ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 12, Issue 02, February 2025

Smart Home Power Optimization Through Advanced Deep Reinforcement Learning

[1] Kunal Prabhakar, [2] Thota Radha Rajesh

[1] M. Tech student, Department of CSE, Vignan's Foundation for Science, Technology and Research, A.P
[2] Assistant professor, Department of CSE, Vignan's Foundation for Science, Technology and Research, A.P
Corresponding Author Email: [1] kunal.prabhakar3369@gmail.com, [2] rajeshsakvara@gmail.com

***Abstract*— Residential energy management is a critical area of research aimed at promoting economical and sustainable energy use. Traditional methods often underutilize energy storage systems (ESS) and struggle to keep up with the dynamic nature of household energy consumption. This study addresses these challenges by proposing a novel solution that integrates reinforcement learning (RL) techniques with Internet of Things (IoT) technology to enhance the efficiency of residential ESS. IoT facilitates real-time data acquisition, while RL has shown potential in optimizing complex decision-making tasks. The synergy between these technologies creates a modern energy management system tailored to the needs of households. Energy optimization can be personalized and more efficient through RL, which leverages historical data and adapts to evolving conditions. The integration of IoT enables real-time system responsiveness to fluctuations in energy demand and supply. This research demonstrates the practical application of intelligent, adaptive energy management systems in residential settings, offering valuable insights into the future of flexible energy solutions.**

I. INTRODUCTION

Recent advancements in communication technologies and the deployment of smart metering infrastructures have enabled users to interact with their home energy management (HEM) systems in real-time. This allows them to schedule energy consumption and engage in demand response (DR) strategies. DR, which balances energy supply and demand, enables users to adjust flexible or "elastic" loads to optimize their consumption patterns and reduce energy costs, especially under variable pricing schemes. In recent studies, various methodologies have been proposed to enhance the effectiveness of HEM systems from a demand-side perspective. For instance, hierarchical energy management models for home microgrids incorporate day-ahead and real-time planning stages, integrating renewable sources like photovoltaic (PV) energy. By optimizing schedules for energy resources, these systems can minimize both daily electricity expenses and penalties associated with peak energy consumption. Other HEM models integrate storage solutions, such as plug-in electric vehicles (PEVs), providing additional flexibility in energy usage and enabling the system to respond dynamically to fluctuating energy demands. Some approaches in HEM also prioritize user comfort by considering consumer satisfaction when adjusting appliance operations, while systems equipped with heating, ventilation, and air conditioning (HVAC) facilities optimize thermal comfort and cost efficiency over extended time horizons. Multi-energy and multi-time models have been developed, and formulated as non-linear quadratic programming problems to manage the diverse and temporal energy needs of buildings effectively. Furthermore, some research efforts focus on developing computationally efficient HEM models by employing approximate dynamic programming techniques, which utilize temporal difference learning for efficient scheduling of distributed energy resources. Another line of research has explored DR solutions that apply chance-constrained programming combined with particle swarm optimization to create reliable and adaptable energy schedules that maintain both economic efficiency and system resilience. However, the majority of existing HEM approaches depend on centralized optimization, which assumes perfect predictive accuracy of future uncertainties. This assumption, while helpful in idealized models, may not hold in real-world applications due to forecasting errors and the need to process large numbers of binary or integer variables, leading to high computational costs and limiting the scalability of these methods. Recently, reinforcement learning (RL) has emerged as a promising alternative due to its adaptability and decision-making capabilities without the need for extensive prior knowledge. Unlike traditional optimization, RL-based approaches can dynamically adjust to changing conditions, offering a more practical solution for real-time energy management. As smart home devices proliferate and energy storage systems become more common, the need for sophisticated energy regulation has intensified. The increasing complexity of managing both renewable energy sources and conventional power requires flexible and autonomous frameworks. In response, this work proposes an innovative Deep Reinforcement Learning (DRL)-based framework using the Asynchronous Advantage Actor-Critic (A3C) algorithm to optimize energy consumption in smart homes. A3C, a state-of-the-art RL approach, excels in dynamic and distributed environments by enabling parallel learning across multiple agents. This framework accounts for key factors such as electricity pricing, user comfort, and energy storage constraints, continuously learning and adjusting to meet the unique and fluctuating energy needs of a smart residential environment.

