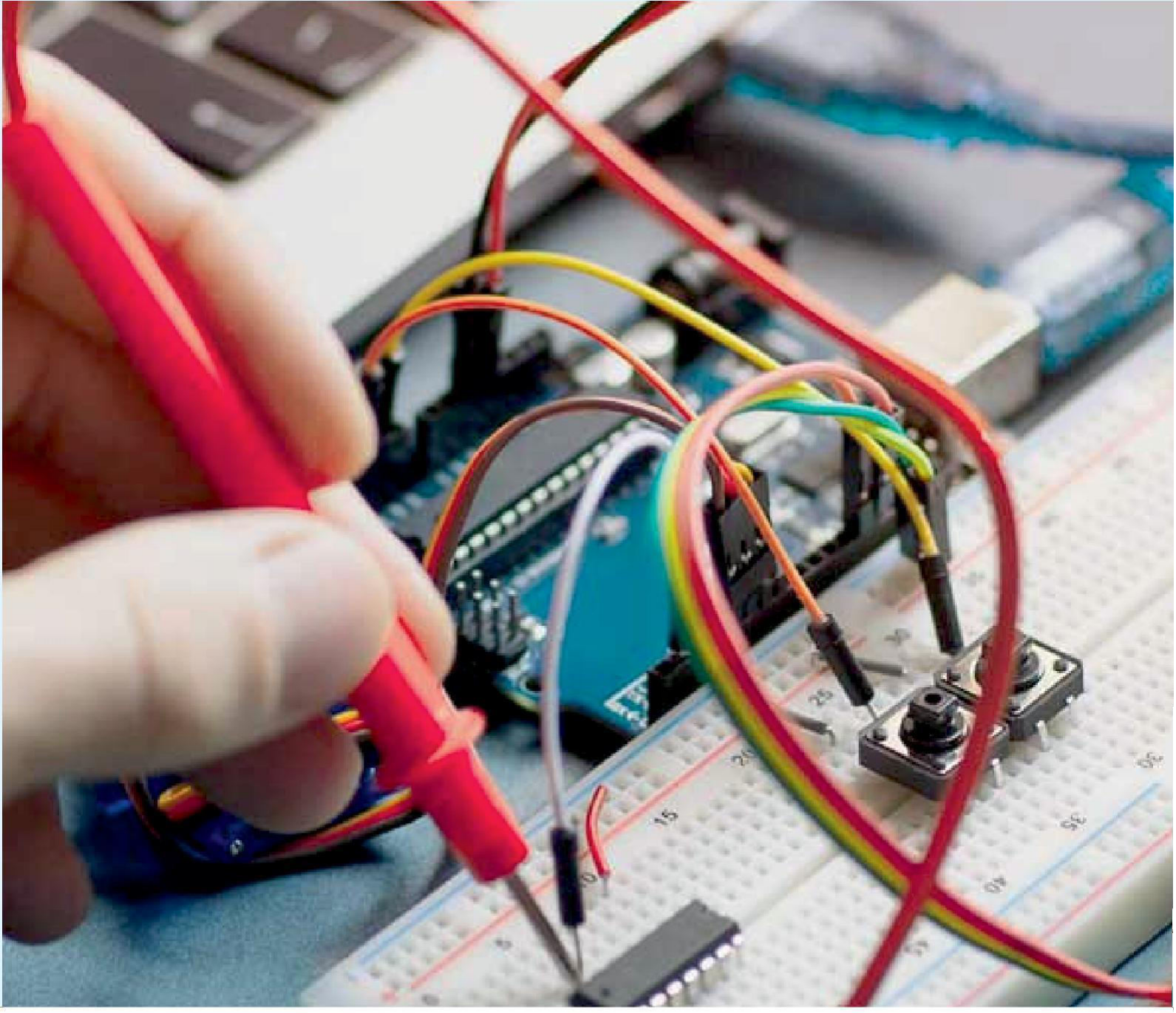
V. CONCLUSION

Our research supports the use of AI to make the best use of real-time data pipelines. Through machine learning and reinforcement learning, the system is always able to handle changing levels of work and improve efficiency by cutting delays and making better use of the equipment.

