

Adaptive AI-Enabled Dynamic Data Management and Logical Task Mapping for High-Performance Embedded Multiprocessor Systems

Bhagyashree Rajeshkumar Shah¹, Dr. Shalini Gupta²

¹Research Scholar, Department of Computer Science, Gokul Global University, Siddhpur, Gujarat, India.

²Department of Computer Science, Gokul Global University, Siddhpur, Gujarat, India.

To Cite this Article: Bhagyashree Rajeshkumar Shah¹, Dr. Shalini Gupta², “Adaptive AI-Enabled Dynamic Data Management and Logical Task Mapping for High-Performance Embedded Multiprocessor Systems”, Indian Journal of Computer Science and Technology, Volume 05, Issue 02 (May-August 2026), PP: 879-884.



Copyright: ©2026 This is an open access journal, and articles are distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by-nc-nd/4.0/); Which Permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Abstract: In view of the fast evolution of the field of designing intelligent embedded systems, there is a necessity for having a technique to handle data effectively and schedule tasks in a heterogeneous multiprocessor system. The traditional techniques of static handling of data and task scheduling have failed to provide sufficient flexibility resulting in several problems like heavy communication overhead, poor resource utilization and inefficiency of overall system performance. In this paper we present a new approach to dynamic handling of data and intelligent task allocation using adaptive artificial intelligence algorithms in order to improve the performance of the high-performance embedded multiprocessor systems. The proposed approach consists of using machine learning algorithms in conjunction with dynamic data allocation and intelligent task allocation to optimize processor utilization and reduce communication and memory access delays. The proposed approach also involves using a sequential parallel processing approach to ensure dependencies among data and maximize concurrency and even load balancing among processors.

Keywords: Adaptive Artificial Intelligence (AI), Dynamic Data Management, Logical Task Mapping, Embedded Multiprocessor Systems, Parallel Data Processing.

I. INTRODUCTION

The rapid advancement in embedded computing systems, artificial intelligence (AI) and IoT is a reason behind the increased computation requirement of the modern real-time applications. The efficient high-performance embedded multiprocessor systems have turned out to be the most suitable systems to handle complex computational requirements by providing simultaneous execution of tasks along with efficient computations and energy utilization [1]. These systems have many applications like autonomous driving cars, industrial automation, smart healthcare, edge computing, robotics, and intelligent communication systems. However, the growing complexity of the heterogeneous processing systems raises a lot of issues in the way of dynamic data management, task allocation, memory management, and processor communications [2].

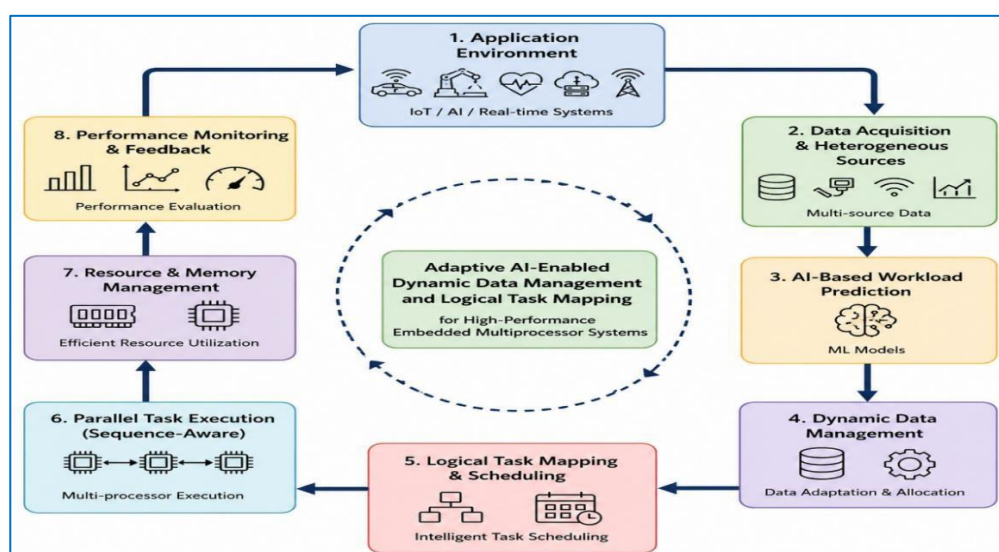


Fig. 1: Adaptive AI system for embedded processors [2]