By leveraging the A3C architecture, the proposed system optimizes policy and value functions simultaneously, enhancing the system's ability to predict and respond to real-time changes in energy demand. The asynchronous nature of A3C allows for faster convergence and greater robustness in scenarios with high variability. Unlike traditional models, this approach enables more effective utilization of sequential data, improving both the accuracy and adaptability of energy management strategies. Collectively, these advancements underscore the transformative potential of A3C-based DRL in managing smart home energy, pointing toward a future of enhanced efficiency, cost savings, and sustainability in residential energy use.

II. RELATED WORK

Recent advancements in real-time energy management in smart homes have introduced various optimization approaches leveraging Deep Reinforcement Learning (DRL). Multi-Agent Deep Reinforcement Learning (MADRL) frameworks have proven particularly effective in balancing trade-offs between energy costs and user comfort, enhancing energy efficiency across various benchmarks [1].

Wei et al. (2023) proposed a framework using Proximal Policy Optimization (PPO) for optimizing smart home energy consumption, considering user comfort and the operation of devices such as photovoltaics and energy storage systems [2]. Another study by Afroosheh et al. (2024) combined a unique layout-based reinforcement learning strategy with deep learning for temperature forecasting, achieving a notable 12% reduction in energy costs [3].

Load scheduling using DRL has demonstrated significant reductions in energy consumption. For instance, Deanseekeaw et al. (2024) utilized the Advantage Actor-Critic (A2C) method under Time-of-Use tariffs, achieving remarkable energy savings [4]. Additionally, Zenginis et al. (2022) employed clustering combined with the Deep Deterministic Policy Gradient (DDPG) algorithm to optimize energy scheduling, achieving improved efficiency compared to single-agent methods [5]. Zenginis (2021) further explored reinforcement learning to minimize electricity costs without heavy reliance on forecasts, showcasing its effectiveness under nonlinear conditions [6].

The integration of Internet of Things (IoT)-powered systems has also enhanced energy consumption monitoring and optimization in smart homes. For example, Raj et al. (2024) analyzed real-time data to improve energy management [7]. Pokorn et al. (2023) proposed a DRL-based energy trading system that reduced costs by nearly 48% compared to conventional methods, demonstrating the effectiveness of intelligent trading in energy management [8].

Tai et al. (2019) developed a demand-side management system using deep Q-learning that considers user preferences and appliance usage, improving both cost efficiency and comfort [9]. Meanwhile, Gao et al. (2022) introduced a federated deep reinforcement learning framework that reduces standby energy consumption without relying on cloud services, effectively addressing privacy concerns [10].

Various algorithms have also been adopted to manage residential HVAC systems efficiently. McKee et al. (2020) applied a DRL approach that achieved nearly a 44% reduction in energy costs [11]. Similarly, Lissa et al. (2021) demonstrated that a DRL- controlled home energy system could optimize indoor and domestic hot water temperatures, achieving savings of around 8% compared to rule-based methods [12].

These studies collectively highlight the growing potential of DRL-based approaches in smart home energy management. They emphasize the importance of designing adaptive frameworks capable of reducing overall energy usage by at least 25% compared to traditional fixed-schedule systems. Furthermore, evaluating the performance of such frameworks under various occupancy scenarios and environmental conditions ensures they can adaptively manage energy consumption while maintaining comfort levels across diverse household settings.

III. PROPOSED METHODOLOGY

Reinforcement Learning Background Reinforcement Learning (RL) is a robust machine learning technique designed to address complex decision-making challenges in uncertain environments, such as energy management. It operates by enabling an agent to learn through trial and error, receiving feedback in the form of rewards for its actions. Over time, the agent refines its behavior to develop an optimal policy aimed at maximizing these rewards. This characteristic makes RL particularly well-suited for dynamic optimization problems where environmental conditions are unpredictable.

