FPT UNIVERSITY

CAPSTONE PROJECT DOCUMENT

Application of Deep Reinforcement Learning Algorithm for Co-optimizing Energy, and Thermal Comfort in Office Building under Vietnamese Climate Context

AI04	
Group Members	Ta Khoi Nguyen – Leader – DE170642 Nguyen Van Hon – Member – DE170592 Nguyen Ha Phuong – Member – HE171166 Doan Van Quoc Hoan – Member – DE170533
Supervisor	Dr. Nguyen Gia Tri
Capstone Project Code	AIP491

December 2025

Contents

Acknowledgement

We would like to extend our profound gratitude to all individuals and organizations whose support and contributions were essential to the successful completion of this capstone project.

First, we sincerely thank the Faculty of the Artificial Intelligence Department at FPT University. Their commitment to academic excellence, thoughtful guidance, and constant encouragement laid the groundwork for this project. The knowledge, methodologies, and analytical skills gained through their instruction have significantly shaped our direction and understanding throughout the entire process.

We are equally grateful to our project supervisors and advisors for their insightful feedback, technical expertise, and dedicated mentorship. Their experience and constructive perspectives helped us navigate challenges, refine our ideas, and ensure that our work met both academic and industry expectations.

Our heartfelt appreciation also goes to our team members for their enthusiasm, resilience, and collaborative spirit. Every member's dedication, creativity, and shared responsibility brought this project from concept to completion. Working together through obstacles and solutions has made this journey both intellectually enriching and personally meaningful.

We further acknowledge the valuable role of external platforms, tools, and open-source technologies that supported our research, development, and evaluation. Open datasets, software frameworks, cloud services, and online technical communities were indispensable in transforming our ideas into a functional system.

Lastly, we appreciate the academic community, industry professionals, and peers whose work, insights, and collaboration have indirectly expanded the depth and quality of our project.

This achievement reflects the collective efforts of many, and we are truly thankful to everyone who, directly or indirectly, contributed to the success of this endeavor.

Definition and Acronyms

Acronym	Definition
AHU	Air Handling Unit
IAQ	Indoor Air Quality
DDPG	Deep Deterministic Policy Gradient
DQN	Deep Q-Network
DRL	Deep Reinforcement Learning
HVAC	Heating, Ventilation, and Air Conditioning
RBC	Rule-based control
MPC	Model Predictive Control
BMS	Building Management Systems
BAS	Building Automation Systems
TCN	Temporal Convolutional Networks
EPW	EnergyPlus Weather file
PM	Project Manager
PMP	Project Management Plan
PPO	Proximal Policy Optimization
RL	Reinforcement Learning
TD3	Twin Delayed Deep Deterministic Policy Gradient
WBS	Work Breakdown Structure

List of Tables

List of Figures

I. Project Introduction

1. Overview

1.1 Project Information

This project, referred to as RL-HVAC, explores how Deep Reinforcement Learning can be aimed to minimize energy consumption while maintaining indoor environmental quality, including both thermal comfort and air quality within office buildings exposed to the unique climatic conditions of Vietnam.

Full Title: Application of Deep Reinforcement Learning for Co-optimizing Energy and Thermal Comfort in Office Buildings under Vietnamese Climate Context.

Vietnamese Title: Ứng dụng thuật toán Học Tăng Cường Sâu để tối ưu đồng thời năng lượng và tiện nghi nhiệt/ không khí cho tòa nhà văn phòng trong bối cảnh khí hậu Việt Nam.

Duration: 15 weeks, 7 sprints

Start Date: September 8th, 2025

End Date: December 17th, 2025

The following is the list of our supervisor and team members dedicated to this project:

Table 1: Supervisor

Full Name	Role	Email	Title
Nguyen Gia Tri	Mentor	tring2@fe.edu.vn	PhD

Table 2: Team Members

Full Name	Role	Email
Ta Khoi Nguyen	Leader	nguyentkde170642@fpt.edu.vn
Nguyen Van Hon	Member	honnvde170592@fpt.edu.vn
Nguyen Ha Phuong	Member	phuongnhhe171166@fpt.edu.vn
Doan Van Quoc Hoan	Member	hoandvqde170533@fpt.edu.vn

1.2 Project Overview

This study investigates the optimization of Heating, Ventilation, and Air Conditioning (HVAC) systems in office buildings in Vietnam, aiming to minimize energy consumption while maintaining indoor environmental quality, including both thermal comfort and air quality. The work explores a Deep Reinforcement Learning approach that enables the controller to learn adaptive policies responsive to weather variability, time-of-day patterns, and occupancy levels. The research is contextualized for Vietnam, where a humid tropical climate and distinct seasonal variations require control strategies that can generalize effectively across diverse environmental conditions.

The proposed framework optimizes HVAC systems in office buildings under Vietnamese climate through three innovative aspects:

1. **Forecast-aware decision-making:** Introduces a forecast-aware control mechanism that fuses short-term outdoor temperature prediction with DRL, enabling proactive behaviors such as adaptive precooling and load shifting where an integration rarely operationalized in current HVAC control research.

2. **Multi-objective optimization:**minimize energy costs while maintaining indoor environmental quality, including both thermal comfort and air quality(CO_2) . and time-varying electricity tariffs through a unified, dynamically weighted reward structure tailored to Vietnam's climate conditions.

3. **Improvement Over Baseline Methods:** Establishes a predictive, data-driven control paradigm that moves beyond rule-based and fixed-setpoint HVAC strategies (RBC & MPC) by enabling continuous policy adaptation to changing weather, occupancy patterns, and operational variability.

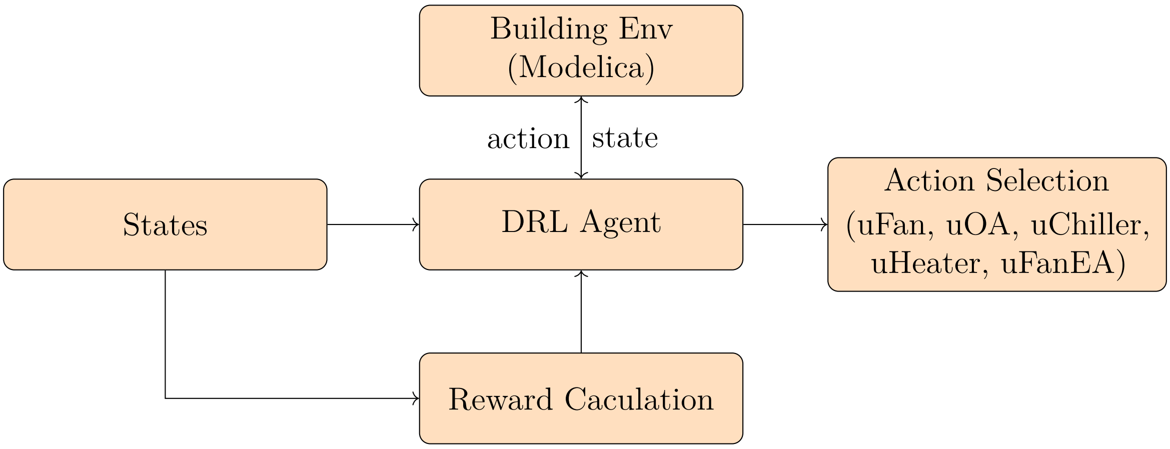


Figure 1: Deep Reinforcement Learning Framework for HVAC Control

In this architecture, we first execute the FMU-based simulation in Modelica to generate the required time-series data including temperature, humidity, CO_2, and thermal loads. These environmental variables form the state input to the reinforcement learning agent. Upon receiving this state, the agent evaluates current conditions and selects an HVAC control action such as adjusting temperature setpoints, modifying airflow rates, or increasing/decreasing fan speed. The selected actions are then applied to the simulated building environment, where indoor conditions evolve according to the underlying thermal dynamics. From the resulting changes in energy consumption, thermal comfort, and indoor air quality, a reward signal is computed and returned to the agent. Through this iterative interaction loop, the agent progressively learns an optimal control policy that balances energy efficiency and thermal comfort.

2. Project Background

2.1 Problem Context

Vietnam's rapid urbanization has significantly increased the energy demands of commercial buildings, positioning HVAC systems at the forefront of operational challenges and sustainability pressures. Consequently, modern buildings must confront several critical issues, including:

- Commercial buildings have expanded quickly, becoming major energy consumers in urban areas.
- HVAC systems alone account for 40–60% [28] of total building energy use, representing the largest operational load.
- Vietnam's hot and humid tropical climate, coupled with the distinct environmental characteristics of Ha Noi, Da Nang, and Ho Chi Minh City, makes cooling and dehumidification indispensable for comfort and productivity.
- These systems, however, are highly inefficient and contribute substantially to operational costs and carbon emissions.
- Traditional HVAC control strategies (RBC, MPC) remain reactive, rely on simplified assumptions and fixed set-points, and do not adapt to dynamic building conditions.

2.2 Algorithms

Reinforcement Learning (RL) presents itself as an appropriate paradigm precisely because it is designed for sequential decision-making in dynamic, uncertain, and reward-driven environments. Unlike supervised learning, which depends on labeled datasets, RL agents learn optimal policies by interacting with their environment through trial and error, continuously adjusting behavior to maximize long-term cumulative rewards [13].

The adoption of Deep Reinforcement Learning (DRL) for building energy management aligns with a significant global technology trend, where DRL is being intensively investigated for applications in smart buildings, smart grids, and demand response initiatives [36]. Implementing this advanced control paradigm in Vietnam represents a strategic opportunity to bridge the domestic technology gap and facilitate adherence to international standards for energy efficiency and building intelligence.