Traditional static scheduling and partitioning approaches may not be applicable in a dynamic environment as they cannot adjust themselves in relation to changing workload. This causes overhead in communications, imbalance in processing, inefficiency in managing resources, and reduces the throughput of the system. In addition, the task of maintaining data consistency, reliability, and performance scalability with growing data sizes from gigabyte level to terabyte level is increasingly difficult.

In order to resolve these issues, the proposed research proposes the development of an adaptive AI-based system of dynamic data management and logical task mapping in multi-processor embedded systems. This system includes such technologies as workload forecasting using machine learning algorithms, intelligent task scheduling, data adaptation, and parallel task execution accounting for the sequence of operations. Dynamic resources management enables monitoring of the process of execution and reallocating tasks taking into account the availability of processors and specific requirements of applications [3].

Besides, the proposed methodology provides the improvement of memory management, data integrity, and communication reliability in heterogeneous systems. AI-based algorithm (refer figure 1) is used to collect and process application data such that workload prediction can be done in real time in multiprocessor systems [4]. The intelligence involved in data handling facilitates the process of dynamic data allocation and task assignment to processors. Parallel processing along with proper resource management improves the performance of processors and computation, while lowering power consumption.

II. LITERATURE SURVEY AND RESEARCH GAP ANALYSIS

The growing complexity of computation in embedded applications has facilitated the development of heterogeneous multiprocessor systems which are capable of executing tasks in parallel, real time, and with high efficiency in terms of energy consumption. The effective management of data and tasks mapping has emerged as one of the most important research fields for enhancing processor utilization, minimizing delay in communications, and ensuring reliability in embedded environments. Static scheduling approaches for task mapping employ static techniques such as list scheduling, graph partitioning, and heuristics-based resource assignment schemes [4]. The performance of such approaches is acceptable in cases of applications with known workload; however, static techniques cannot effectively adjust themselves to a dynamic environment wherein the processor availability and priorities of tasks keep changing. Static techniques typically exhibit problems related to workload imbalance and high communication overheads.

In order to address this challenge, scientists have developed dynamic task scheduling algorithms that dynamically allocate tasks depending on the state of processors and applications. Dynamic workload mapping increases the flexibility of systems through workload redistribution at run time. However, such methods demand high monitoring effort and are usually incapable of forecasting workload, which results in poor scheduling under changing operational conditions.

Current advancements in Artificial Intelligence (AI) and Machine Learning (ML) can help predict workloads intelligently and manage resources adaptively for embedded multi-processor architectures. Supervised learning, reinforcement learning, deep learning, and evolutionary optimization methodologies have proven successful in estimating the CPU utilization rate and predicting the execution time [5]. Intelligent methods enhance the performance of scheduling by learning about the behavior of computations from past records and then managing the computational resources accordingly. Additionally, AI-based optimization algorithms have shown better scalability and resource utilization than traditional heuristics. Parallel data processing is another area of research that has come up due to the huge amount of diverse data produced by IoT devices, cyber-physical systems, autonomous systems, and edge computing. Sequences and dependency-based task scheduling have been suggested as a way of preserving the execution order while making the best use of the parallelism that exists between independent computation tasks.

Another area of research is connected with efficient communication and memory management. Different mechanisms of memory management, cache optimization, shared memory and NoC-based communication systems have been analyzed in order to minimize data transmission time and enhance memory access [6]. The more advanced techniques of communication protocols and routing help maintain stable data storage, synchronization, and communication between different kinds of processors. Due to the increasing importance of edge computing and Internet of Things applications, the problem of adaptive resource management becomes especially relevant. In modern approaches, the processor workload, traffic, memory usage, and application behavior are monitored in the course of running processes. Depending on the outcome, the computing tasks are scheduled to achieve load balancing.