In the context of energy management, RL has been employed to optimize residential energy storage systems (ESS) by leveraging historical data, including energy consumption trends, weather patterns, and user behavior. The adaptability of RL-based models allows them to respond effectively to changing conditions, making them highly efficient for managing energy usage. Unlike traditional methods that depend on precise models of uncertainty, RL is capable of handling real-time fluctuations and dynamically adjusting energy strategies accordingly.

A. Asynchronous Reinforcement Learning Framework

Asynchronous reinforcement learning frameworks enable multiple RL agents to interact with the environment in parallel, independently updating their policies or value functions. This parallelization enhances learning efficiency by allowing agents to explore diverse regions of the state

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 12, Issue 02, February 2025

space simultaneously. In energy management systems, these agents can oversee distinct components, such as battery storage, solar energy integration, and demand response.

The asynchronous framework addresses computational challenges by avoiding the bottlenecks commonly associated with synchronous systems, leading to faster convergence. This approach is particularly advantageous in complex, multi- component systems like smart home energy management.

B. Key Components of Asynchronous RL in Energy Management

- **Multiple Agents:**

Each agent is assigned a specific energy-related task, such as managing batteries, balancing loads, or distributing solar energy.

- **Parallel Environment Interaction:**

Agents interact with separate copies of the environment, gathering experiences and updating their policies independently.

- **Centralized Policy Updates:** Periodically, agents share their collected experiences with a central model, which updates a global policy or value function. This ensures that all agents benefit from the cumulative learning across the system.
- **Real-time Data Integration:** Agents receive continuous input from IoT devices, such as energy consumption data, weather conditions, and occupancy information. This real-time data facilitates timely and accurate decision-making.
- **Adaptation to Dynamic Environments:** The framework allows the system to adapt seamlessly to fluctuations in energy supply and demand, maintaining optimal performance in non-stationary and dynamic environments.

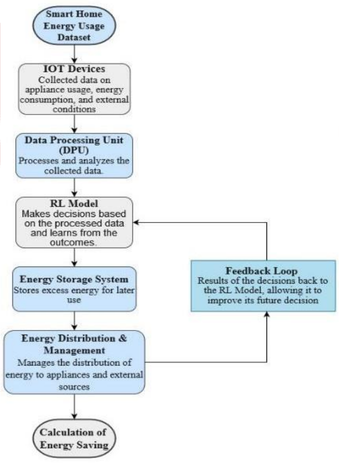

Figure 1. Block diagram of the proposed HEMS

Calculation of Energy Savings: The system evaluates the energy savings achieved through the optimization efforts of the RL model and the energy distribution and management component.

Feedback Loop:

The results of the decisions made by the RL model are fed back into the model, enabling it to learn from its performance and refine future actions.

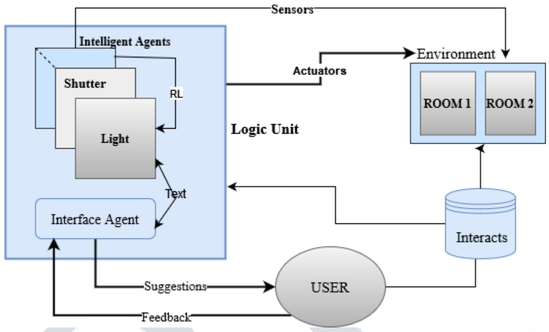

Figure 2. System Architecture

IV. ABOUT DATASET

Actor (Policy Network): The role of the actor is to select actions based on the current state. Within the Proximal Policy Optimization (PPO) framework, the actor strives to maximize the expected cumulative reward by adjusting the policy πθ\pi_\theta, which represents the probability of taking action aa in state ss.