In this project, we first examined DDPG for several reasons: its ability to handle continuous action spaces, its stable actor–critic learning mechanism, and its capacity for precise multi-objective HVAC control. In addition, the project also investigates and compares representative DRL algorithms such as DQN or PPO to examine differences in learning stability, adaptability to environmental variability, convergence speed, and their effectiveness in optimizing energy use while maintaining indoor environmental quality. This comparison helps identify the algorithm best suited to Vietnam's climate conditions and the thermal-load characteristics of office buildings.

2.3 Simulation Environment

The project adopts Modelica as the simulation platform because we currently lacks sufficient real-world data; most collected datasets are from periods too far in the past to be directly applicable. Therefore, a simulation-first approach (Modelica/EnergyPlus) is appropriate to create a unified environment for comparing RBC, MPC, and DRL under consistent conditions. Initially, the team selected Modelica due to several advantages: the Buildings Library from Lawrence Berkeley National Laboratory enables detailed modeling of thermal and fluid systems; seamless integration with Python via PyFMI; its acausal modeling paradigm accurately captures nonlinear interactions and thermal inertia; support for FMU/FMI facilitates coupling with control algorithms; and its modular structure allows scalable model development for diverse testing scenarios [26].

3. Project Objectives

The overarching objective of this project is to design, develop, and validate an intelligent HVAC control framework that is capable of learning, adapting, and anticipating in order to address the conflicting goals of energy efficiency, occupant comfort, and indoor air quality (IAQ) within the challenging context of Vietnamese office buildings. To achieve this, the project is structured around five key technical objectives that collectively form the foundation for a forecast-aware Deep Reinforcement Learning (DRL) controller:

Objective 1: Development of the Deep Reinforcement Learning Agent. The first objective is to develop and train a DRL agent capable of making sequential control decisions for HVAC operations. Rather than relying on rigid schedules or predefined rule sets, the RL agent learns from continuous interaction with a simulated environment how to switch HVAC subsystems on and off, adjust cooling or ventilation power levels, and fine-tune airflow rates in a manner that balances competing objectives over time.

Objective 2: Establishment of a stimulative Environment. The second objective is to establish a simulation environment using Modelica and Functional Mock-up Units (FMUs). By interacting with this high-fidelity simulation platform, the DRL agent is exposed to realistic building and HVAC dynamics, system constraints, and nonlinear behaviors. This environment provides a reliable training ground that enhances the transferability of learned policies to real-world deployments in Vietnamese office buildings.

Objective 3: Reward Function Design. The third objective is to design an effective reward function that captures the essence of the control challenge: the necessity to simultaneously balance between the energy consumption and occupant thermal comfort. The reward must penalize excessive energy use, especially during peak tariff hours, while also penalizing deviations of indoor conditions from acceptable comfort setpoints. Additionally, the reward design will incorporate indoor air quality penalties, particularly for elevated concentrations of CO_2 transforming the problem into a true multi-objective optimization framework that reflects the operational realities of urban Vietnamese office buildings.

Objective 4: Integration of Forecasting Capabilities. The fourth objective is to integrate short-term outdoor air temperature forecasting capabilities into the control framework. By adding these forecast values to the state of the reinforcement learning agent (RL agent), the controller can act proactively instead of merely reacting to real-time changes. However, due to limited resources and increasingly severe climate change, the forecasting results are not always as expected. Therefore, we will refine a basic framework to provide a direction for future projects to develop and optimize more accurately.

Objective 5: Comparative Contribution and Practical Impact. The final objective is to establish a forecast-aware RL framework as a reference for the HVAC and smart-building research community by demonstrating clear advantages over traditional control approaches such as Rule-Based Control (RBC) and Model Predictive Control (MPC). Through systematic comparisons against RBC and MPC baselines, the project seeks not only to demonstrate superior performance but also to provide strong evidence of robustness, scalability, and practical applicability in real-world commercial buildings.

4. Significance of the Project

The significance of this project extends far beyond its immediate technical contribution. It represents a scientific, practical, societal, and policy-relevant intervention into one of the most pressing issues of sustainable urban development: how to design and operate buildings that are simultaneously energy-efficient, health-promoting, and resilient to environmental and economic variability. In the context of Vietnam, where office buildings are proliferating rapidly in major cities such as Hanoi, Ho Chi Minh City, and Da Nang, and where climate change and urban pollution are intensifying existing challenges, the outcomes of this project could serve as both a proof of concept and a model for future practices. The following paragraphs elaborate the significance across multiple dimensions in detail.

5. Scientific Significance

- This project expands HVAC control research by treating Indoor Air Quality (IAQ) as an equal objective alongside energy efficiency and thermal comfort, addressing a gap where prior studies often isolate IAQ from the control problem.
- It advances Reinforcement Learning (RL) for cyber-physical systems by integrating predictive signals such as weather, occupancy, and pollution forecasts, enabling proactive rather than reactive control.
- The project introduces a dynamic, context-aware reward function that adapts to electricity tariffs, comfort deviations, and IAQ thresholds, offering a methodological innovation applicable to other multi-objective control domains.

6. Practical Significance

- The framework addresses real challenges for building operators in Vietnam, where HVAC dominates energy use and peak-time tariffs drive high operational costs.
- By aligning HVAC control with forecasts and tariff schedules, the system reduces energy expenditures while maintaining comfort and IAQ.
- Unlike rigid rule-based systems, the forecast-aware RL approach adapts to changing conditions, reducing manual intervention, minimizing occupant complaints, and improving compliance with emerging IAQ requirements.

- In Vietnam's rapidly expanding commercial building sector, adopting intelligent control helps prevent long-term inefficiencies and sets a higher standard for indoor environmental performance.

7. Societal Significance

- Integrating IAQ into HVAC control enhances indoor environmental health, mitigating the effects of elevated CO_2 and PM2.5 on cognition and long-term well-being.
- Benefits include improved productivity, reduced absenteeism, and healthier working conditions, which is critical for Vietnam's growing knowledge-based economy.
- Ensuring clean air and stable comfort contributes to stronger organizational performance and healthier communities.

8. Policy Significance

- The project aligns with Vietnam's national green-growth strategies and its net-zero 2050 commitment by demonstrating how AI-driven HVAC control can yield substantial energy savings without sacrificing comfort.
- It provides insights that can inform future building codes, energy standards, and incentive policies promoting intelligent control technologies.
- As IAQ becomes a rising regulatory priority, the project anticipates future Vietnamese standards by embedding IAQ directly into the control framework, positioning it as a model for compliance and best practice.

9. Project Scope and Limitations

9.1 Project Scope

This project focuses on a Deep Reinforcement Learning based HVAC control model designed for deployment, simulation, and operational evaluation in office buildings. The scope includes:

- A functional control package that enables integration of the RL-based controller into real or simulated HVAC systems, supporting components such as AHUs, FCUs, and zone-level control interfaces.
- A simulation environment developed using a building thermal model and environmental datasets to emulate indoor temperature, humidity, IAQ conditions, and HVAC operational responses under different weather and occupancy scenarios.
- User guide and technical documentation, covering:
  - How to install and configure the control framework.
  - How to run simulations with predefined and custom building scenarios across multiple climate regions (Hanoi, Da Nang, Ho Chi Minh City).

  - How to deploy the trained DRL model into a compatible building management system or digital twin.

- A pre-trained DRL model and associated parameters required for deployment and further finetuning.
- Support for integration with real-time or batch data pipelines, including indoor sensor streams, HVAC operational logs, and short-term outdoor temperature forecasts.

9.2 Project Limitations

Despite its practical contributions, this project also includes some limitations:

- The evaluation is conducted primarily in a simulation environment. Real-world deployment and field testing are not included within the scope of this project.
- The study focuses mainly on cooling-related HVAC operations. Heating modes and more complex hybrid HVAC configurations are not comprehensively addressed.
- DRL requires long training times and substantial computational resources; retraining the agent whenever the building configuration or HVAC system changes can incur significant operational costs. As a result, the objective of minimizing overall cost may not be fully achieved in practice.
- The framework does not fully account for practical HVAC safety constraints or operational rules (e.g., limits on setpoint change rates, equipment maintenance schedules), and therefore requires additional adjustments before deployment in real building systems.

II. Project Management Plan

1. Team Work

1.1 Team Structure and Roles

Our project team operates within a collaborative group structure consisting of one leader and three members. To promote effective coordination, task management, and timely delivery, each role has been assigned clearly defined responsibilities.

Team Members:

- Contribute to core development, testing, and documentation activities.
- Prepare the trained model for deployment to ensure practical applicability.
- Develop user-oriented deployment and simulation guidelines.
- Establish and integrate the simulation environment to support evaluation.

Team Leader:

- Coordinates and maintains the overall project plan, milestones, and timeline.
- Allocates tasks equitably while monitoring progress to ensure deadlines are achieved.
- Serves as the primary liaison with the project advisor.
- Engages actively in all project activities alongside the members.

