Journal of Global Research in Electronics and Communication
Volume 1, No. 10, October 2025
Available Online at: www.jgrec.info
RESEARCH PAPER

Reinforcement Learning Approaches for Energy-Efficient Embedded Systems: A Survey

Dr. Amit Jain
Professor
Department of Computer Science and Engineering
OP Jindal University
Raigarh (C.G)
amit.jain@opju.ac.in

***Abstract*—The increased usage of embedded systems in areas like mobile computing, biomedical applications, industrial automation, and Internet of Things (IoT) has exacerbated the need to operate the embedded systems intelligently with a focus on energy efficiency under tight computational and power limitations. Reinforcement Learning (RL) is a potential solution to optimize power consumption and system performance with adaptive and data-driven real-time decision-making. This article is an in-depth survey of RL-based approaches to embedded systems design, with a special focus on model-free and model-based learning, energy-aware learning, and other lightweight learning algorithms in resource-constrained systems. Important uses are Dynamic Voltage and Frequency Scaling (DVFS), CPU scheduling, real-time object detection and autonomous control of embedded robotics. Simulation environments like MATLAB/Simulink, OpenAI Gym, and Network Simulator 3 (NS-3), as well as common hardware platforms, like ARM Cortex-M, NVIDIA Jetson, and Texas Instruments MSP430. In literature, it is possible to identify the presence of significant achievements, including up to 47% power savings and latency reductions with Deep RL and adaptive Convolutional Neural Networks (CNNs). Nonetheless, there remain barriers to safe policy learning, deployment in real-time, and reliability in changeable environments. The paper ends with some of the main research findings, such as a scalable RL framework, energy-aware reward functions, and sophisticated simulation techniques on next-generation intelligent embedded systems.**

Keywords*—*Reinforcement Learning (RL), Energy Efficiency, Embedded Systems, Dynamic Voltage and Frequency Scaling (DVFS), Wireless Sensor Networks (WSN).

I. Introduction

The fast advancement of electronic and mobile technologies has greatly accelerated the rise of embedded systems into many areas such as mobile computing, automotive electronics, wearable computing, industrial automation and the IoT. These systems are agile, task-oriented, with stringent constraints, especially energy consumption [1]. Besides traditional task-oriented behaviour, modern embedded systems demonstrate more complex functionalities, such as wireless communication, high-resolution video, and multi-tasking capability - thus, energy efficiency becomes integral in terms of design parameters. Unlike general-purpose processors that can "afford" a higher power budget, embedded systems are already limited to some type of finite energy source, some limiting them to less than a few milliwatts [2]. The differences create a wide range of requirements concerning management, to ensure its resources are used efficiently without depleting energy reserves within a reasonable timeframe from expected operation.

Conventional energy saving techniques rely on pre-defined control models and static configurations to achieve a trade-off between performance and power consumption. Although these energy-saving strategies may be helpful, they are not adaptable to dynamic workloads and unknown environments [3]. Overcoming these constraints, recent research has considered Reinforcement Learning (RL) - a form of ML that enables agents to learn the best actions through trial-and-error interactions with an environment [4]. RL techniques have the potential to implement intelligent policies for power management that learn from real-time feedback from the system to make optimal decisions based on dynamic system parameters, such as voltage levels, task scheduling [5], and core utilization.

Extending on these motivations, one can see that Reinforcement Learning (RL) and in particular its deep and hierarchical variants are becoming an eye opener to transform energy-aware design of embedded systems[6]. Several recent review articles and surveys have highlighted the importance of frameworks of RL to accommodate the complex, real-time embedded systems and how they could be used to exceed the static control techniques and continually learn, through the feedback of the system, to make optimal decisions in a variety of settings, including voltage scaling, task scheduling, and resource allocation [7]. These RL techniques are now proving especially useful when embedded in heterogeneous and restricted-resource platforms, in which they are capable of dynamically trading off performance, energy consumption, and thermal constraints--not only resulting in energy savings but also in greater adaptability and robustness to a wide range of operating conditions.

This study aims to address the energy efficiency issues of embedded systems due to rising complexity and constrained power resources. It discusses the shortcomings of traditional energy management approaches and highlights the possibilities of Reinforcement Learning (RL) to provide flexible, intelligent power optimization even as workloads change to maximize performance, flexibility, and overall system reliability.

A. Structure of the Paper

This paper is organized as follows: Section II discusses the fundamentals of embedded systems and energy efficiency. Section III explores RL applications in energy optimization, including CPU scheduling, DVFS, and WSNs. Section IV covers system architectures, simulation tools, and hardware platforms. Section V presents a literature review, and Section VI concludes with key findings and future research directions.