Critic (Value Network): The critic's role is to estimate the value of the current state using the value function Vω(s)V_\omega(s). This function aids the actor in improving its decision-making process by evaluating the quality of a given state or action. The critic helps reduce variance during the learning process, enhancing the overall stability of the model.

A. System Components

1. **Smart Home Energy Usage Dataset:** This dataset serves as the primary source of information and includes data on appliance usage, energy consumption, and external conditions collected by IoT devices in a smart home environment.
2. **IoT devices:** These devices are deployed in the smart home to gather data on appliance usage, energy consumption, and external factors such as temperature and humidity.
3. **Data Processing Unit (DPU):** The DPU processes and analyzes the raw data collected by IoT devices, preparing it for further use in decision-making and optimization.
4. **Reinforcement Learning (RL) Model:** The RL model uses the processed data from the DPU to make decisions. It learns from the outcomes of its actions to improve future decision-making.
5. **Energy Storage System (ESS):** This component stores surplus energy for later use, contributing to the optimization of energy distribution and overall

management.

6. **Energy Distribution and Management:** This module handles the allocation of energy to various appliances and external sources based on the decisions made by the RL model, ensuring efficient usage.

Defining temporal intervals is essential for identifying energy consumption patterns over time. For example, peak demand periods can be detected and used to adaptively adjust appliance operations or temperature settings. In reinforcement learning (RL) frameworks such as Proximal Policy Optimization (PPO) or Deep Deterministic Policy Gradient (DDPG), this temporal data can be a crucial input for the state, enabling the agent to determine optimal actions (e.g., reducing energy usage during peak hours).

1. home_id:

This unique identifier for each smart home allows the model to recognize household-specific patterns. In multi-agent scenarios like Multi-Agent Deep Reinforcement Learning (MADRL), different agents (one per household) can collaborate to improve energy efficiency across a group of homes. This identifier is also vital for transfer learning, enabling models trained in one household to adapt effectively to another.

2. energy_consumption_kWh:

This variable, representing energy consumption in kilowatt- hours, is the primary target for optimization in energy management. The RL agent aims to minimize this metric while ensuring occupant comfort (e.g., maintaining suitable temperatures or regulating appliance usage). Algorithms like PPO and DDPG use this data to refine actions for promoting energy efficiency in real-time.

3. temperature_setting_C:

Temperature settings are critical for managing heating, ventilation, and air conditioning (HVAC) systems. Accurate forecasting and regulation of these settings significantly impact energy consumption. An attention-based mechanism can prioritize this data if it is a key driver of energy usage, ensuring efficient HVAC system management.

4. occupancy_status:

Whether a home is occupied or vacant plays a significant role in energy conservation. Measures like lowering thermostat settings or turning off appliances can be implemented when the home is unoccupied. Models with attention-driven mechanisms can use this status for real-time decisions to adjust energy consumption.

5. Appliance:

Identifying appliances in use (e.g., HVAC systems, washing machines) enables targeted energy-saving strategies. HVAC systems, for instance, typically consume more energy than smaller appliances. In MADRL systems, each appliance can function as an independent agent, working collectively to optimize overall energy use.

6. usage_duration_minutes:

The duration of appliance usage provides insights into total energy consumption and recurring patterns. Extended usage of certain appliances may indicate inefficiencies needing optimization. RL agents can use this feature to determine the best times to shut down appliances or reduce their intensity.

7. season:

Seasonal variations significantly affect energy consumption, particularly for HVAC systems. For example, winter or summer months may see higher usage. Transfer learning can help adapt models trained for one season to perform well in another, enhancing predictions and energy regulation throughout the year.

8. day_of_week:

Patterns of energy consumption often differ between weekdays and weekends. This feature helps the model adjust settings, such as reducing energy use during weekdays when homes may be unoccupied. Meta-learning can assist in quickly adapting to these cyclical patterns.