1.2 Communication Plan

Table 3: Project Communication and Coordination Plan

Activity	Participants	Responsibility	Schedule	Platform
Daily Meeting	All members	Team leader moderates discussions, consolidates updates, and ensures alignment of tasks.	9:00 AM Daily	Offline / Online
Mentor Meetings	All Team & Mentor	Team leader prepares the agenda, progress reports, slides, and meeting minutes to share with the supervisor.	17:30 PM Every Tuesday	Offline / Online (Google Meet)
Project Management	All members	The leader monitors progress, reviews set goals for completion, assigns and manages tasks.	Ongoing	Jira
Chat Communication	All members	Brief communications, critical questions, and prompt team notifications.	As needed	Facebook, Zalo, Discord
Monthly Milestone Review	All Team & Mentor	Evaluate results against project milestones and adjust scope if necessary.	End of each month	Offline

Effective communication plays a critical role in fostering seamless collaboration among team members, supervisors, and stakeholders. To ensure consistency and clarity in the exchange of project-related information, the communication plan specifies the platforms, purposes, and responsibilities as follows:

- **Google Meet:** employed for scheduled meetings with mentors and academic advisors, ensuring professional engagement and guidance.
- **Excel:** utilized as the primary platform for task allocation, progress monitoring, and systematic project tracking.
- **Facebook, Zalo:** applied for rapid inquiries, urgent notifications, and instant peer-to-peer communication to support timely coordination.

- **Google Drive:** used to store reference materials, raw data, and experimental results.

2. Project Management Approach

2.1 Project Management Framework

In this project, we apply Scrum, an Agile framework illustrated in the Figure 2, to manage development through iterative sprints and continuous feedback. Scrum is chosen because it enables flexible planning, effective task management and tracking through sprints, continuous feedback, and incremental improvement throughout the project. The development process is organized into seven sprints, each lasting two weeks, allowing regular progress evaluation and timely adjustment of project objectives and implementation strategies.

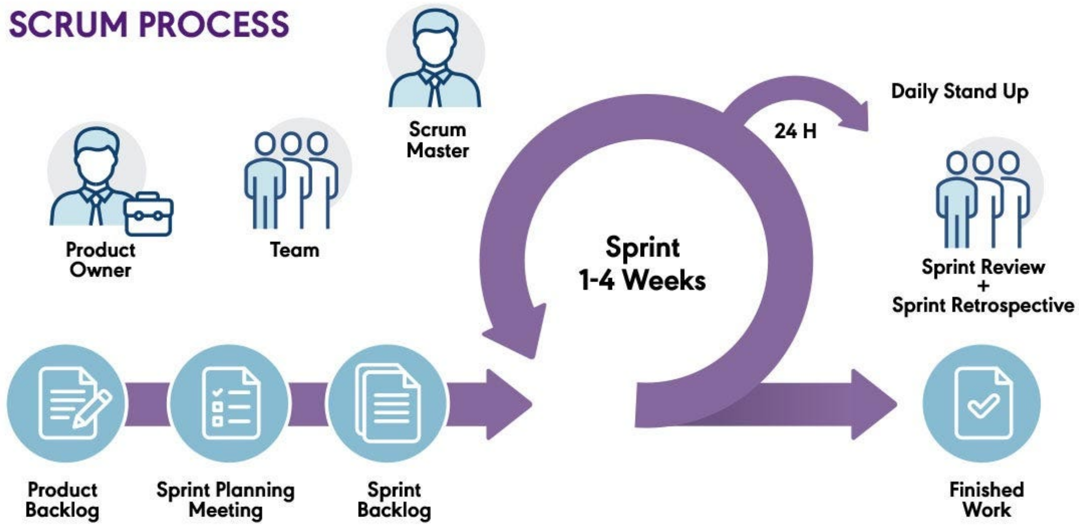


Figure 2: Scrum Framework

Each sprint includes planning, implementation, review, and retrospective phases, ensuring that project goals are clearly defined, progress is continuously monitored, and improvements are incorporated into subsequent sprints. This structured yet flexible approach supports steady development, risk mitigation, and alignment with project requirements throughout the project lifecycle.

2.2 Milestone List

The table 4 below summarizes the key milestones of the SmartHVAC Project, focusing on major phase completions and gate reviews. While smaller milestones are not shown, they are detailed in the project schedule and Work Breakdown Structure (WBS) in Table 5. Any schedule delays that could affect a milestone or delivery date must be reported promptly to the project manager to allow timely mitigation. All approved changes to milestones or timelines will be communicated to the project team by the project manager.

Table 4: Project Milestones

Milestone	Description	End Date
Complete Requirements Analysis	Define project objectives and scope; review related studies; analyze HVAC systems in Vietnam; collect and assess datasets; select simulation tools and RL frameworks.	21/09/2025
Complete Simulation Environment Setup	Establish the Modelica-based simulation environment; integrate weather datasets; define the MDP structure; implement baseline thermostat control; validate and document the simulation architecture.	05/10/2025
Complete Forecasting Model Development	Collect and preprocess weather data; train and evaluate forecasting models; integrate forecasts into the RL state space; complete documentation.	19/10/2025
Complete DRL Agent Development	Finalize the state–action–reward design; train DRL agents; tune hyperparameters; evaluate convergence and seasonal performance; log results and save checkpoints.	02/11/2025
Complete Optimization Phase	Optimize the reward function; fine-tune DRL agents; compare algorithm performance; benchmark DRL with and without forecasting integration.	16/11/2025
Complete Benchmarking and Evaluation	Compare DRL controllers with baseline methods; evaluate energy consumption, comfort, and computational cost; test under varying weather conditions; generate evaluation visualizations.	30/11/2025
Complete Documentation and Handover	Finalize documentation; consolidate results; prepare presentation and handover materials; clean the codebase; summarize limitations and future work.	17/12/2025

2.3 WBS

Table 5: Task Details

No.	Sprint	Task Details	PIC	Start Date	End Date
1	Requirement Analysis			08/09/2025	21/09/2025
1.1		Project objective & scope	All	08/09/2025	14/09/2025
1.2		Review case studies	PhuongNH	15/09/2025	17/09/2025
1.4		Problem context	PhuongNH	15/09/2025	17/09/2025
1.5		Data collection	HonNV	15/09/2025	20/09/2025
1.6		Assess gaps in local datasets and propose synthetic data generation	HoanDVQ	15/09/2025	18/09/2025
1.7		Study common HVAC system designs in Vietnam (cooling-focused)	NguyenTK	15/09/2025	17/09/2025
1.8		Choose a simulation tool	HonNV	16/09/2025	21/09/2025
1.10		Review Python frameworks for RL integration (Stable-Baselines3, RLlib)	NguyenTK	18/09/2025	20/09/2025
2	Simulation Environment Setup			22/09/2025	05/10/2025
2.1		Install and configure simulation tool (Modelica)	All	22/09/2025	23/09/2025
2.3		Integrate synthetic and real weather datasets	HonNV	24/09/2025	29/09/2025
2.4		Design MDP model	NguyenTK	25/09/2025	27/09/2025
2.8		Implement reward function placeholder	NguyenTK	29/09/2025	03/10/2025
2.9		Run baseline test: thermostat on/off in environment	PhuongNH	03/10/2025	05/10/2025
2.10		Debug simulation outputs (stability check)	HoanDVQ	03/10/2025	05/10/2025
2.11		Document simulation environment architecture	HoanDVQ	29/09/2025	05/10/2025
3	Forecasting Model Development			05/10/2025	19/10/2025

Continued on next page

No.	Sprint	Task Details	PIC	Start Date	End Date
3.1		Weather data collection and preprocessing	PhuongNH	06/10/2025	08/10/2025
3.2		Train forecasting models (ARIMA, LSTM/GRU, TCN)	HoanDVQ	08/10/2025	12/10/2025
3.3		Evaluate errors and select the best-performing model	HoanDVQ	12/10/2025	14/10/2025
3.4		Integrate forecast outputs into the RL state	HonNV	14/10/2025	17/10/2025
3.5		Documentation	PhuongNH	17/10/2025	19/10/2025
4	DRL Agent Development			20/10/2025	02/11/2025
4.1		Finalize the state–action–reward design	NguyenTK	20/10/2025	23/10/2025
4.2		Train multiple agents (DDPG, PPO)	All	24/10/2025	28/10/2025
4.3		Tune hyperparameters	NguyenTK	29/10/2025	30/10/2025
4.4		Evaluate convergence and seasonal performance	All	31/10/2025	01/11/2025
4.5		Log results and save checkpoints	HonNV	31/10/2025	02/11/2025
5	Optimization			03/11/2025	16/11/2025
5.1		Optimize the reward function	HonNV	03/11/2025	06/11/2025
5.2		Fine-tune DRL agents	NguyenTK	07/11/2025	09/11/2025
5.3		Compare performance across algorithms	HoanDVQ	10/11/2025	12/11/2025
5.4		Benchmark DRL with and without forecasting	NguyenTK	13/11/2025	16/11/2025
6	Benchmark and Evaluation			17/11/2025	30/11/2025
6.1		Compare DRL with baseline controllers	HoanDVQ	17/11/2025	20/11/2025
6.2		Evaluate energy use, comfort, and computation cost	NguyenTK	21/11/2025	23/11/2025
6.3		Test under varying weather conditions	HonNV	24/11/2025	26/11/2025
6.4		Generate charts, heatmaps, and evaluation results	PhuongNH	27/11/2025	30/11/2025

Continued on next page

No.	Sprint	Task Details	PIC	Start Date	End Date
7	Documentation and Handover			01/12/2025	17/12/2025
7.1		Write technical and research documentation	PhuongNH	01/12/2025	04/12/2025
7.2		Consolidate results and visualizations	HoanDVQ	05/12/2025	07/12/2025
7.3		Prepare slides, demo, and handover materials	All	08/12/2025	10/12/2025
7.4		Clean codebase, write README, publish repository	All	11/12/2025	14/12/2025
7.5		Summarize limitations and future work	All	15/12/2025	17/12/2025

2.4 Deliverables

Table 6: Deliverables by Sprint

No.	Sprint Phase	Deliverables
1	Requirements Analysis	Requirements specification; KPI list; selected tools and RL frameworks; climate and HVAC study report.
2	Simulation Environment Setup	Simulation environment (Modelica); RL-compatible Gym wrapper; baseline thermostat results; simulation architecture documentation.
3	Forecasting Model Development	Weather forecasting models (ARIMA, LSTM/GRU, TCN); forecast performance report; integrated forecasting pipeline.
4	DRL Agent Development	DRL agent prototypes (DDPG, PPO, SAC); training logs and checkpoints; initial performance comparison.
5	Optimization	Optimized DRL and forecasting model; algorithm comparison charts; ablation and stress test results.
6	Benchmark and Evaluation	Benchmark report (DRL vs baseline controllers); energy and comfort evaluation charts; heatmaps and sensitivity analysis.
7	Documentation and Handover	Full technical documentation; final research report; presentation slides and demo; cleaned GitHub repository with README; deployment recommendations.