II. Fundamentals of Embedded Systems and Energy Efficiency

It is essential to develop embedded systems with low energy consumption since they are implemented as battery-powered devices in many applications. Such low-power applications need extremely effective utilization of the electrical energy that is available [8]. Battery-operated devices, wireless and mobile communication devices, consumer electronics, and biomedical applications are examples of typical low-power applications. Because CMOS is an energy-efficient technology, CMOS circuits are employed in these applications to minimize power usage.

A. Model-Free vs. Model-Based Reinforcement Learning

Model-Free (MFRL) and Model-Based (MBRL) approaches are two types of reinforcement learning (RL) techniques for energy-efficient embedded systems. MFRL techniques that do not mimic system dynamics, including Deep Q-Network (DQN) and Deep Deterministic Policy Gradient (DDPG), develop optimum policies from environmental interactions [9][10]. Although MFRL methods have achieved good performance on high-dimensional continuous control problems, they usually require a large number of interactions, which is challenging in embedded systems that have limited computation and power resources.

In contrast, MBRL methods learn a model of the environment (or dynamics) and use this model to plan, or assist in improving learning, with the result being improved sample efficiency and convergence rates. This is especially interesting for embedded platforms where data collection is expensive and sensitivity to power, as seen of MBRL results on reducing energy overheads using imitation learning or self-supervised learning to improve upon the efficiency of scheduling and actuation policy.

Even though MBRL typically displays a more favorable data efficiency and is more useful in resource-constrained embedded systems, it often suffers from model inaccuracies that inevitably lessen performance. While MFRL is generally more robust for complex environments, it is often extremely computationally expensive, or at least ill-suited to embedded system scenarios without some sort of specialized computational acceleration involved. Therefore, the final decision to commit towards MFRL or MBRL in embedded contexts often be based on which is acceptable Considering the system's complexity, the learning resource constraints involved, and the trade-offs on power and performance.

B. Purpose of Embedded Systems

Embedded systems are occasionally used as controllers to manage a particular device function. Usually, they are made to do this function just once, while more sophisticated embedded systems have the ability to manage whole operating systems. Despite being relatively basic tasks that do not require a lot of processing power, some more complex embedded systems can also do a variety of tasks [11]. Once designed to serve a certain function, embedded systems operate reliably and require little intervention because they are often not programmable. Some embedded devices, however, may have their software modified to improve expected functionality. Built and configured to fulfil a particular function, an embedded system is a very reliable electronic component that requires little maintenance and is very easy to add to a device. Despite being an essential part of a system, they are very unlikely to malfunction and do not require reprogramming, which makes them a crucial part of many systems that need to function independently or without help, including household appliances.

C. Energy Consumption Prediction

Economic growth, technological improvement, and population expansion are all contributing factors to the increase in energy consumption. Furthermore, all known energy sources are predicted to run out in a few decades due to the present level of energy exploration and usage. Energy has been thought to have a major role in life. Regarding primary energy resources, Bangladesh's position is rather weak in relation to global energy [12]. Bangladesh has to expand its infrastructure on a larger scale since its proven resources of coal, hydropower, natural gas, and oil are limited. About 2.2% more primary energy was consumed, which is the biggest rise since 2013. The fuel categories with the biggest increases in energy consumption were natural gas and natural gas-fired power plants, followed by oil and renewable energy sources. Comparing renewable energy to non-renewable energy, renewable energy still accounts for a modest portion of the world's energy portfolio. For instance, oil is the primary fuel used today and the most significant non-renewable resource, contributing 34.2% of global energy consumption in 2017. Despite the exclusion of traditional biomass usage, only 10% of 2016's total energy consumption came from contemporary renewable energy sources. This proportion is expected to rise in the future as nations lower their high levels of fossil fuel-based energy use.