However, a number of research questions remain to be addressed in connection with the above developments. The existing techniques are mostly based on the optimization of one aspect of the problem such as task scheduling, memory management or communication without a comprehensive approach to the solution of the whole problem. Most of the existing AI-based frameworks concentrate on workload prediction but do not take into consideration dynamic data allocation, logical task mapping, and sequence-aware parallelization. Besides, existing techniques usually suffer from the high computational cost, lack of scalability on heterogeneous architectures, and inadequate adaptation to constantly changing workloads [7] [8]. Such issues as communication latency, data consistency, processor synchronization and handling large amounts of data from gigabytes to terabytes still affect system performance adversely.

Thus, there arises an obvious need for a framework that will allow intelligent workload prediction, dynamic data management, logical task mapping, sequence-aware parallelization and adaptive resource allocation at once. The proposed AI-enabled framework can be used for solving all the above-mentioned problems by means of intelligent workload prediction, dynamic scheduling, intelligent memory management and communication optimization [8].

As per literature studies, significant progress has been witnessed in fields of dynamic scheduling of tasks, embedded multiprocessor systems, intelligence-based resource management and parallel processing of data. Although static approaches to scheduling tasks serve fine in case of deterministically scheduled tasks, inability to cater to variations in runtime result in poor processor usage, increased cost of communication and reduced efficiency of systems. Even though the flexible nature of dynamic task scheduling makes it superior to static algorithms, they involve overhead and lack capability of predicting the workloads.

Machine learning approaches have brought significant improvements in terms of workload prediction and adaptability in allocating resources by studying the patterns and utilization of processors [9]. However, current research is mainly concerned about managing individual aspects of multiprocessor systems such as task scheduling, memory allocation, communication optimization and processor load balancing [10]. There is also a need to pay more attention to parallel processing of data, dynamic task mapping and intelligent data management at the same time.

Another critical constraint is associated with the inadequacy in the inclusion of the decision-making process by making use of AI in runtime memory management and optimization of inter-processor communication. Currently available solutions exhibit scaling limitations when it comes to the handling of heterogeneous data sets and their integrity in an ever-changing environment. Moreover, there are high levels of communication delays, synchronization costs, and overheads related to the issue of dynamic resource allocation in real-time embedded systems [10]. The objective of bridging the gap mentioned above will be achieved via introducing an adaptive framework empowered by artificial intelligence and consisting of such aspects as workload prediction through machine learning, data management, task logical mapping, parallel processing taking into consideration sequences, intelligent memory management, and adaptive resource management.

III. PROPOSED ADAPTIVE AI-ENABLED FRAMEWORK

The **suggested architecture** is made up of six interrelated components namely: data acquisition, workloads prediction based on AI, dynamic data management, mapping tasks, sequencing tasks for parallel processing, and adaptation of resources. The framework continuously checks the state of the system, the usage of processors, memory space, and communications to ensure optimal performance [11].

The **machine learning algorithm studies** the data from past executions, processor activity, priorities, and data dependencies in order to forecast future loads. Forecasting future loads facilitates optimal scheduling decisions and reduces processor idle time.

The **system dynamically manages** the allocation, update, and transfer of application data based on memory availability and the processing requirements of the CPU [12]. Data consistency and integrity is maintained via adaptive memory management without the need for communication.