2.5 Risk Management

Effective risk management is critical to ensuring the stability and successful execution of the HVAC–DRL project. By systematically assessing risks in terms of their impact and likelihood, potential challenges can be anticipated early, enabling the implementation of appropriate mitigation strategies to maintain project continuity.

Key risks identified include high computational requirements and practical deployment constraints. To address these challenges, scalable cloud computing resources were utilized for flexible CPU and GPU allocation, while Docker-based containerization was adopted to ensure software reproducibility, avoid dependency conflicts, and support seamless integration with existing Building Management Systems (BMS). The identified risks and their corresponding mitigation measures are summarized in Table 7.

Table 7: Risk Assessment Summary

No.	Risk Description	Impact	Probability	Priority
1	Insufficient data availability or poor data quality may lead to inaccurate HVAC modeling and unreliable DRL training outcomes.	High	High	High
2	Intermittent or outdated weather data may reduce prediction accuracy and distort control performance.	High	High	High
3	Lack of official IAQ regulatory standards in Vietnam may weaken evaluation credibility.	Medium	Medium	Medium
4	DRL model instability or non-convergence can prevent policy learning and invalidate experimental results.	High	Medium	High
5	Overfitting of forecasting models may degrade real-world generalization performance.	Medium	Medium	Medium
6	Limited practical HVAC operational experience in Vietnam may lead to unrealistic assumptions.	Medium	High	High
7	Difficulty in designing multi-objective reward functions may cause biased optimization.	High	Medium	High
8	Integration challenges between PyFMI, Python, and FMU models may cause simulation failures.	High	High	High
9	Computational resource constraints may slow training and limit experiment scale.	High	High	High
10	Communication and collaboration gaps within the project team may delay progress.	Medium	Medium	Medium

The project's risk mitigation framework consisted of:

- Continuous identification and documentation of risks related to data quality, model performance, and computational resources throughout the research lifecycle.

- Regular risk reviews integrated into sprint reviews and project meetings to reassess priorities and adjust mitigation plans.
- Clear assignment of responsibilities within the team for monitoring specific risks (e.g., data pipeline, DRL training stability, forecasting accuracy).
- Consultation with academic supervisors and domain experts when facing high-impact risks such as convergence issues or resource limitations.

Table 8: Risk Mitigation and Contingency Plans

No.	Mitigation Strategy	Contingency Plan
1	Use public and synthetic datasets; perform data cleaning, normalization, and augmentation; document assumptions.	Simplify model scope; reduce state space; rely on validated benchmark datasets.
2	Split data seasonally; combine multiple EPW/weather sources; validate against recent averages.	Restrict experiments to representative climate scenarios.
3	Adopt WHO and ASHRAE reference standards; justify thresholds academically.	Reframe IAQ metrics as reference indicators rather than compliance metrics.
4	Hyperparameter tuning; use stable algorithms (PPO); early stopping and checkpointing.	Switch to simpler control baselines (RBC/MPC).
5	Apply cross-validation, dropout, and regularization; monitor validation loss.	Retrain using simplified architectures or reduced feature sets.
6	Consult HVAC engineers; reference ASHRAE and TCVN standards; design realistic scenarios.	Restrict conclusions to simulation-based insights; propose future field validation.
7	Use weighted and seasonal reward formulations; benchmark against baseline controllers.	Decompose objectives and evaluate them separately.
8	Start with simplified FMU models; detailed error logging; modularized integration code.	Replace FMU with reduced-order models.
9	Optimize GPU usage; leverage cloud or HPC resources; efficient experiment scheduling.	Reduce episode length or model complexity.
10	Structured communication plan; GitHub version control; weekly sprint reviews.	Escalate issues to supervisors; reassign responsibilities if needed.

2.6 Cost Management Plan

The Cost Management Plan outlines how project costs are planned, monitored, and controlled throughout the project lifecycle. As the project relies on open-source tools, including Modelica and Python-based RL frameworks, and uses offline simulation, cost management focuses on time allocation, development effort, and computational resources rather than direct financial expenditure. Cost performance is tracked through progress against planned milestones and sprints, with the project leader responsible for cost control and approval of changes affecting scope or resources. Cost status is reviewed during regular sprint meetings and documented as part of the overall Project Management Plan.

2.7 Quality Management

Ensuring a high level of quality in all deliverables is crucial for the achievement and academic integrity of the HVAC project. To guarantee that the outputs fulfill both technical and presentation requirements, a structured quality management framework has been established throughout the project's duration. This framework directs the examination and refinement of all artifacts including documentation, code, datasets, and interface prototypes with the goal of enhancing clarity, accuracy, and functionality. The quality assurance process is driven by three fundamental principles: individual responsibility, collaborative assessment, and expert supervision.

Documentation Quality Management

Self-Review: Each team member holds individual accountability for the accuracy, clarity, and scholarly presentation of their assigned sections. Before advancing to the group stage, contributors carefully verify grammar, referencing style, formatting standards, and the technical soundness of their work to ensure academic integrity.

Group Review: After individual contributions are finalized, the team engages in structured collaborative review sessions. These discussions aim to achieve content consistency, logical flow, and alignment with the project's stated objectives and scope. Constructive feedback is exchanged openly, with revisions systematically documented to maintain transparency and traceability.

Mentor Review: Following the completion of internal quality checks, the consolidated document is submitted to the academic mentor or supervisor. This stage serves as the final quality assurance gate, where expert feedback addresses subject-matter accuracy, methodological rigor, academic tone, and the overall quality of presentation.

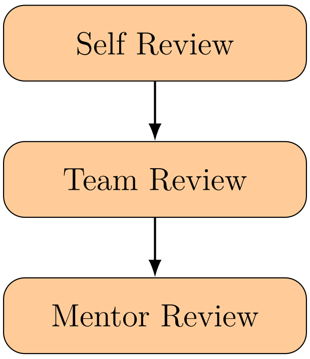


Figure 3: Document quality management

Code Quality Management

To ensure reliability, maintainability, and reproducibility of the HVAC–DRL software framework, a structured code quality management strategy was adopted. This strategy combines standardized coding practices, automated testing, and collaborative review workflows to preserve code integrity throughout development.

Unit Testing: Core modules, including environment interfaces, reinforcement learning agents, and forecasting components, are validated through unit testing. Tests are integrated into the continuous integration (CI) pipeline to detect logical errors early and prevent regression during system evolution.

Coding Standards: The codebase follows consistent Python coding conventions covering naming, modular structure, documentation, and formatting. Automated linting and formatting tools enforce these standards to support readability and long-term maintainability.

Code Review and Version Control: All code changes undergo peer review via GitHub pull requests to ensure correctness, reproducibility, and adherence to project guidelines. Git-based version control with structured branching strategies enables collaborative development, traceability, and controlled integration of new features.

Tool Used: GitHub.