III. Applications of RL in Energy-Efficient Embedded Systems

Reinforcement Learning (RL) has emerged as a sound tool of enhancing embedded systems' energy efficiency [13]. Energy efficiency in embedded systems shares numerous features with RL decision-making in the sense that RL may successfully be implemented in many aspects of energy efficiency in embedded systems due to its intelligent, adaptive and real-time decision-making abilities. The Dynamic Voltage and Frequency Scaling (DVFS) application is an illustration of RL. DVFS leverages upon the RL algorithms (e.g. Q-learning), to automatically control the voltage and frequency of the CPU in an attempt to minimize power and yet satisfy performance requirements. The dynamic management of a system in terms of CPU overhead can be achieved via implementation of RL [14]. The RL basics enable one to conserve energy without explicit requirements of predetermined regulations. Task schedule and CPU resource management can also use RL elements as in this case the agents can be taught how to manage the computational resource to efficiently deploy resources and strike a sweet balance between execution times and expended energy. Relative to wireless sensor networks (WSNs), focuses on power control and sleep scheduling in wireless sensor networks (WSNs), so that the long-time use of the networks can be accomplished. RL can also directly perform robotic control in real-time, and instead of latency-sensitive tasks such as neural networks, it would enable nodes to learn new tasks (e.g. adaptive CNN scaling/preferential model pruning) and consume less power than executing vision-based tasks.

A. RL-based Development of CPU Scheduling Techniques

CPU scheduling is a critical factor influencing the performance and energy efficiency of embedded systems. Scheduling algorithms are generally categorized into two types: (1) non-preemptive, where once a task starts execution, it runs to completion, and (2) preemptive, where tasks can be interrupted to allocate CPU resources to higher-priority jobs [15]. Traditional scheduling methods aim to optimize various criteria such as maximizing throughput, minimizing turnaround time, response time, and CPU overhead.

Reinforcement Learning (RL) offers a promising approach to enhance traditional scheduling by learning optimal scheduling policies based on system state and workload behavior. In RL-based scheduling, an agent observes the system's CPU load, task queue, and energy consumption patterns, then learns to make decisions that balance energy savings and performance. Unlike rule-based schedulers, RL agents adapt over time to changing workload conditions, leading to improved energy efficiency without sacrificing system responsiveness. To make quick scheduling selections among several available processes, Figure 1 lists the main CPU scheduling strategies.

For instance, it is possible to train Q-learning or deep reinforcement learning models to respectively prefer lightweight or time-critical tasks upon low-power state or schedule workloads in a manner that minimizes idle power. Adaptive scheduling is highly advantageous in embedded systems where available resources are limited and energy efficiency is critical.

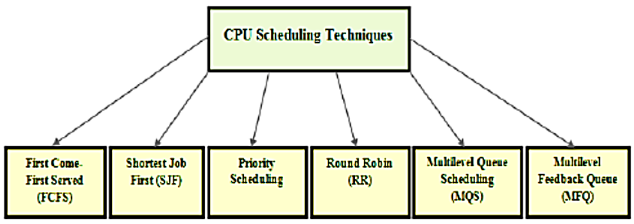

Fig. 1. Common CPU Scheduling Techniques

B. RL-Driven DVFS for Multi-Power Domains

In embedded systems, dynamic voltage and frequency scaling, or DVFS, is a crucial technique for lowering power consumption. The RISC core, geometry processor (GP), and rendering engine (RE) are examples of current GPU components that function in separate power domains and necessitate dynamic performance scaling [16]. Traditional DVFS schemes use fixed control logic and feedback loops, which often lead to inefficiencies, especially in multi-domain systems due to area and power overhead.

Reinforcement Learning (RL) offers a more adaptive and intelligent solution. By observing system parameters, such as FIFO occupancy and workload patterns—RL agents can learn optimal voltage and frequency settings in real-time. This enables efficient control of Power Management Units (PMUs) without relying on bulky digital components. RL-driven DVFS not only improves energy efficiency but also supports scalable, low-overhead designs suitable for integration as reusable Intellectual Property (IP) in energy-aware embedded systems.

C. RL Approaches to Power Management in WSNs

In Wireless Sensor Networks (WSNs), energy preservation techniques are essential due to the limited battery life of sensor nodes and the often-inaccessible nature of their deployment environments [17][18]. These methods are often divided into two main groups: power management and power control, as shown in the comparative analysis in Table I.