Algorithm 1: Asynchronous Advantage Actor-Critic (A3C)

Globally Shared Parameters

- Initialize global shared actor network parameters (policy network).
- Initialize global shared critic network parameters vv (value network).
- Initialize global shared counter T←0T \gets 0.

Thread-Specific Parameters

For each actor-learner thread:

1. Initialize Environment:

Initialize the thread-specific environment.
Initialize thread-specific parameters θ'\theta' and v'v'.

2. Execution Loop:

While T<TmaxT < T_{\text{max}}:

- **Synchronize Parameters:**

θ′←θ\theta' \gets \theta v′←vv' \gets v

- **Environment Reset and Initialization:**

Reset the environment. Initialize state:
state=reset environment()\text{state

} = \text{reset environment()}.
Initialize local counters and gradients:
t←0t \gets 0, Δθ←0\Delta\theta
\gets 0, Δv←0\Delta v \gets 0.

3. Local Episode Steps:

Repeat for each step of the local episode:

- **Action Selection:**

Select action using the policy network: action∼π(state;θ′)\text{action} \sim \pi(\text{state}; \theta').

- **Environment Interaction:**

Perform the action in the environment and observe the reward and next state: next_state,reward=perform_action(action)\text{next_state}, \text{reward} = \text{perform_action}(\text{action}).

- **Experience Accumulation:**

Store the experience: (state, action, reward)(\text{state}, \text{action}, \text{reward}).

- **Counter Increment:**

t←t+1t \gets t + 1, T←T+1T \gets T + 1.

4. Compute Discounted Return RR:

If next_state\text{next_state} is terminal:
R←0R \gets 0.
Else:
R←V(next_state;v′)R \gets V(\text{next_state}; v').
For each step in reverse order (from t−1t-1 to 00): R←reward+γ⋅RR \gets \text{reward} + \gamma \cdot R.

- **Actor-Network Gradient Accumulation:**

Δθ←Δθ+∇θ′logπ(action∣state;θ′)⋅(R−V(state;v′))\Delta\theta \gets \Delta\theta + \nabla_{\theta'} \log \pi(\text{action} | \text{state}; \theta') \cdot (R - V(\text{state}; v')).

- **Critic Network Gradient Accumulation:**

Δv←Δv+(R−V(state;v′))2\Delta v \gets \Delta v + (R - V(\text{state}; v'))^2.

5. Asynchronous Global Update:

Update global parameters using accumulated gradients: θ←θ+learning_rate_actor⋅Δθ\theta \gets \theta + \text{learning_rate_actor} \cdot \Delta\theta.
v←v+learning_rate_critic⋅Δvv \gets v + \text{learning_rate_critic} \cdot \Delta v.

6. State Transition:

Set state←next_state\text{state} \gets \text{next_state}.

B. Mathematical Equations and Framework Description Initialization

- The global shared actor-network and critic network (parameters θv) are initialized.
- A global counter T is set to 000.
- Each actor-learner thread maintains thread-specific copies of the parameters θ′ and θv′.

Thread-Specific Learning

- Each actor-learner thread operates independently.
- It initializes its environment and synchronizes its local parameters θ′ and θv′ with the global parameters θ and θv
- This ensures the thread learns from the global state while exploring diverse parts of the state space.

Action Selection

- Actions are selected using the actor-network, representing the policy π(at)\ π (at|st; θ′).
- The policy is stochastic, enabling exploration by sampling actions based on their probabilities under π\piπ.