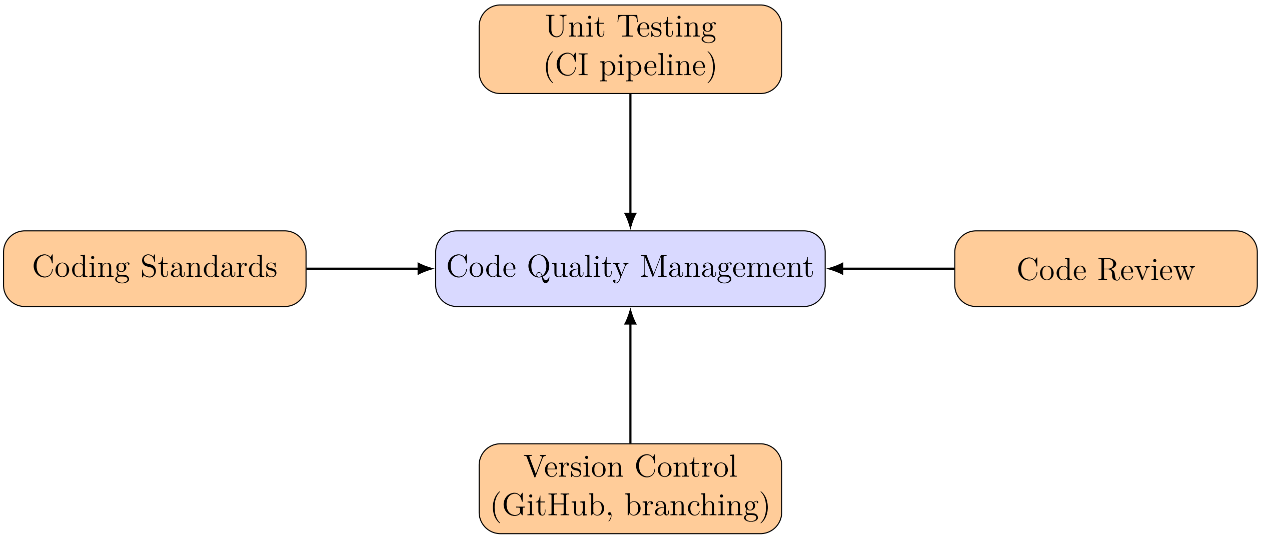


Figure 4: Code Quality Management

Dataset Quality Management

The foundational phase of the data management protocol was dedicated to establishing the integrity and consistency of the meteorological and operational datasets. This involved a systematic curation process, beginning with the harmonization of raw data drawn from heterogeneous sources such as EnergyPlus Weather (EPW) and the International Weather for Energy Calculations (IWEC). A rigorous data cleaning pipeline was developed to identify and rectify anomalies, including the imputation of missing values through statistical methods and the standardization of all physical quantities to SI units. To mitigate the risk of systemic bias inherent in any single source, a cross-validation methodology was employed. Key variables were compared across datasets, and statistical discrepancies were analyzed to produce a robust, synthesized time-series that minimizes source-specific artifacts and provides a more reliable basis for simulation. The provenance of each dataset, every transformation applied, and all underlying assumptions are explicitly recorded in the metadata. This documentation transparently states all known limitations such as the 25-year age of certain climatic data and its potential deviation from current conditions, ensuring that the research remains reproducible and that the results can be interpreted within a well-defined context of uncertainty.

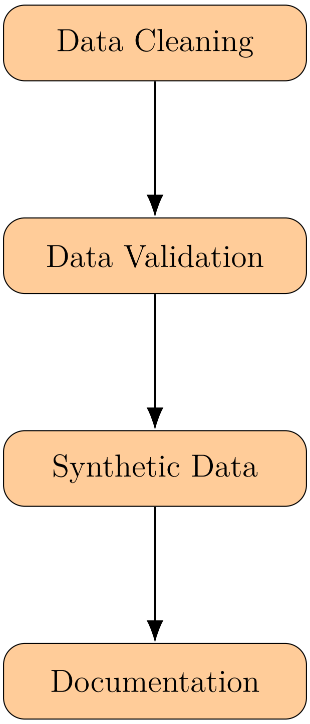


Figure 5: Dataset Quality Management

III. Existing Systems

1. Overview of the Field

Heating, Ventilation, and Air Conditioning (HVAC) systems play a central role in modern buildings, typically accounting for 40–60% of total energy consumption [5]. This makes HVAC both essential for occupant comfort and one of the most impactful targets for improving energy efficiency and reducing emissions.

Traditionally, buildings have relied on rule-based controls, fixed schedules, heuristics, and PID loops [12]. While simple and robust, these approaches struggle under real-world variability such as changing weather, fluctuating occupancy, and nonlinear thermal dynamics. Model Predictive Control (MPC) introduced a more advanced optimization framework, but its dependence on detailed building models limits its scalability, particularly as buildings age or usage patterns shift.

Reinforcement Learning (RL) has emerged as a promising model-free alternative, capable of learning control policies directly from data. RL has demonstrated 26.3% HVAC energy savings over rule-based baselines [4] by adapting dynamically to sensor inputs, occupancy trends, and electricity prices. With the integration of deep neural networks, Deep RL (DRL) extends this capability to high-dimensional states and continuous action spaces.

A range of DRL algorithms have been applied to HVAC control, including Deep Q-Networks (DQN) for discrete decisions, Deep Deterministic Policy Gradient (DDPG) and Twin Delayed DDPG (TD3) for stable continuous control, and Trust Region Policy Optimization (TRPO) and Proximal Policy Optimization (PPO) for reliable policy updates. Actor-critic variants such as A2C and A3C further accelerate training through parallelization.

Recent research increasingly integrates additional objectives including Indoor Air Quality (IAQ), demand response signals, and carbon intensity indicators positioning buildings as active participants in smart energy systems [1]. The ecosystem surrounding these methods has also matured, with standardized platforms such as BOPTEST, Sinergym, and EnergyPlus/Modelica co-simulation enabling reproducible benchmarking, and digital twins reducing the gap between simulation and deployment.

Despite rapid progress, relatively few studies validate DRL controllers in real buildings. Persistent challenges include ensuring training safety, achieving generalization across heterogeneous infrastructures, integrating with legacy Building Automation Systems (BAS), and gaining occupant acceptance. Current directions such as safe RL, offline RL using historical BAS data, and transfer learning aim to address these gaps.

Overall, the field is transitioning from rigid, model-based control strategies toward adaptive, data-driven approaches. As DRL technologies mature, emphasis is shifting from proving feasibility to ensuring scalability, interpretability, and trust, enabling HVAC systems to evolve into intelligent, responsive components of sustainable and resilient built environments.

2. Historical Context

The control of HVAC systems has a long history, starting from basic thermostats and rule-based systems and gradually advancing toward intelligent methods. Early studies emphasized manual schedules and PID controllers, which offered stable but limited regulation of indoor environments. These controllers reacted to deviations in temperature but lacked foresight, leading to frequent inefficiencies. For example, studies demonstrated how PID-based baselines often wasted energy during unoccupied hours, highlighting the need for smarter, adaptive methods [35].

During the late 1990s and early 2000s, researchers began to apply Model Predictive Control (MPC) in building operations. MPC allowed anticipatory strategies, using optimization over prediction horizons to manage heating and cooling loads. It provided practical demonstrations of MPC in radiant heating, showing improved efficiency but also revealing that performance degraded when the building model was inaccurate [39]. These challenges underscored the difficulty of developing detailed and reliable thermal models for diverse buildings, motivating the search for more data-driven approaches.

The first integration of Reinforcement Learning into HVAC appeared in simplified case studies. Researchers explored RL for optimizing heat pump control in Modelica-based building models [25], while others applied deep RL to space heating [22]. Both works confirmed that RL could adapt to uncertain dynamics and reduce energy consumption compared to static controllers. However, these early algorithms were typically tabular or discrete-action Q-learning, which limited scalability beyond single-zone or toy environments.

With the rise of DRL around 2018–2019, more sophisticated algorithms like DQN and A3C were introduced. For example, multi-agent RL was applied to thermostatically controlled loads [14], while DRL frameworks for advanced building control were demonstrated [11], highlighting their potential to outperform rule-based methods. These studies marked a turning point, as they showed that RL could process high-dimensional sensor data directly and optimize continuous control variables without handcrafted features.

Between 2020 and 2023, research expanded into continuous control and multi-zone coordination. Works such as those employing DDPG for energy-efficient thermal comfort [8] and multi-agent RL for intelligent multi-zone systems [19] advanced the field. At the same time, transfer learning [20] and offline RL emerged to tackle the issue of sample inefficiency [19]. These studies reflected a shift from proof-of-concept simulations toward methods addressing practical deployment challenges.

In the 2024–2025 period, the focus of RL research for HVAC shifted toward enhancing algorithmic robustness and expanding control objectives beyond energy savings. A significant trend involved the standardized evaluation of established algorithms, with studies systematically assessing the performance and adaptability of agents such as SAC and TD3 in complex scenarios [5]. Concurrently, new algorithmic refinements emerged; for instance, EntropySAC was introduced, demonstrating superior energy savings by using entropy to screen high-value information [30]. The scope of control also broadened to include multi-objective optimization, integrating carbon emissions and IAQ, as demonstrated by research combining DRL with transformer-based load forecasting for simultaneous cost and emission reduction, and the development of AI tools to design smarter HVAC filters for better IAQ.

3. Key Studies and Theories

The emergence of reinforcement learning in building control has been accelerated by a series of foundational studies that gradually expanded the scope of objectives, methods, and application contexts. Rather than evolving in isolation, these works build upon one another, collectively shaping a coherent trajectory for reinforcement learning in HVAC research.

Several studies have laid the foundation for RL application in building control.

Deep reinforcement learning control for co-optimizing energy consumption, thermal comfort, and indoor air quality in an office building [18]. A seminal contribution in this lineage is the study on multi-objective co-optimization of energy consumption, thermal comfort, and indoor air quality in office buildings. While most early efforts confined themselves to balancing energy efficiency and thermal comfort, this work pushed the boundary by introducing indoor environmental quality into the optimization framework, specifically CO_2 concentration and $PM_{2.5}$ levels. By employing the Deep Deterministic Policy Gradient (DDPG) algorithm, the study demonstrated how deep reinforcement learning could simultaneously navigate four inherently conflicting objectives. To achieve this, the authors constructed a hybrid simulation environment that combined detailed HVAC dynamics, building envelope characteristics, and pollutant transport processes. This environment struck a balance between physical fidelity and computational feasibility, enabling effective training of the DRL agent. The results were compelling: the

DRL controller reduced energy consumption by 21.4% compared to rule-based control, while maintaining comfort and showing greater robustness under disturbances than Model Predictive Control (MPC). Beyond numerical improvements, the theoretical significance lies in proving that model-free DRL can extend beyond conventional dual-objective paradigms to tackle multi-dimensional, real-world optimization tasks. In doing so, the study provided both a methodological breakthrough and a benchmark against which later work could be built.