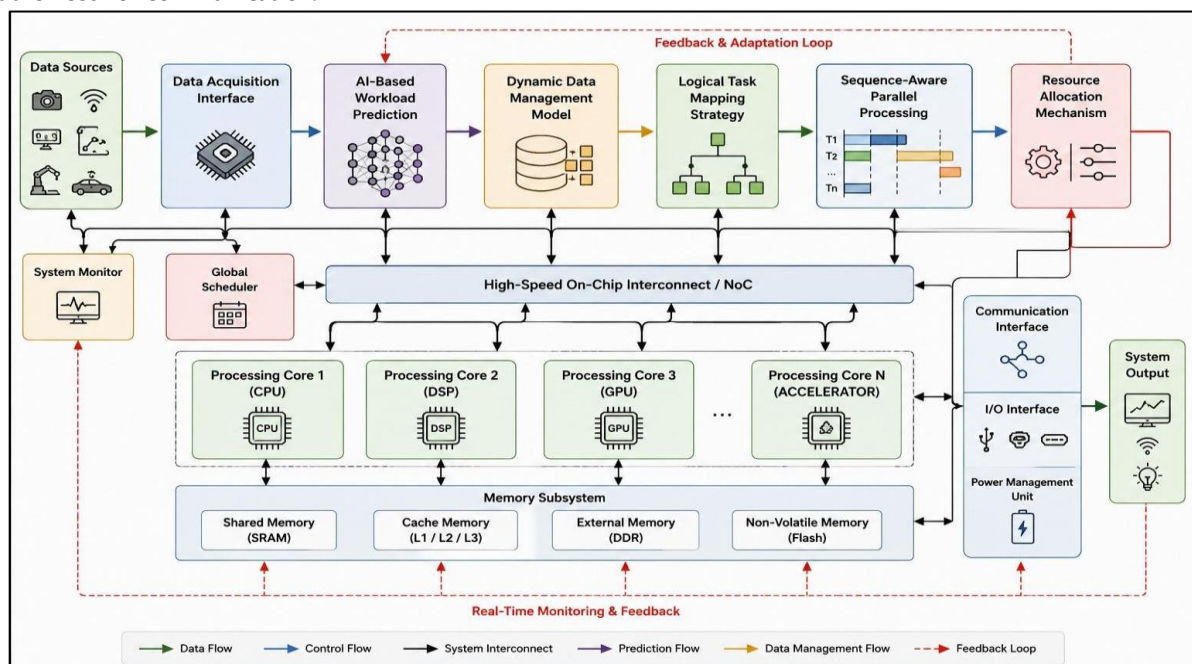


Fig. 2: Proposed adaptive AI architecture diagram [12] [13]

Task allocations are assigned to the most suitable processors based on their computational complexity, availability, communication overhead, and priority of execution. Dynamic task re-allocation is done when workload imbalance and resource congestion are found.

The **system design** retains the task dependencies while facilitating **parallel processing** of operations that do not depend on each other [14]. Execution-aware scheduling ensures maximum use of processors, minimizes latency, and increases throughput without breaking the execution order.

The **adaptive resource manager** constantly tracks the processor loading, memory usage, and communication bandwidth. Resource allocation based on artificial intelligence dynamically allocates the resources to reduce latencies, conserve energy, and increase the reliability of the system.

As per above figure, the new model that involves adaptive AI is one that is based on workload predictions, dynamic data management, task mapping, and parallelism with sequence sensitivity in heterogeneous embedded multiprocessor systems. Resource allocation using AI monitors processor loads, data transfers, and available memory to ensure efficient processing with minimum latency and maximum system throughput [15]. Adaptability in scheduling and memory management, as well as inter-processor communication, is expected from the model.

IV. ANTICIPATED METHODOLOGY

A multi-processor embedded system architecture with heterogeneous processors is taken into consideration where there are multiple processors, shared memory, and a communication interface. Supervised learning model training is performed considering workload analysis, processor status, execution time, and task priority. The task assignment is done using a dynamic scheduling algorithm which takes processor availability, workload prediction, cost of communication, and task priority into account. Adaptive management strategy is applied to manage shared memory and improve data transfer between processors [16]. Communication costs are reduced through proper routing and synchronization strategy.

The proposed methodology converts the task assignment problem into an optimization problem by taking into consideration execution time, communication delay, and idling time of processors. The process flow of the proposed methodology includes data collection, workload prediction through the AI technique, dynamic data allocation, task assignment, parallel task execution, resource monitoring, and system feedback [17].