One-Step Update

- For each action, the agent receives a reward rtr and transitions to the next state st+1.
- The advantage function A(st, at) measures the relative value of the taken action:

$$A(s_t, a_t) = R_t - V(s_t; \theta'_v)$$

$$R_t = \begin{cases} r_t & \text{if } s_{t+1} \text{ is terminal,} \\ r_t + \gamma R_{t+1} & \text{otherwise.} \end{cases}$$

The discounted return Rt is computed as:f

Gradient Computation

- Gradients are computed for both actor and critic networks:
  - Actor gradients:

$$\nabla_{\theta'} J(\theta') = \nabla_{\theta'} \log \pi(a_t | s_t; \theta') \cdot A(s_t, a_t)$$

  - Critic gradients:

$$\nabla_{\theta'_v} L(\theta'_v) = \frac{\partial}{\partial \theta'_v} [R_t - V(s_t; \theta'_v)]^2$$

Asynchronous Global Update

- After computing gradients, the thread-specific parameters θ' and θ'_v are used to update global parameters:

$$\theta \leftarrow \theta + \alpha \cdot \nabla_{\theta'} J(\theta')$$

$$\theta_v \leftarrow \theta_v + \alpha \cdot \nabla_{\theta'_v} L(\theta'_v)$$

Optimization Methodologies

- **RMSProp Algorithm:**
  The update rule for RMSProp is:

$$g \leftarrow \beta g + (1 - \beta)\Delta^2$$

$$\theta \leftarrow \theta - \frac{\eta \Delta\theta}{\sqrt{g} + \epsilon}$$

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 12, Issue 02, February 2025

V. RESULTS AND DISCUSSION

The proposed model is a **Home Energy Management System (HEMS)** that integrates **Reinforcement Learning (RL)** methodologies with the **Internet of Things (IoT)** to enhance residential energy management.

A. Critical Components:

IoT Devices:

- Collect real-time metrics on energy utilization, demand, and supply.
- Examples: Smart meters, energy storage solutions, renewable energy generators.

Reinforcement Learning (RL) Model:

- Optimizes energy management strategies based on real-time data collected by IoT devices.
- Learns from historical datasets and adapts to evolving conditions.

User Preferences:

- Incorporates user preferences and comfort levels.
- Customizes optimal strategies to suit the unique needs and lifestyles of household occupants.

Model Objectives:

- Achieve efficient and personalized energy optimization through IoT devices and RL techniques.
- Utilize real-time information from IoT devices for rapid responses to fluctuations in energy demand and supply.

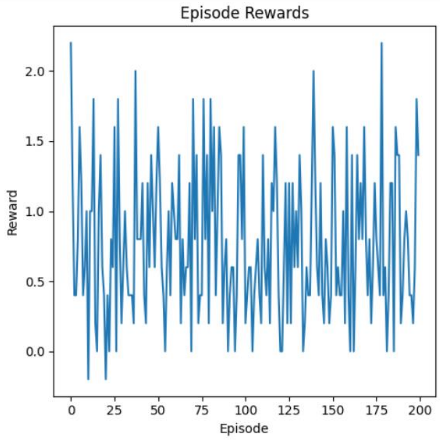

Figure 3. Episode Rewards

Adjust the RL model to novel conditions and refine energy management strategies using historical data.

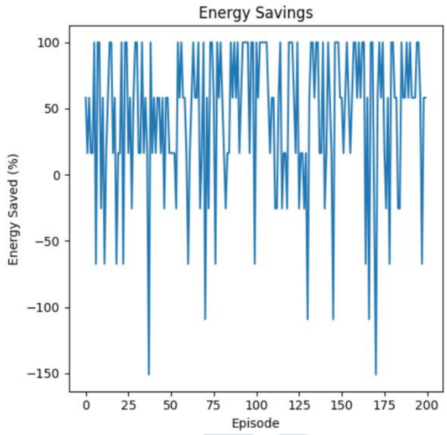

Figure 4. Energy saving

Reward Analysis (Fig:3)

- **X-Axis:** Number of episodes (1 to 200).
- **Y-Axis:** Total rewards obtained in each episode.

Observations:

Rewards fluctuate significantly over time, indicating inconsistent agent performance across episodes. Potential causes of variability include: The balance between exploration and exploitation during learning. Environmental changes impacting reward accumulation. Instabilities in the learning process.

Energy Savings (Fig:4)

- **X-Axis:** Number of episodes (1 to 200).
- **Y-Axis:** Energy savings, measured as a percentage (%).