Modelling building HVAC control strategies using a deep reinforcement learning approach [38]. Key studies in building HVAC control have evolved significantly over time, moving from simple, static models to highly adaptive, intelligent systems. The foundational approach, Rule-Based Control (RBC), relied on predefined rules to manage HVAC operations. While straightforward, this method is often criticized for its rigidity and inability to adapt to the dynamic and non-linear nature of real-world building environments, leading to suboptimal energy efficiency and comfort levels. To address these limitations, the field progressed to Model Predictive Control (MPC). This theory utilizes a system model—white-box (physics-based), black-box (data-driven), or grey-box (hybrid)—to forecast future conditions and optimize control decisions accordingly. However, the effectiveness of MPC is fundamentally tied to the accuracy of its underlying model, which can be difficult and costly to develop and maintain for complex buildings, thus limiting its practical application. The most recent and promising developments are centered around Deep Reinforcement Learning (DRL). This advanced, model-free approach allows an autonomous agent to learn optimal control strategies directly through interaction with the environment. By using a trial-and-error process to maximize a cumulative reward signal—typically balancing energy savings and occupant comfort—DRL systems can dynamically adapt to unforeseen changes like weather fluctuations and occupancy patterns. Applied research demonstrates that DRL overcomes the inherent limitations of both RBC and MPC, offering a more robust and flexible solution with significant potential for enhancing building energy efficiency.

Reinforcement learning for HVAC control and energy efficiency in residential buildings with BOPTEST simulations and real-case validation [29]. The theoretical landscape of HVAC control is undergoing a significant transformation, marked by a pivot from conventional methods to advanced, data-driven strategies, with Reinforcement Learning (RL) at the forefront. Seminal studies in this domain have demonstrated the efficacy of various RL algorithms, such as Deep Q-Networks (DQN) and Deep Deterministic Policy Gradient (DDPG), for optimizing energy consumption while maintaining occupant comfort. Despite these advancements, a persistent theoretical challenge within the literature has been the lack of standardized frameworks for impartially evaluating and comparing the performance of different RL agents. This inconsistency has often hindered the ability to generalize findings across studies. The research presented here addresses this fundamental issue by conducting a rigorous, systematic validation of a novel RL agent. By benchmarking its performance against both traditional Proportional-Integral (PI) and rule-based controllers in simulated and real-world residential settings, this work provides robust empirical evidence. A crucial element of this study is its progression from high-fidelity simulations to deployment in real-world residential settings, directly confronting the sim-to-real transfer problem that often impedes the practical application of RL. The findings confirm the agent's superior adaptability and efficiency, thereby offering a crucial contribution to the field by validating the practical viability of RL-based control beyond pure simulation and providing a more structured methodology for future comparative studies.

Towards healthy and cost-effective indoor environment management in smart homes: A deep reinforcement learning approach [37]. Based on this article, the Deep Deterministic Policy Gradient (DDPG) algorithm is referenced as a significant deep reinforcement learning method employed in previous studies for managing thermal comfort. Specifically, the paper cites the work of Gao et al., who utilized the DDPG algorithm to learn a control strategy that minimizes a joint cost function of HVAC energy consumption and thermal discomfort. DDPG is recognized as an actor-critic, model-free algorithm designed to operate in continuous, high-dimensional action spaces, making it well-suited for nuanced control tasks like adjusting HVAC systems where actions are not merely "on" or "off". It concurrently learns a Q-function (the critic) and a policy (the actor), using the critic to guide the learning of the actor. While the paper acknowledges the application of DDPG in the field, it ultimately proposes a different approach for indoor environment management: a combination of double deep Q-network (DDQN) with prioritized experience replay (PER), addressing the complexities and uncertainties of their problem formulation.

Energy-efficient control of thermal comfort in multi-zone residential HVAC via reinforcement learning [37]. The Deep Deterministic Policy Gradient (DDPG) algorithm is applied to derive an optimal control strategy for a multi-zone HVAC system, with the dual objectives of minimizing energy consumption while maintaining occupant thermal comfort. DDPG, a model-free, deep reinforcement learning algorithm, is specifically selected for its proficiency in handling continuous action spaces, making it inherently suitable for the nuanced task of adjusting HVAC setpoints. It functions on an actor-critic framework, wherein the actor network determines the control action (e.g., a specific temperature setpoint), and the critic network evaluates the quality of that action to guide learning. A crucial consideration in this application is that the DDPG agent is integrated with a pre-trained thermal comfort prediction model (SVR-DNN). The comfort value predicted by this model serves as a key component of both the state representation and the reward function, effectively steering the agent's learning toward policies that balance energy efficiency and occupant satisfaction. Moreover, DDPG operates without discretizing the action space, allowing for more granular and flexible control that avoids potential performance degradation from action quantization.

These studies form a coherent chain of theoretical and empirical contributions. Starting from the co-optimization of energy, comfort, and air quality, the field advanced toward scalable thermodynamic modeling, operational robustness in multi-zone precooling, occupant personalization through multi-agent systems, and hybrid predictive control integration. Each study addressed a specific gap while linking logically to the next, collectively charting the evolution of RL-based HVAC control from proof of concept to a robust paradigm for intelligent building management. This trajectory underscores reinforcement learning's transformative potential—not merely as a tool for efficiency, but as the cornerstone of adaptive, scalable, and human-centered building systems in the era of sustainable and smart urban environments.

4. Technological Advancements

In recent years, the domain of intelligent HVAC control has been profoundly reshaped by several technological advancements that collectively provide both the theoretical underpinnings and the practical tools for next-generation building management. These developments span algorithmic breakthroughs in reinforcement learning, predictive modeling, and cyber physical integration, each of which exerts a transformative influence on the feasibility and effectiveness of projects such as the one proposed here.

A central advancement is the maturation of Deep Reinforcement Learning algorithms. Early explorations of reinforcement learning in building control were limited to tabular Q-learning or simplistic policy iteration, which, while conceptually sound, proved impractical for the high-dimensional, continuous dynamics inherent in HVAC systems. The emergence of modern DRL methods including Deep Deterministic Policy Gradient (DDPG), Proximal Policy Optimization (PPO), and Deep Q-learning (DQN) has altered this landscape decisively [34]. These algorithms are capable of learning stable policies directly from raw, high-dimensional data streams, handling both continuous action spaces (such as modulating fan speeds or chiller loads) and stochastic state transitions.

For this project, these advancements are foundational: they permit the design of controllers that transcend binary on/off switching and instead discover nuanced, fine-grained strategies that adapt dynamically to the non-stationary conditions of real buildings. The theoretical improvements in stability, sample efficiency, and convergence reliability associated with these algorithms substantially lower the barriers to real-world deployment, ensuring that control policies are not only effective in simulation but also robust under uncertain operational environments.

In parallel, the accuracy of forecasting models has advanced significantly, driven by innovations in machine learning and statistical learning methods. Traditional HVAC control systems have relied almost exclusively on reactive strategies, adjusting to the current state without anticipation of near-future conditions. Recent developments in time-series forecasting, including recurrent neural networks (RNN), temporal convolutional networks (TCN), and hybrid physics-informed models, have dramatically improved the ability to predict dynamic exogenous factors such as outdoor temperature, solar irradiance, occupancy schedules, and even dynamic electricity tariffs [40]. These forecasting capabilities directly enhance the functionality of DRL agents by embedding foresight into the decision making process: rather than simply reacting to disturbances after they occur, the controller can proactively adjust system operation in anticipation of predicted events.

To build an accurate climate forecasting model for the context of Vietnam, we propose an advanced hybrid approach that combines artificial intelligence and multi-source climate data. The foundation of this model is the integration of data from various sources, including standardized weather data from EnergyPlus (EPW), information from local meteorological stations, and real-time data collected from IoT sensors installed in buildings. To enhance accuracy and stability, we apply a hybrid AI architecture, combining the power of deep learning models such as RNN and TCN with the robustness of traditional statistical models such as ARIMA and SARIMA.

The synergy between predictive forecasting and Deep Reinforcement Learning (DRL) represents a critical architectural element of this advanced control system, elevating it from a reactive to a proactive paradigm [27]. In this framework, the forecasting model functions as the eyes of the DRL agent, extending its perceptual horizon into the future. By integrating predictions of key variables—such as ambient temperature, solar irradiance, occupancy levels, and electricity price fluctuations—into the agent's state representation, the system can formulate policies that are not just instantaneously optimal but strategically robust over time. This foresight allows the control agent to make anticipatory decisions, such as pre-cooling a space before a predicted heatwave or shifting energy consumption to off-peak hours, thereby unlocking a level of operational intelligence and efficiency unattainable for purely reactive controllers.

An important strategy to improve the adaptability of RL agents is to divide the training data according to the seasons typical of Vietnam. Specifically, the data is clearly separated into hot and humid summers and cold and dry winters in the North, along with rainy and dry seasons in the South. This approach allows the agent to learn optimal HVAC control behaviors appropriate to each specific seasonal climate condition. However, implementation faces practical limitations, as historical data in Vietnam is often discontinuous, asynchronous, and can be 20–25 years old. Therefore, pre-processing steps such as cleaning, interpolation, and even synthetic data generation are extremely necessary. In the current project framework, the model mainly focuses on forecasting outdoor temperature, but it can be fully extended to include other important air quality parameters such as CO_2 and $PM_{2.5}$ in the future, provided that the input data source is large and reliable enough.