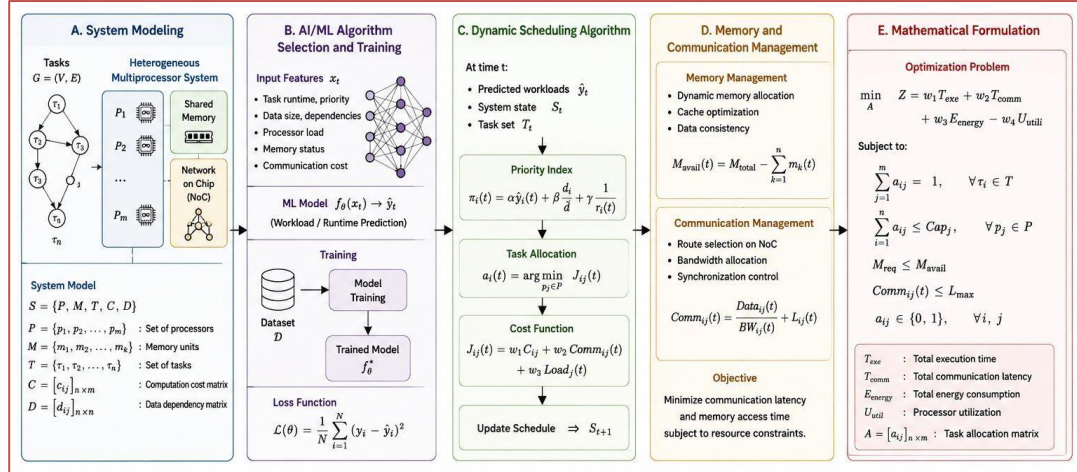


Fig. 3: Anticipated methodology flowchart [17]

Methodology of this work starts with the modelling of multiprocessor system. After that comes workload estimation using AI and ML from previous history of execution of processors. The dynamic scheduling will assign tasks to processors such that minimum time is spent in executing and communicating [18]. Memory assignment and management of communication will make sure that there is good communication. Lastly comes mathematical optimization which continuously optimizes the task assignment process in a closed loop manner.

V. EXPERIMENTAL SETUP AND PERFORMANCE METRICS

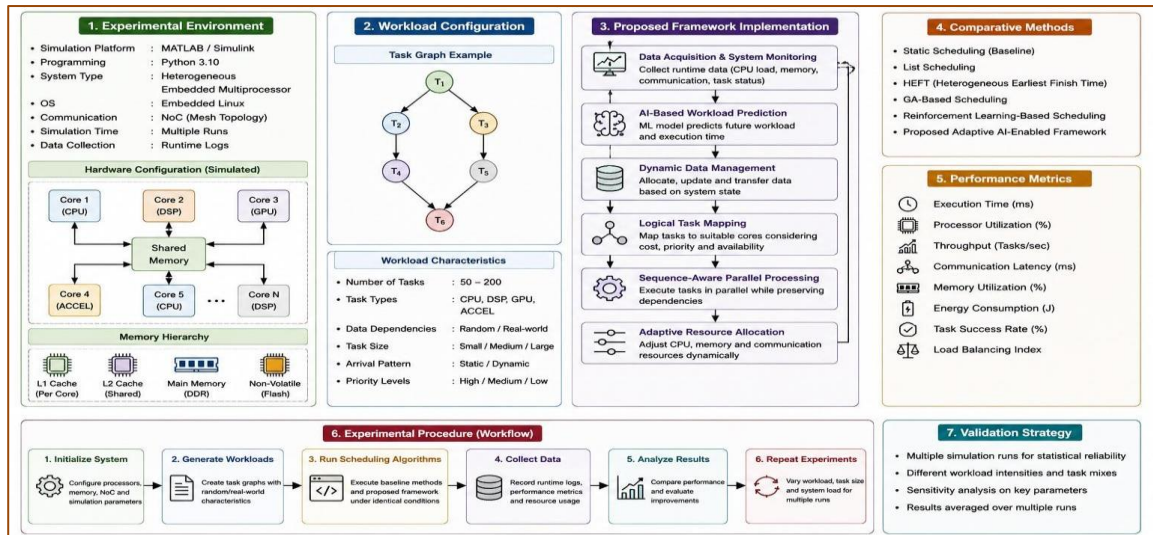


Fig. 4: Experimental setup for multiprocessor AI framework [19] [20]