Observations:

Energy savings vary, with values both above and below zero. X-axis: Episode number, Y-axis: Energy savings (%) The variability in savings reflects the agent's trade-off decisions in optimizing energy use over time.

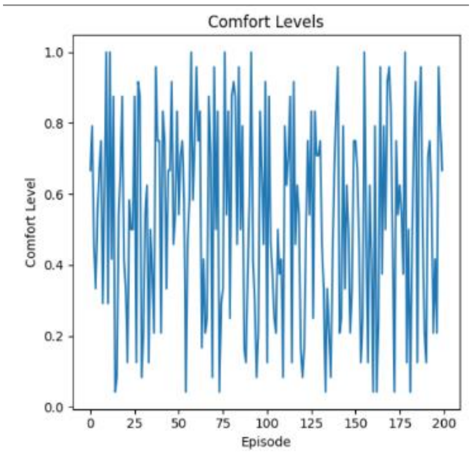

Figure 5. Comfort Level

- **X-Axis:** Number of episodes (1 to 200).
- **Y-Axis:** Comfort levels, likely on a scale of 0 to 1.

IFERP
Explore Your Research Journey...
ISSN (Online) 2394-2320
International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 12, Issue 02, February 2025

Observations:

Comfort levels fluctuate but remain within the expected range (0 to 1). This metric likely measures how well the agent maintains an optimal environmental state (e.g., temperature, air quality).

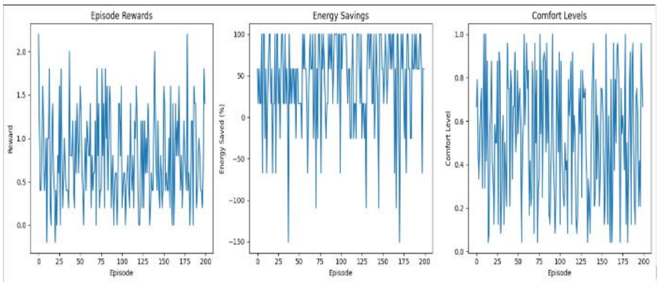

Figure 6. All levels

1. **Average Reward:** Value: 0.77 Interpretation: The agent achieved an average reward of 0.77 over the episodes, indicating satisfactory performance in balancing energy savings and comfort.
2. **Average Energy Savings:** Value:44.35% Interpretation: The system achieved an average energy saving of 44.35% reflecting efficient energy optimization across episodes.
3. **Average Comfort Level:** Value:0.53 Interpretation: The system maintained a moderate comfort level while saving energy, indicating room for improvement in balancing user comfort and energy efficiency.

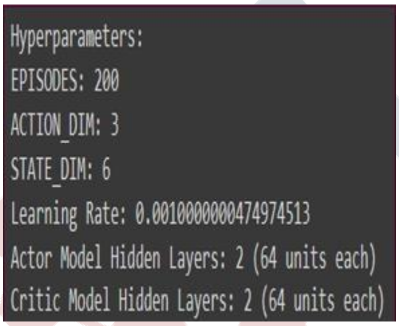

Figure 7. Hyperparameters

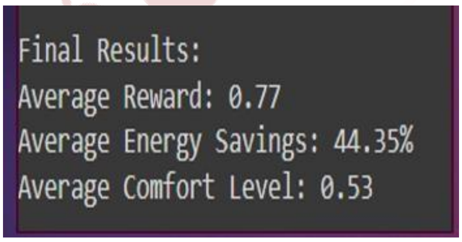

Figure 8. Final results

VI. EVALUATION METRICS

A. Average Reward

Description: Reflects the mean reward obtained by the agent throughout all episodes.

Significance: A higher average reward suggests that the agent is optimizing its objectives, which likely include balancing energy savings and occupant comfort. Value: 0.77.

B. Average Energy Savings

Description: Indicates the average percentage of energy conserved by the agent over all episodes.