Equally transformative has been the rise of Digital Twin technology and high-fidelity simulation platforms. The integration of advanced building simulators such as EnergyPlus, TRNSYS, and Modelica-based co-simulation frameworks with real-time sensing data has given rise to digital twins—virtual replicas of physical building systems that mirror their thermal, mechanical, and operational dynamics in real time. Digital twins serve as powerful intermediaries, allowing algorithms to be trained, validated, and stress-tested in controlled yet realistic environments before field deployment. For HVAC-focused DRL research, this is particularly indispensable: real-world exploration can be unsafe, costly, and time-consuming, whereas training exclusively in simplified models risks poor generalization.

By leveraging digital twins, it becomes possible to conduct thousands of training episodes at negligible cost, evaluate performance under extreme scenarios that would be infeasible to test in reality, and transfer well-tuned policies to physical systems with minimal adaptation. For the proposed project, this capability shortens the development cycle, reduces deployment risk, and ensures that the control strategy achieves operational robustness from the very first day of use.

Taken together, these technological advancements—algorithmic maturity in DRL, predictive power in forecasting models, and integration via digital twins, define the current frontier of smart building control. Each advancement resolves a longstanding limitation: DRL addresses the inability of classical controllers to adapt to dynamic complexity; forecasting models overcome the shortsightedness of reactive strategies; and digital twins mitigate the risks and inefficiencies of real-world experimentation.

Beyond performance within a single building, a core objective of this framework is to establish a blueprint for scalable and transferable intelligent control. A bespoke DRL model trained exhaustively for one building may not perform optimally in another due to unique architectural features, occupancy patterns, and HVAC system configurations. For this project, the convergence of these technologies provides not merely incremental improvements but a fundamentally new operational paradigm. By building on this foundation, the proposed system can aspire to deliver scalable, adaptive, and sustainable HVAC control, aligning with both the technical demands of modern building management and the broader societal imperative of energy efficiency and decarbonization. This drastically reduces the data, time, and computational requirements for each new deployment, creating a viable and cost-effective methodology for scaling advanced AI control across entire portfolios of commercial and residential buildings.

5. Comparison of Existing Systems

Table 9: Comparison of Existing Systems

HVAC Control Strategy	Description	Advantages	Disadvantages
Rule-Based Control (RBC)	Operates based on predefined if–then rules, e.g., if temperature > 25°C, turn on AC.	• Simple and easy to implement. • Low installation cost. • Predictable behavior.	• Inability to adapt to changing environments. • Purely reactive, cannot leverage forecasts. • Low energy efficiency, leading to waste.
Model Predictive Control (MPC)	Uses a mathematical model of the building to predict and optimize control actions over a future horizon.	• Proactive, capable of using forecast data. • Handles system constraints effectively. • High optimal performance with an accurate model.	• Heavily dependent on model accuracy. • Very difficult and costly to build and calibrate for each building. • Less flexible when building conditions change.
Reinforcement Learning (RL)	An agent learns to make optimal decisions through trial-and-error by interacting directly with the environment to maximize a reward signal.	• Capable of continuous learning and adaptation. • Does not require a mathematical model (model-free). • Handles complex, non-linear environments.	• Can still be reactive if not integrated with forecasts. • Requires large datasets and long training times. • Safety and stability challenges in real-world deployment.

6. Gaps and Justification in the Literature/Technology

To address the inherent challenges in applying Reinforcement Learning (RL) to HVAC control systems, we propose an integrated architecture that combines the RL control module and short-term forecasting models (e.g., weather, electricity prices). This approach systematically addresses four key constraints: the Sim-to-Real Gap, safety and stability, reward function design, and generalization ability.

6.1 Overcoming the Sim-to-Real Gap

Challenge: The performance of DRL agents often degrades significantly in real-world deployments due to inevitable discrepancies between the simulated environment and the actual building dynamics.

Proposed solution: By integrating short-term weather forecasting models, the system dynamically adapts to upcoming environmental conditions. For example, when the model predicts a peak heat wave within the next two hours, the RL agent can proactively adjust its control strategy. This allows the RL agent to respond flexibly to sudden changes in the real environment, instead of adhering to an outdated simulation model, thereby effectively narrowing the gap between simulation and reality.

6.2 Ensuring System Safety and Stability

Challenge: DRL's exploration process can lead to unwanted actions, causing inconvenience to users or wear and tear on equipment due to sudden changes in state.

Proposed solution: A proactive control approach based on forecasting directly addresses this problem. Instead of performing random exploratory actions, the system can look ahead and plan actions within a safe range. For example, a heatwave forecast allows the system to pre-cool the space gradually, avoiding overloading the equipment and ensuring that the space does not become stuffy when the temperature rises. This strategy ensures that the exploration and operation process always stays within defined safety and comfort limits.

6.3 Designing a Dynamic and Economically Optimized Reward Function

Challenge: Designing a reward function that effectively balances the two often conflicting goals of saving energy and maximizing comfort is extremely difficult.

Proposed solution: We propose a dynamic, economically intelligent reward function that automatically adjusts the weights of the objectives based on the forecast of the electricity price. When the system predicts that the electricity price is approaching peak hours, it will give maximum priority to saving energy (e.g., taking advantage of previously stored thermal energy). Conversely, during low electricity price hours, it will give more priority to maximizing user comfort. This creates an optimal operating strategy that is not only technically but also financially efficient.

6.4 Enhancing Generalization and Transferability

Challenge: A control policy optimized for one building is often not transferable to another building, requiring costly and time-consuming retraining.

Proposed solution: The proposed architecture clearly separates the predictive module and the DRL control module, providing superior transferability. In the predictive module, when deployed in a new building, only a relatively small amount of data is needed to retrain this module to match the thermodynamic characteristics of that building.

IV. Methodology

1. Building Modeling and HVAC system

The Building Modeling and HVAC System constitutes a high-fidelity digital twin that emulates the thermodynamic, fluidic, and control interactions of a modern office building. Developed in Modelica using the Buildings and Modelica Standard libraries, the model employs acausal equations to maintain rigorous energy and mass conservation across coupled domains.

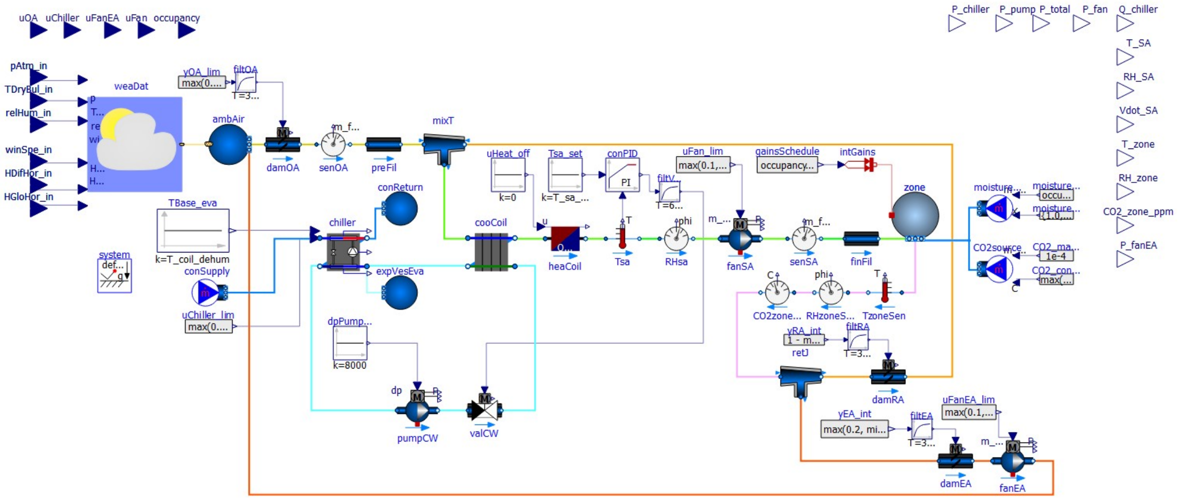


Figure 6: System architecture of the HVAC digital twin showing color-coded inter-domain connections.

- **Blue**: Weather & boundary data, operational references, safety/logic limits.
- **Green**: Supply-air (SA) stream from outside/return mixing → coils → supply fan → delivered to the zone.
- **Purple/Magenta**: Return/Exhaust-air (RA/EA) stream from zone through return damper and exhaust fan, with a branch back to mixing.
- **Cyan**: Chilled-water (CHW) loop: chiller ↔ pump ↔ control valve ↔ cooling coil ↔ back to chiller.

- **Gray/Black (thin lines)**: Control/measurement signals (setpoints, PI output, limiters, filters).
- **Orange frame**: Logical boundary for the AHU + Zone supervisory layer (scheduling, occupancy-dependent gains).

Each component is parameterized with empirical performance data, capturing real equipment characteristics and operational constraints. The integrated framework facilitates dynamic interaction with advanced control algorithms, including Reinforcement Learning (RL), enabling co-simulation via Functional Mock-up Interfaces (FMI). This configuration ensures accurate reproduction of multi-zone energy exchanges and environmental responses under Vietnam's tropical climatic conditions, providing a robust platform for intelligent HVAC control development and validation.