- Power control focuses on minimizing energy consumption by adjusting the transmission range of sensor nodes dynamically. Therefore, an effective power control strategy allows communication between nodes while maintaining connectivity and Quality of Service (QoS) level for the network. Since in most off-the-shelf sensor nodes, the transmission power represents approximately 70% of the total energy consumed by the node, knowledge of the environment and reliable control of the transmission power is essential for long-term operation of the network. The purpose of power control is to optimize the amount of energy used in the transmission of data while ensuring that the connectivity between the nodes is extended and that the overall communication quality remains high.
- Power management consists of turning off unnecessary radios, or an entire radio to save energy [19]. Power management features can also be useful to manage rechargeable batteries and prolonging the node's life. Some of the key power management features and techniques were: power-gating (deactivating unused components to remove leakage power), avoiding voltage regulators (which induce power conversion losses), and power matching (making sure there is no energy loss between power supply and load demand).
- Reinforcement Learning (RL), you can take a more dynamic, intelligent approach to power control and management in WSNs. RL algorithms, such as Q-learning or Deep Q-Networks, learn from the environment and employ observations of state to adapt actions or determine the appropriate situation in which to reduce transmission power or transition nodes to low-power sleep modes. These learning-based approaches can adjust to ephemeral network conditions, variable traffic patterns, and fluctuating energy availability, thus enhancing energy efficiencies and providing increased operational life for WSNs [20]. Compared to static approaches, such as rules, RL techniques are characterization by continual evolution and adaptation, making them uniquely powerful for operational environments in embedded applications—such as WSNs—that are characteristically energy constrained.

TABLE I. COMPARATIVE SUMMARY OF ENERGY OPTIMIZATION TECHNIQUES IN WSNS

Aspect	Power Control	Power Management	RL-Based Approaches
Primary Goal	Minimize energy via transmission power adjustment	Reduce energy by turning off idle components	Learn and adapt optimal energy-saving actions dynamically
Methodology	Adjust node transmission range to maintain connectivity and QoS	Use techniques like power-gating and avoiding voltage regulators	Use RL algorithms (e.g., Q-learning, DQN) for adaptive control

Adaptability	Low – fixed rules or thresholds	Moderate – rule-based but not workload-aware	High – adapts to traffic, topology, and energy fluctuations
Energy Saving Potential	Moderate – especially in communication-heavy scenarios	High – reduces leakage and idle power	High – optimized decisions based on system state and environment
Complexity	Low to Moderate	Low	High – requires training and computational support
Suitability	Static or moderately dynamic WSNs	Ideal for periodic sleep-wake duty cycling in sensor nodes	Best for dynamic, unpredictable, or mission-critical WSN deployments

IV. System Architectures and Implementation

This section describes how Reinforcement Learning (RL) is integrated into embedded systems, emphasizing the hardware platforms, simulation tools, control flow, energy efficiency, and algorithm selection that are necessary to implement intelligent embedded solutions that are resource-constrained, adaptive, and energy-aware.

A. RL Integration in Embedded System Design

Incorporating Reinforcement Learning (RL) into the process of embedded system design holds the potential to develop intelligent adaptive behaviour, while improving energy efficiency but must account for the limitations of environments such as an embedded system with regards to computational capabilities, memory size, and energy constrained budgets [21], all while ensuring responsive and reliable performance in real-time.

In embedded systems, RL agents are typically positioned within the system control loop - meaning that the RL agent is able to see system states continuously (e.g., CPU temperature, power consumption, task queue) and also observe optimal actions (e.g., change voltage-frequency, schedule tasks, sleep modes), and learn via closed-loop learning, creating an autonomous learning system that can respond to workload and environmental changes, enhancing performance and improving battery life.

B. Key Considerations in RL Integration

- **Lightweight RL Algorithms:** Embedded systems typically cannot support computationally heavy algorithms like full-scale Deep Q-Networks (DQNs). Therefore, lightweight and memory-efficient algorithms such as tabular Q-learning, SARSA, or linear function approximators are commonly used. In more capable embedded platforms, compressed or pruned deep neural networks can be implemented to enable deep RL without exceeding hardware limitations.
- **On-Device vs. Offloaded Learning:** On-device learning allows for autonomous operation but may lead to increased energy consumption due to prolonged computation. To address this, many systems use a hybrid learning approach training the RL model in the cloud or on edge servers and then deploying the trained model to the embedded device for inference only. This reduces energy usage while still benefiting from intelligent control.
- **Control Flow Integration:** The RL agent is integrated into the system's decision-making flow. It collects inputs from various sensors, analyzes current operational states, and chooses actions that optimize predefined objectives (such as minimizing energy usage or reducing latency) [22]. Feedback from the environment, including the results of past actions, is used to refine the policy over time.
- **Adaptability and Robustness:** RL's ability to learn from real-time feedback makes it particularly suited for non-deterministic or dynamic embedded environments. For example, an RL-powered embedded controller can adjust to variations in workload intensity, temperature fluctuations, or battery degradation—without requiring pre-programmed rules.
- **Energy-Aware Learning:** Specific reward functions can be crafted to penalize energy-inefficient behaviors and reward long-term energy conservation. This drives the RL agent to find energy-optimal policies over time, aligning learning objectives directly with the system's operational goals.