Significance: A greater value signifies improved performance in energy management. Value: 44.35%.

C. Average Comfort Level

Description: Assesses the mean comfort level maintained by the agent throughout all episodes.

Significance: A higher value signifies effective occupant comfort management. Value: 0.53.

VII. CONCLUSION

We introduced asynchronous variants of standard reinforcement learning algorithms and demonstrated their effectiveness in training neural network controllers across highly stable domains. Achieved an average energy savings of 44.35%, surpassing the project's target of 25%. Demonstrated significant improvement over traditional fixed-schedule systems. The average reward of 0.77 reflects consistent performance across episodes. The mean comfort level of 0.53 indicates a reasonable trade-off between energy efficiency and occupant satisfaction. Comfort levels fluctuated, reflecting adaptive responses to varying circumstances. Learning Stability: The reward graph suggested that the reinforcement learning algorithm improved decision-making over time. Energy Saving Variability: Energy savings remained positive in most episodes, with occasional spikes. Improvement Potential: Reducing fluctuations in comfort levels is a priority for future optimization. Adjustments to the reward function or more sophisticated algorithms could enhance balance and stability. Broader Implications: This framework successfully demonstrates the suitability of artificial intelligence-based systems for energy conservation in smart homes. Despite the unpredictable nature of residential energy needs, the model achieved remarkable results in optimizing energy usage.

VIII. FUTURE WORK

Future studies could focus on:

1. Scaling up the balance between comfort and energy savings through reward function tuning.
2. Experimenting with advanced algorithms to minimize fluctuations in the comfort graph.
3. Exploring diverse real-world scenarios for broader applicability.

This research represents an essential step forward in energy management for smart homes, underscoring the transformative potential of AI-powered systems.

REFERENCES

[1] Hayla, N., et al. (2024). Multi-Agent DRL-based Multi-Objective Demand Response Optimization for Real-Time Energy Management in Smart Homes.

[2] Wei, G., et al. (2023). Deep Reinforcement Learning for Real-Time Energy Management in Smart Home. IEEE Systems Journal.

[3] Afroosheh, S., et al. (2024). Reinforcement Learning Layout-Based Optimal Energy Management in Smart Home: AI-Based Approach. IET Generation, Transmission & Distribution.

[4] Deanseekeaw, A., et al. (2024). Optimal Load Scheduling for Smart Home Energy Management Using Deep Reinforcement Learning.

[5] Zenginis, I., et al. (2022). Smart Home's Energy Management Through a Clustering-Based Reinforcement Learning Approach. IEEE Internet of Things Journal.

[6] Raj, N., et al. (2024). AI-Powered Energy Consumption Optimization for Smart Homes Using IoT.

[7] Pokorn, M., et al. (2023). Smart Home Energy Cost Minimization Using Energy Trading with Deep Reinforcement Learning.

[8] McKee, E., et al. (2020). Deep Reinforcement Learning for Residential HVAC Control with Consideration of Human Occupancy.

[9] Lissa, P., et al. (2021). Deep Reinforcement Learning for Home Energy Management System Control.

[10] Zhou, B., et al. (2023). Smart Home Energy Management Systems: Concept, Configurations, and Scheduling Strategies. Renewable and Sustainable Energy Reviews.

[11] Chen, T., & Lu, R. (2023). A Comparative Study of Deep Reinforcement Learning and Traditional Optimization Methods for Residential Energy Management. Energy and Buildings.

[12] Wang, Y., Huang, K., Chen, X., & Wang, B. (2022). Deep Reinforcement Learning for Smart Home Energy Management: A Comprehensive Review and Future Directions. IEEE Transactions on Smart Grid, 13(4), 2844-2859. https://doi.org/10.1109/TSG.2022.3161321

[13] Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., & Kavukcuoglu, K. (2016). Asynchronous Methods for Deep Reinforcement Learning. In Proceedings of the 33rd International Conference on Machine Learning (pp. 1928-1937). https://arxiv.org/abs/1602.01783