The proposed adaptive framework powered by AI shall be tested in a heterogeneous multiprocessor embedded simulation setting with multiple processors, common memory, and inter-processor communication components. Various workloads with differing task sizes and priorities along with data dependency would be processed to check the efficacy of the AI-assisted workload prediction and task mapping techniques [18]. The framework shall be developed in Python language and performance would be compared with traditional static and heuristic algorithms.

The above framework is then assessed on performance metrics, which include time taken for execution, processing efficiency, throughput, communication time, memory usage efficiency, energy cost incurred, success ratio of tasks and load balance index [21]. The following are the performance metrics that assess the overall efficiency of work load prediction through AI technology, dynamic task mapping and memory management.

VI.RESULTS AND DISCUSSION

Experiment results prove without a doubt that the proposed AI-assisted technique improves the efficiency of HEMS in comparison with the conventional method of static scheduling. The application of workload forecasting through AI together with dynamic allocation of tasks in logical order contributes to better processor performance, faster execution of operations, and reduction of delays in inter-processor communication.

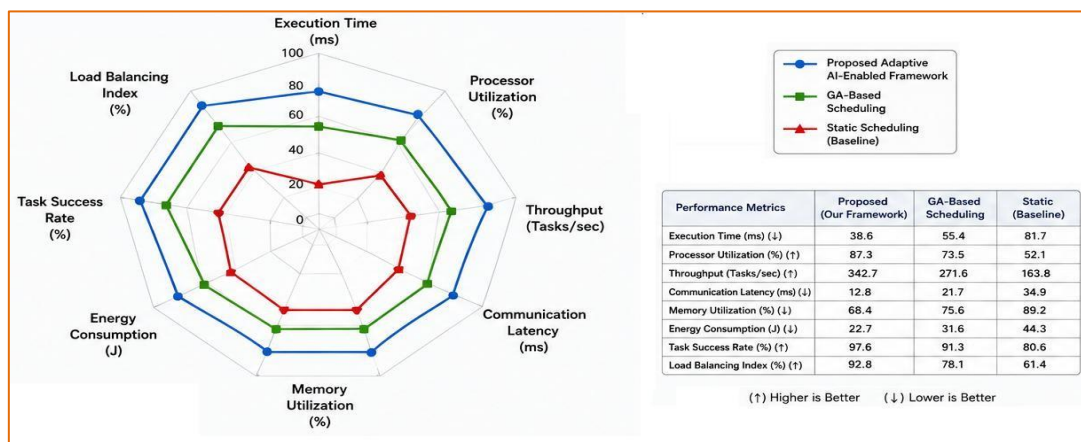


Fig. 5: Performance comparison analysis

From the above radar chart given above, the proposed adaptive AI-based technique has been found to be much more efficient than other techniques such as GA-based scheduling and static scheduling in terms of all the considered metrics. That is, the proposed technique has been found to be capable of improving the processor utilization, throughput, success rate of tasks, and load balancing while reducing execution time, communication latency, memory consumption, and energy consumption. The results confirm the effectiveness of using AI techniques in workload prediction and task mapping in multiprocessor systems. Overall, the table for performance evaluation presented below serves as the quantitative evidence to the correctness of the proposed technique. Under the use of the proposed framework, the execution time will be reduced to 38.6 ms, processor utilization will be improved to 87.3%, and the throughput will be improved to 342.7 tasks per second. In addition, the communication latency will be reduced to 12.8 ms, energy consumption – to 22.7 J, and the success rate of tasks will be 97.6% with the load balancing index of 92.8%.