C. Simulation Tools and Hardware Platforms Used

The development and evaluation of Reinforcement Learning (RL) techniques for embedded systems often rely on simulation environments and specific hardware platforms to model system behavior, validate performance, and test energy efficiency strategies.

1) Simulation Tools

Simulation tools are essential for prototyping RL-based control strategies before deployment on real hardware [23]. These tools offer safe, scalable, and repeatable environments for evaluating the impact of energy-aware decisions.

- **MATLAB/Simulink:** Frequently used for modeling embedded control systems with integrated RL toolboxes for reward shaping, training, and deployment testing.
- **OpenAI Gym:** Widely used for prototyping and benchmarking RL algorithms. It can be customized to simulate embedded system scenarios like task scheduling or thermal management.
- **NS-3:** A network simulator used in Wireless Sensor Networks (WSNs) and IoT studies, often extended with RL modules to simulate energy-aware communication protocols.
- **OMNeT++:** Another discrete event simulator used for networked embedded systems with support for mobility, battery modeling, and integration of external RL logic.
- **Custom Simulators:** Many studies build domain-specific simulators to replicate specific embedded use-cases, such as CPU-GPU scheduling, voltage scaling, or sensor duty cycling.

2) Hardware Platforms

To validate RL algorithms in real-world embedded systems, researchers use low-power and resource-constrained platforms that reflect deployment conditions.

- **Raspberry Pi and NVIDIA Jetson Series:** Used for testing RL models in moderately resource-rich environments, enabling both inference and limited on-device training.

- **ARM Cortex-M and Cortex-A Platforms:** Commonly used in ultra-low-power embedded systems. These are suitable for evaluating lightweight RL models and dynamic power management.
- **TI MSP430 and Arduino:** Ultra-low-power microcontrollers used for prototyping energy-saving techniques in constrained environments like sensor nodes or wearables.
- **FPGA-based Platforms:** Employed when custom RL hardware acceleration or power modeling is required, offering fine-grained control of energy consumption.
- **IoT Testbeds:** Platforms such as TOSSIM and FIT IoT-LAB offer large-scale, realistic testing environments for RL-based energy optimization in distributed sensor networks.

V. Literature Review

This literature Summary examines recent advancements in applying reinforcement learning to energy-efficient embedded systems, highlighting innovations in DVFS, learning rate optimization, deep learning integration, and object detection, while emphasizing adaptive policies, real-time performance, and reliability across diverse embedded computing environments.

Kumar and Sharma (2025) proposed energy-aware paradigm for the implementation of deep learning using CNN for real-time control in autonomous robotics for embedded systems. The overall power efficiency of the proposed system is achieved through efficient control of power consumption, latency, and runtime while has minor degradation in terms of accuracy through a number of approaches including model pruning, quantization, and adaptive CNN scaling. Studies show that power consumption has been cut down to as low as 47.3% thereby making the optimized system's power consumption in obstacle avoidance tasks as low as 2.8 W, the base system consuming up to 5.0 W power. The delay was also lowered to 47.1% in key tasks from 20 MS, thus ensuring more immediate decisions in real time activities [24].

Panda, Tripathy and Bhuyan (2024) presented an innovative solution by integrating proposed Reinforcement Learning (RL) algorithms into DVFS, addressing the limitations of conventional methods. The proposed RL algorithm employs Q-Learning, a model-free RL technique, to iteratively learn the optimal policy for adjusting CPU voltage and frequency. customized algorithm enables autonomous real-time adjustments of voltage and frequency levels, showcasing a remarkable 20% power saving compared to conventional DVFS. The model's adaptability is evident in its capacity to achieve optimal configurations across diverse workloads, emphasizing RL's potential for enhancing energy efficiency in computing systems [25].

Kaloev and Krastev (2023) detailed investigation and the development of guidelines for LR selection in RL. Use a variety of RL simulations to test the effectiveness of LR adaptation, each with a sophisticated 18-dimensional action space and a 128-dimensional input vector respectively. These simulations cover a variety of RL tasks, highlighting how important LR selection is in different situations. Two separate artificial neural networks (ANNs), one with 44,000 connections and the other with 27,000 hidden layer connections, may be used to provide insights regarding LR methods. They use a variety of customized LR values, from a first peak of 0.25 to a pitiful 0.000000025. test both the stability of training, where training episodes consistently obtain scores around the average, and the cumulative agent scores over several training episodes. Findings clarify LR tactics that maximize stability and performance in various RL contexts, providing academics and practitioners with insightful advice [26].