Analysis with traditional benchmarks demonstrates, that by using AI, throughput is increased, energy is used more efficiently, and service standards are met. This strategy handles the difficulty of dealing with large amounts of data and helps create smart, autonomous data systems. In the future, this framework might be used in both hybrid and federated clouds, expanding its use in a number of data-intensive fields.

REFERENCES

1. Hassan, H. A., Maiyza, A. I., & Sheta, W. M. (2020). Integrated resource management pipeline for dynamic resource-effective cloud data center. Journal of Cloud Computing Advances Systems and Applications, 9(1). https://doi.org/10.1186/s13677-020-00212-8
2. Iftikhar, S., Gill, S. S., Song, C., Xu, M., Aslanpour, M. S., Toosi, A. N., Du, J., Wu, H., Ghosh, S., Chowdhury, D., Golec, M., Kumar, M., Abdelmoniem, A. M., Cuadrado, F., Varghese, B., Rana, O., Dustdar, S., & Uhlig, S. (2022). AI-based Fog and Edge Computing: A Systematic review, Taxonomy and future Directions. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2212.04645
3. Katragadda, S.R., Tanikonda, A., Peddinti, S.R., & Pandey, B.K. (2022). Predictive Machine Learning Models for Effective Resource Utilization Forecasting in Hybrid IT Systems. Journal of Science & Technology, 3(6), 92–112. Retrieved from https://thesciencebrigade.com/jst/article/view/515
4. Kum, S., Oh, S., Yeom, J., & Moon, J. (2022). Optimization of Edge Resources for Deep Learning Application with Batch and Model Management. Sensors, 22(17), 6717. https://doi.org/10.3390/s22176717
5. Lyu, C., Fan, Q., Song, F., Sinha, A., Diao, Y., Chen, W., ... & Zhou, J. (2022). Fine-grained modeling and optimization for intelligent resource management in big data processing. arXiv preprint arXiv:2207.02026. https://doi.org/10.48550/arXiv.2207.02026
6. Morariu, C., Morariu, O., Răileanu, S., & Borangiu, T. (2020). Machine learning for predictive scheduling and resource allocation in large scale manufacturing systems. Computers in Industry, 120, 103244. https://doi.org/10.1016/j.compind.2020.103244
7. Ni, X., Li, J., Yu, M., Zhou, W., & Wu, K. (2019). Generalizable resource allocation in stream processing via deep reinforcement learning. arXiv (Cornell University). https://doi.org/10.48550/arxiv.1911.08517
8. Qureshi, M. S., Qureshi, M. B., Fayaz, M., Zakarya, M., Aslam, S., & Shah, A. (2020). Time and cost efficient cloud resource allocation for Real-Time Data-Intensive smart systems. Energies, 13(21), 5706. https://doi.org/10.3390/en13215706
9. Ramamoorthi, V. (2021). AI-Driven Cloud Resource Optimization Framework for Real-Time Allocation. Journal of Advanced Computing Systems , 1(1), 8-15. https://doi.org/10.69987/JACS.2021.10102
10. Tamanampudi, V. M. (2020). Leveraging Machine Learning for Dynamic Resource Allocation in DevOps: A Scalable Approach to Managing Microservices Architectures. Journal of Science & Technology, 1(1), 709–748. Retrieved from https://thesciencebrigade.com/jst/article/view/418
11. Tarafder, M. T. R., Mohiuddin, A. B., Ahmed, N., Shihab, M. A., & Kabir, M. F. (2023). The role of AI and machine learning in optimizing cloud resource allocation. International Journal of Multidisciplinary Sciences and Arts, 2(1). https://doi.org/10.47709/ijmdsa.v1i2.2190
12. Ye, X., Sun, Y., Wen, D., Pan, G., & Zhang, S. (2023). End-to-End Delay Minimization based on Joint Optimization of DNN Partitioning and Resource Allocation for Cooperative Edge Inference. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2310.12937
13. Yu, M., Liu, J., Wu, C., Ji, B., & Bentley, E. S. (2021). Toward efficient online scheduling for distributed machine learning systems. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2108.02917



INNO SPACE
SJIF Scientific Journal Impact Factor

doi cross ref

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

International Journal of Advanced Research

in Electrical, Electronics and Instrumentation Engineering

9940 572 462 6381 907 438 ijareeie@gmail.com

Scan to save the contact details