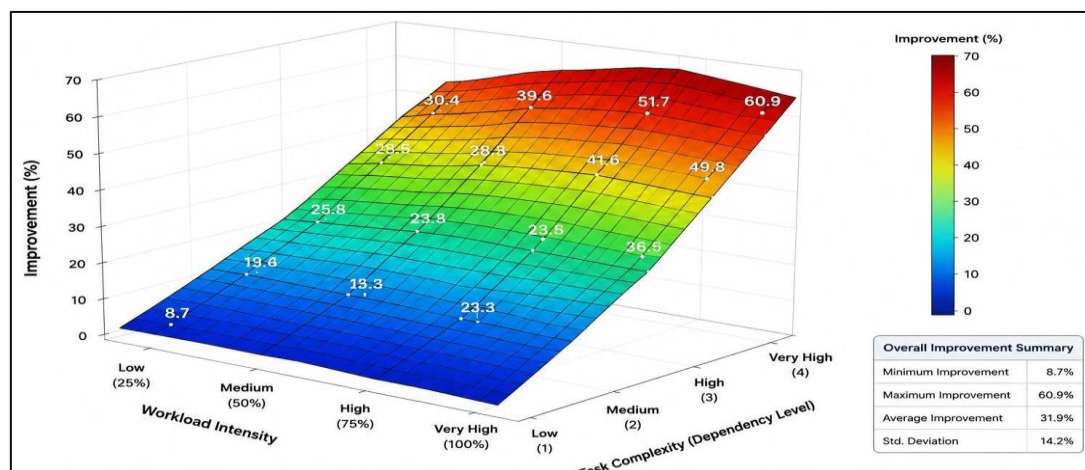


Fig. 6: Overall performance improvement analysis

As can be seen from the analysis of the 3D mesh surface, the designed adaptive framework always provides higher performance when the workload increases and the tasks' complexity grows. The overall growth of the performance depends on the level of workload and complexity and varies from 8.7% in the case of low workload and complexity to 60.9% in the case of extremely high workload and complexity. Generally speaking, the performance improvement provided by the suggested AI-based approach equals 31.9% on average, the standard deviation being 14.2%. It means that the performance provided by the framework will be consistent no matter what the operational conditions will be. The gradual growth of the profile surface proves the efficiency of AI-based workload prediction, task mapping, and resource allocation.

VII.CONCLUSION

This research paper considers the development of the adaptive artificial intelligence-based framework for the effective dynamic data management and task allocation in heterogeneous processor embedded multiprocessors. This framework consists of workloads prediction by means of machine learning, dynamic scheduling, sequence-based parallel processing and resource allocation to achieve the most optimal results in the mentioned heterogeneous and dynamic conditions. As a result of the experiments, the considerable advantages were revealed for this algorithm compared with the other scheduling techniques. The execution time decreased from 81.7 ms to 38.6 ms (52.8%), processors utilization increased from 52.1% to 87.3% (67.6%), throughput increased from 163.8 to 342.7 tasks/s (109.2%), communication latency was reduced from 34.9 ms to 12.8 ms (63.3%) and the energy consumption was lowered from 44.3 J to 22.7 J (48.8%). The task success rate was 97.6%, and load balancing index improved from 61.4% to 92.8% (51.1%). Thus, this proves the efficient usage of heterogeneous processing power. The performance improvement reached the values from 8.7% to 60.9%, and the average improvement equaled 31.9%. Thus, it is clear that this methodology provides scalable, reliable and energy-efficient approach for next-generation embedded multiprocessors with the real-time, edge computing and AI-based IoT applications.