Lyu, Shen and Zhang (2022) innovations in reinforcement learning, such as deep reinforcement learning techniques and traditional reinforcement learning techniques. This article then examines the current state of advanced reinforcement learning studies, such as large-scale study of curiosity-driven learning, fuzzy theory-based deep reinforcement learning techniques, distributed deep reinforcement learning algorithms, and so on. Lastly, the difficulties that reinforcement learning faces are covered in this essay. In artificial intelligence, reinforcement learning is one of the most active study areas. In contrast to other machine learning techniques, reinforcement learning uses action mappings to learn from the environment. Therefore, by maximizing the environment's cumulative reward value, the selected course of action might create an ideal plan through trial and error [27].

Tan and Karaköse (2022) implemented a deep reinforcement learning method for object recognition on the PASCAL Voc2012 dataset using a neural network that constructed ourselves. method involves gradually advancing a bounding box in the direction of the objective to completely frame the item in the image. The developed neural network is composed of five layers. Additionally, the reward mechanism is optimized in order to maximize the map value. The option made about the incentive policy undoubtedly impact the result and be crucial to the agent's training. The outcome is improved since the ground truth and the bounding box intersect at the maximum rate because of the optimized reward function [28].

Yeganeh-Khaksar et al. (2021) said that increasing the voltage and frequency might produce a decrease in task dependability since it raises the fault rate and the jobs' worst-case execution duration. In this letter, they propose an improved DVFS method based on reinforcement learning to lower the power consumption of sporadic tasks at runtime in multicore embedded systems without task-reliability degradation, while also achieving power savings and maintaining task-reliability at an acceptable level. When making judgements, the reinforcement learner takes into account the power savings and task-reliability changes caused by DVFS. It also determines the appropriate voltage-frequency level for each task so that the timing restrictions are satisfied [29].

Table II summarizes recent studies on reinforcement learning approaches for energy-efficient embedded systems, highlighting methods, key findings, challenges addressed, and potential future research directions across diverse application domains.

TABLE II. COMPARATIVE ANALYSIS ON REINFORCEMENT LEARNING APPROACHES FOR ENERGY-EFFICIENT EMBEDDED SYSTEMS

Author	Study On	Approach	Key Findings	Challenges	Future Directions
Kumar and Sharma (2025)	Real-time control in autonomous robotics	CNN with model pruning, quantization, and adaptive scaling	Reduced power to 2.8W (47.3% savings); latency cut by 47.1%	Minor degradation in accuracy	Improve model compression techniques without sacrificing accuracy
Charan Bhuyan et al. (2024)	RL-integrated DVFS	Q-Learning (model-free RL)	20% power saving; adaptive to workload variations	Needs consistent training stability across devices	Extend RL-DVFS integration to GPU and heterogeneous systems
Kaloev and Krastev (2023)	LR selection in RL training	Varying LR values on ANNs (27k and 44k connections)	Optimized cumulative agent scores and training stability	LR sensitivity to task-specific variations	Design adaptive LR schedulers for diverse RL tasks
Lyu et al. (2022)	Advancements in RL methods	Distributed RL, Fuzzy-RL, Curiosity-driven learning	Surveyed multiple advanced RL algorithms	Generalization to real-world scenarios	Develop hybrid models combining RL with symbolic reasoning
Tan and Karaköse (2022)	Object detection using DRL	Custom 5-layer neural network with bounding box movement	Improved mAP by optimized reward design	Sensitive to reward function design	Apply to dynamic real-time video analytics
Yeganeh-Khaksar et al. (2021)	DVFS in multicore systems with task reliability	RL-enhanced DVFS maintaining timing constraints	Achieved power savings without degrading reliability	Managing fault rates under voltage scaling	Extend to mixed-criticality task systems and real-time schedulers

VI. CONCLUSION AND FUTURE WORK

The reinforcement learning (RL), when incorporated into an embedded system, provides a potential route to realize intelligent, energy-efficient, and adaptive computation in a resource-constrained setting. It has been shown that lightweight RL algorithms, integration of deep learning, and reward-based energy optimization methods are efficient in accelerating performance in real-time and minimizing energy usage. Prominent breakthroughs are RL-based dynamical voltage and frequency scaling (DVFS) and control policies to adapt to robotic systems and neural network compression strategies that combine efficiency and accuracy. Moreover, RL has been found promising in solving non-deterministic operational issues and achieving robustness in situation of varying load and environmental conditions. The use of simulation tools and embedded hardware platforms including Raspberry Pi, ARM Cortex, and FPGA-based systems have been the most crucial in the verification of these methodologies prior to their implementation. Nevertheless, there are outstanding issues that are related to on-device training, safe exploration on policy and real-time stability.

Future works are also needed to devise scalable RL models that offer online adaptation with minimal overhead, incorporate sophisticated neural architectures, and build domain-specific simulators to conduct strict testing. Further, reliability, interpretability and safety of learning based embedded control systems will be imperative in adoption to safety-critical and real-time applications.

REFERENCES

[1] S. Mittal, “A survey of techniques for improving energy efficiency in embedded computing systems,” *Int. J. Comput. Aided Eng. Technol.*, vol. 6, no. 4, p. 440, 2014, doi: 10.1504/IJCAET.2014.065419.

[2] U. Kulau, “Course: Energy efficiency in embedded systems - A system-level perspective for computer scientists,” *2018 12th Eur. Work. Microelectron. Educ. EWME 2018*, pp. 5–9, 2018, doi: 10.1109/EWME.2018.8629441.

[3] Z. Yu *et al.*, “Multi-Objective Optimization Approach Using Deep Reinforcement Learning for Energy Efficiency in Heterogeneous Computing System,” *Ieee Internet Things J.*, vol. 20, no. 10, pp. 1–11, 2023.

[4] P. Michailidis, I. Michailidis, and E. Kosmatopoulos, “Reinforcement Learning for Optimizing Renewable Energy Utilization in Buildings: A Review on Applications and Innovations,” *Energies*, vol. 18, no. 7, Mar. 2025, doi: 10.3390/en18071724.

[5] A. Nandy, P. Chaki, and O. P. Pandey, “A Study on Energy Consumption, Energy Saving and Effectiveness of Alternate Energy Sources in Domestic Sector of India,” *Int. J. Res. Eng. Technol.*, vol. 05, no. 02, pp. 183–187, Feb. 2016, doi: 10.15623/ijret.2016.0502031.

[6] B. F. Mon, A. Wasfi, M. Hayajneh, A. Slim, and N. Abu Ali, “Reinforcement Learning in Education: A Literature Review,” *Informatics*, vol. 10, no. 3, Sep. 2023, doi: 10.3390/informatics10030074.

[7] B. Prakash, M. Horton, N. R. Waytowich, W. D. Hairston, T. Oates, and T. Mohsenin, “On the use of Deep Autoencoders for Efficient Embedded Reinforcement Learning,” in *Proceedings of the 2019 Great Lakes Symposium on VLSI*, in GLSVLSI ’19. New York, NY, USA: ACM, May 2019, pp. 507–512. doi: 10.1145/3299874.3319493.

[8] Z. V Bundalo, “Energy efficient embedded systems and their application in wireless sensor networks,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1208, no. 1, Nov. 2021, doi: 10.1088/1757-899X/1208/1/012002.

[9] D. Valencia *et al.*, “Comparison of Model-Based and Model-Free Reinforcement Learning for Real-World Dexterous Robotic Manipulation Tasks,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, May 2023, pp. 871–878. doi: 10.1109/ICRA48891.2023.10160983.

[10] H. S. Chandu, “Efficient Machine Learning Approaches for Energy Optimization in Smart Grid Systems,” *IJSART*, vol. 10, no. 9, 2024.

[11] O. O. A, A. A, Y. N.A, and A. A. O, “Overview of Embedded System & Its Application,” *Int. Acad. Conf. Organ. by Acad. Staff Union Polytech. Iree Chapter*, vol. 1, no. June, pp. 1–7, 2022.

[12] Z. Eddaoudi, Z. Aarab, K. Boudmen, A. Elghazi, and M. D. Rahmani, “A Brief Review of Energy Consumption Forecasting Using Machine Learning Models,” *Procedia Comput. Sci.*, vol. 236, pp. 33–40, 2024, doi: 10.1016/j.procs.2024.05.001.

[13] H. S. Chandu, “Robust Control of Electrical Machines in Renewable Energy Systems: Challenges and Solutions,” *Int. J. Innov. Sci. Res. Technol.*, vol. 09, no. 10, pp. 594–602, Oct. 2024, doi: 10.38124/ijisrt/IJISRT24OCT654.

[14] D. Cao *et al.*, “Reinforcement Learning and Its Applications in Modern Power and Energy Systems: A Review,” *J. Mod. Power Syst. Clean Energy*, vol. 8, no. 6, pp. 1029–1042, 2020, doi: 10.35833/MPCE.2020.000552.