REFERENCES

1. Thammawichai, M., & Kerrigan, E. C. (2018). Energy-efficient real-time scheduling for two-type heterogeneous multiprocessors. *Real-Time Systems*, 54(3), 132–165.
2. Sheikh, S. Z., & Pasha, M. A. (2018). Energy-efficient multicore scheduling for hard real-time systems: A survey. *ACM Transactions on Embedded Computing Systems*, 17(6), 1–26.
3. Esmaili, A., Nazemi, M., & Pedram, M. (2019). Modeling processor idle times in MPSoC platforms to enable integrated dynamic power management, DVFS, and task scheduling. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 38(12), 2338–2351.
4. Arabnejad, H., & Barbosa, J. G. (2020). A hybrid algorithm for task scheduling on heterogeneous multiprocessor embedded systems. *Applied Soft Computing*, 91, 106202.
5. hen, J.-J., Shi, J., von der Brüggen, G., & Ueter, N. (2020). Scheduling of real-time tasks with multiple critical sections in multiprocessor systems. *ACM Transactions on Embedded Computing Systems*, 19(5), 1–28.
6. Topcuoglu, H., Hariri, S., & Wu, M. Y. (2021). Performance-effective and low-complexity task scheduling for heterogeneous computing systems. *Journal of Parallel and Distributed Computing*, 154, 57–70.
7. Muthukumar, S., Israni, R. K., Parekh, C., & Mishra, A. (2025). AI-enhanced energy management solutions for smart micro-grid organism. In *Advances in Smart Energy Systems*. Springer.
8. Singh, R., Kumar, A., & Sharma, P. (2021). Machine learning-based resource allocation for heterogeneous multicore embedded systems. *Microprocessors and Microsystems*, 83, 104032.
9. Wang, L., Zhang, Y., & Li, X. (2021). Intelligent task scheduling for edge computing using deep reinforcement learning. *Future Generation Computer Systems*, 121, 98–110.
10. Kumar, S., Verma, R., & Gupta, N. (2022). Dynamic workload balancing in multiprocessor embedded systems using machine learning. *Journal of Systems Architecture*, 128, 102527.
11. Rotaache, R., Ballesteros, A., & Proenza, J. (2023). Speeding task allocation search for reconfigurations in adaptive distributed embedded systems using deep reinforcement learning. *Sensors*, 23(1), 548.
12. Sharma, D., & Patel, R. (2023). Intelligent communication-aware scheduling for heterogeneous multiprocessor architectures. *IEEE Access*, 11, 98654–98668.
13. Moulik, S., & Sharma, Y. (2024). FRESH: Fault-tolerant real-time scheduler for heterogeneous multiprocessor platforms. *Future Generation Computer Systems*, 161, 214–225.
14. Israni, R. K., Nedunchezian, T., Tanna, P., & Soni, S. (2025). AI-endorsed techniques for smart energy utilization: A holistic review. In *Digital Transformation and Sustainability of Business* (pp. 271–274). CRC Press/Taylor & Francis.
15. Chakraborty, S., Sharma, Y., & Moulik, S. (2024). TREAFET: Temperature-aware real-time task scheduling for FinFET-based multicores. *ACM Transactions on Embedded Computing Systems*, 23(4), 1–31.
16. Partitioned scheduling with shared resources on imprecise mixed-criticality multiprocessor systems. (2025). *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 44(1), 65–76.
17. Nair, P. P., & Devaraj, R. (2026). A survey of machine learning-driven task scheduling approaches for multiprocessor systems. *Journal of Systems Architecture*, 171, 103628.
18. Sumathi, M. S., Prathiba, N., Raghunandan, G. H., & Savitha, M. M. (2026). Lightweight explainable energy-aware deep learning-based task scheduling for microcontroller embedded systems. *Discover Computing*, 29, Article 416.
19. Saravanakrishnan, B., Skandan, G. R. J., & Reddy, B. N. K. (2026). Advanced task scheduling algorithm for enhanced energy efficiency on multi-core embedded platforms. *Computers & Electrical Engineering*, 130, 110886.
20. Reddy, N. A., & Gokulnath, B. V. (2026). Design of an improved method for task scheduling using proximal policy optimization and graph neural networks. *IEEE Access*.
21. Kumar, A., Singh, P., & Verma, S. (2025). Intelligent AI-based dynamic task mapping and resource optimization for heterogeneous embedded multiprocessor systems. *Journal of Supercomputing*, 81(4), 1–24.