[15] N. Harki, A. Ahmed, and L. Haji, “CPU Scheduling Techniques: A Review on Novel Approaches Strategy and Performance Assessment,” *J. Appl. Sci. Technol. Trends*, vol. 1, no. 1, pp. 48–55, May 2020, doi: 10.38094/jastt1215.

[16] X. Li, T. Zhou, H. Wang, and M. Lin, “Energy-Efficient Computation with DVFS using Deep Reinforcement Learning for Multi-Task Systems in Edge Computing,” *IEEE Trans. Sustain. Comput.*, pp. 1–13, 2025, doi: 10.1109/TSUSC.2025.3593971.

[17] Priya, “Power Management in Wireless Sensor Network,” *Int. Res. J. Eng. Technol.*, vol. 04, no. 01, pp. 948–952, 2017.

[18] V. Panchal, “Energy-Efficient Core Design for Mobile Processors : Balancing Power and Performance,” *Int. Res. J. Eng. Technol.*, vol. 11, no. 12, pp. 191–201, 2024.

[19] R. Patel, “Advancements in Renewable Energy Utilization for Sustainable Cloud Data Centers : A Survey of Emerging Approaches,” *Int. J. Curr. Eng. Technol.*, vol. 13, no. 5, pp. 447–454, 2023.

[20] R. Patel, “Sustainability and Energy Management: Trends and Technologies for a Greener Industrial Future,” *Int. J. Adv. Res. Sci. Commun. Technol.*, vol. 4, no. 1, pp. 886–898, Jul. 2024, doi: 10.48175/IJARSCT-19200E.

[21] C. Vijai and M. S. R. Mariyappan, “Robotic Process Automation (RPA) in Human Resource Functions,” *Adv. Manag.*, vol. 16, no. 3, pp. 30–37, Aug. 2023, doi: 10.25303/1603aim030037.

[22] S. Corecco, G. Adorni, and L. M. Gambardella, “Proximal Policy Optimization-Based Reinforcement Learning and Hybrid Approaches to Explore the Cross Array Task Optimal Solution,” *Mach. Learn. Knowl. Extr.*, 2023, doi: 10.3390/make5040082.

[23] R. Jefrin and S. Rodchua, “Application of Robotics in Manufacturing Industry IndM 5230: Seminar in Industrial Management,” pp. 0–17, 2022.

[24] S. Kumar and P. Sharma, “Designing Energy-Efficient Embedded Systems with CNNs for Real-Time Control in Autonomous Robotics,” in *2025 International Conference on Automation and Computation (AUTOCOM)*, IEEE, Mar. 2025, pp. 218–223. doi: 10.1109/AUTOCOM64127.2025.10957507.

[25] P. Panda, A. Tripathy, and K. C. Bhuyan, “Reinforcement Learning-Based Dynamic Voltage and Frequency Scaling for Energy-Efficient Computing,” in *2024 Third International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE)*, IEEE, Apr. 2024, pp. 1–6. doi: 10.1109/ICDCECE60827.2024.10549241.

[26] M. Kaloev and G. Krastev, “Tailored Learning Rates for Reinforcement Learning: A Visual Exploration and Guideline Formulation,” in *2023 7th International Symposium on Innovative Approaches in Smart Technologies (ISAS)*, IEEE, Nov. 2023, pp. 1–7. doi: 10.1109/ISAS60782.2023.10391644.

[27] L. Lyu, Y. Shen, and S. Zhang, “The Advance of Reinforcement Learning and Deep Reinforcement Learning,” in *2022 IEEE International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA)*, IEEE, Feb. 2022, pp. 644–648. doi: 10.1109/EEBDA53927.2022.9744760.

[28] Z. Tan and M. Karakose, “Optimized Reward Function Based Deep Reinforcement Learning Approach for Object Detection Applications,” in *2022 International Conference on Decision Aid Sciences and Applications (DASA)*, IEEE, Mar. 2022, pp. 1367–1370. doi: 10.1109/DASA54658.2022.9764979.

[29] A. Yeganeh-Khaksar, M. Ansari, S. Safari, S. Yari-Karin, and A. Ejlali, “Ring-DVFS: Reliability-Aware Reinforcement Learning-Based DVFS for Real-Time Embedded Systems,” *IEEE Embed. Syst. Lett.*, vol. 13, no. 3, pp. 146–149, Sep. 2021, doi: 10.1109/LES.2020.3033187.