Here's a concise table of the main blocks in our Modelica HVAC:

Table 10: Main Blocks in our Modelica HVAC

Domain	**Full Name**	`Abbrev`	**Description / Function**
Control	Reheat Disable Logic	`uHeat_off`	Boolean logic to inhibit reheat during cooling-only.
	SA Temp Setpoint	`Tsa_set`	Reference value for supply-air temperature.
	PI Temp Controller	`conPID`	Tracks `Tsa_set` by modulating valve/fan.
	Occupancy Schedule	`gainsSchedule`	Time-of-day signal driving gains/ventilation.
	OA Damper Limiter	`yOA_lim`	Clamps outdoor-air command to regulatory bounds.
	EA/Fan Limiter	`yEA_int`	Limits exhaust damper/fan command magnitude and rate.
	OA Command Filter	`filtOA`	Smooths abrupt outdoor-air damper commands.
	EA Command Filter	`filtEA`	Avoids oscillatory exhaust damper/fan actions.
	CO2 Clamp	`CO2_conc_max`	Threshold/limit used by IAQ supervisory logic.
	DP Pump Controller	`dpPump`	PI/logic maintaining loop `dp` at setpoint.
	Supply Aggregator	`conSupply`	Combines schedules/limits (e.g., dehumidification).
	Chiller Limiter	`uChiller_lim`	Saturation/rate limit for chiller on/off or load command.
	Base Eva Temp Ref	`TBase_eva`	Base evaporator temperature reference.

Continued on next page

Domain	Full Name	Abbrev	Description / Function
Meter	Chiller Power	`P_chiller`	Instantaneous electrical power of chiller (kW).
	Pump Power	`P_pump`	Instantaneous electrical power of CHW pump (kW).
	Supply Fan Power	`P_fan`	Supply-fan electrical power (kW).
	Exhaust Fan Power	`P_fanEA`	Exhaust-fan electrical power (kW).
	Total Power	`P_total`	Sum of electrical consumers (chiller + pumps + fans).
	Cooling Duty	`Q_chiller`	Cooling capacity at evaporator / coil load (kW).
Output	SA Temperature	`T_SA`	Supply-air temperature monitor (post-coils).
	SA RH	`RH_SA`	Supply-air relative humidity.
	SA Flow Rate	`Vdot_SA`	Supply-air volumetric flow (m^3/s).
	Zone Temperature	`T_zone`	Zone air temperature.
	Zone RH	`RH_zone`	Zone relative humidity.
	Zone CO2	`CO2_zone_ppm`	Zone CO_2 concentration (ppm).
Sensor	RH Sensor – Supply	`RHsa`	Measures supply-air relative humidity.
	SA Temp Sensor	`Tsa`	Measures supply-air temperature downstream of coils.
	SA Flow Sensor	`senSA`	Measures supply-air flow delivered by the fan.
	Zone Temp Sensor	`TzoneSen`	Measures zone air temperature.
	Zone RH Sensor	`RHzones`	Measures zone relative humidity.
	Zone CO2 Sensor	`CO2zone`	Measures zone CO_2 concentration (ppm).
Air	Outdoor Air Node	`ambAir`	Ambient air state feeding the outdoor intake.
	Pre-Filter	`preFil`	First filtration stage; adds pressure drop.
	Mixing Tee / Box	`mixT`	Ideal mixing of outdoor air and return air.
	Final Filter	`finFil`	Final filtration before distribution to occupied space.
	Return-Air Path	`vRA_int`	Internal return-air manifold segment.
	Return-Air Filter	`filtRA`	Filters the return stream before mixing or exhaust.

Continued on next page

Domain	**Full Name**	`Abbrev`	**Description / Function**
Air/ Actuator	Outdoor Air Damper	`damOA`	Modulates outdoor-air fraction.
	Supply Fan	`fanSA`	Drives supply airflow through AHU to the zone.
	Return-Air Damper	`damRA`	Sets recirculation ratio/back-pressure.
	Exhaust-Air Damper	`damEA`	Sets return-air exhaust fraction to ambient.
	Exhaust Fan	`fanEA`	Extracts air to outside.
Load	Internal Gains	`intGains`	Injects occupant/equipment sensible/latent gains.
	Moisture Source	`moisture_occ`	Adds latent moisture generation in occupied periods.
	CO2 Source	`CO2source`	Adds CO_2 generation proportional to occupancy.
Water/ Actuator	CHW Pump	`pumpCW`	Circulates chilled water through coil.
	CHW Control Valve	`valCW`	Modulates CHW flow affecting `Tsa`.
Boundary	Weather Data Bus	`weaDat`	Provides outdoor conditions to the model.
Air/ Sensor	Outdoor Air Sensor	`senOA`	Measures outdoor-air flow/conditions.
Air-Water HX	Cooling Coil	`cooCoil`	Removes sensible/latent heat using CHW loop.
	Heating Coil	`heaCoil`	Adds sensible heat (disabled in cooling mode).
Room/ Air	Thermal Zone	`zone`	Single-zone volume with energy and IAQ balances.
Water	Expansion Vessel	`expVesEva`	Provides expansion volume/compliance.
	Return Tank	`conReturn`	Buffer on return side to smooth flow/temp.
Water/ Plant	Chiller	`chiller`	Provides chilled water to cooling coil.
Water/ Sensor	DP Sensor	`dp`	Measures chilled-water loop differential pressure.

2. Component-Level Modeling of the HVAC System

2.1 Boundary and Weather Data(Blue)

Humidity ratio và enthalpy. The weather module provides barometric pressure p_{atm}, dry-bulb temperature $T_{\text{db,out}}$, relative humidity ϕ_{out}, wind speed v_w, and solar irradiance G. From $(T_{\text{db,out}}, \phi_{\text{out}}, p_{\text{atm}})$ we obtain the humidity ratio ω_{out} and moist-air enthalpy h_{out} according to standard psychrometric relations [2]:

$$\omega_{\text{out}} = \frac{0.62198\, p_v}{p_{\text{atm}} - p_v}, \qquad h_{\text{out}} \approx c_{p,a}\, T_{\text{db,out}} + \omega_{\text{out}}(h_g - h_f), \tag{1}$$

where p_v is the partial pressure of water vapor obtained from standard saturation-pressure formulas, $c_{p,a}$ is the specific heat capacity of dry air, $(h_g - h_f)$ represents the latent heat of vaporization at the reference state, and the constant 0.62198 is the molecular-weight ratio of water vapor to dry air ($M_w/M_a = 18.016/28.966$).

First-order low-pass filter. To attenuate measurement noise, first-order filters are applied to weather inputs (time constant τ_f typically 3–6 s) [6]:

$$\tau_f\, \dot{x}_f = x - x_f, \qquad x \in \{T_{\text{db,out}}, \phi_{\text{out}}, \ldots\}. \tag{2}$$

Enthalpy-based economizer control. Hard limiters enforce damper feasibility and code minima $y_{\text{OA}}, y_{\text{EA}} \in [0, 1]$, with optional economizer logic [3]:

$$y_{\text{OA}} \geq y_{\min}\ (\text{occupied}), \qquad y_{\text{OA}} = \begin{cases} y_{\max}, & h_{\text{out}} < h_{\text{RA}} - \delta_h, \\ y_{\min}, & \text{otherwise}, \end{cases} \tag{3}$$

where δ_h is an enthalpy deadband. Solar and wind influence the envelope thermal model through the exogenous heat gain term Q_{env} (see Sec. C), enabling weather-aware load forecasting and feedforward compensation.

Identification/Calibration. Weather-station bias (e.g., sensor offsets) is estimated by regressing measured vs. reference data. The economizer deadband δ_h is tuned to avoid short-cycling under local climate variability.

Discrete-Time Implementation. For sample period T_s, the continuous filter [42, 24] (2) becomes:

$$x_f[k] = \alpha\, x_f[k-1] + (1 - \alpha)\, x[k], \qquad \alpha = e^{-T_s/\tau_f}. \tag{4}$$

2.2 Supply-Air Stream and Conditioning (Green)

Mixing and Pre-Filtration Outdoor and return air streams mix ideally at the supply-air inlet [2]. The supply-air mass flow is:

$$\dot{m}_{\text{SA}} = \dot{m}_{\text{OA}} + \dot{m}_{\text{RA}}, \tag{5}$$

and the mixed-air enthalpy and humidity ratio are given by mass-weighted averages:

$$h_{\mathrm{mix}} = \frac{\dot{m}_{\mathrm{OA}} h_{\mathrm{out}} + \dot{m}_{\mathrm{RA}} h_{\mathrm{RA}}}{\dot{m}_{\mathrm{SA}}}, \qquad \omega_{\mathrm{mix}} = \frac{\dot{m}_{\mathrm{OA}} \omega_{\mathrm{out}} + \dot{m}_{\mathrm{RA}} \omega_{\mathrm{RA}}}{\dot{m}_{\mathrm{SA}}}. \tag{6}$$

Pre-filters impose a pressure drop Δp_{pre} (added to the required fan static head), as well as a slowly drifting fouling factor $f_{\mathrm{foul}}(t)$ that can be estimated from trending data of Δp vs. volumetric flow rate $\dot{V}$ [2].

Cooling/Reheat Coils and SA Temperature Control The cooling coil removes both sensible and latent loads, while a reheat coil is enabled when humidity control requires decoupling temperature and moisture conditioning. The air-side energy balance across the coil pair [31] is:

$$\dot{m}_{\mathrm{air}}\, c_{p,a} \left(T_{\mathrm{mix}} - T_{\mathrm{sa}}\right) + \dot{m}_{\mathrm{air}}\, h_{\mathrm{lat}} \left(\omega_{\mathrm{mix}} - \omega_{\mathrm{sa}}\right) = Q_{\mathrm{coil,net}}, \tag{7}$$

with net cooling [15] defined as:

$$Q_{\mathrm{coil,net}} = Q_{\mathrm{cool}} - Q_{\mathrm{reheat}}. \tag{8}$$

A PI controller regulates the supply-air temperature T_{sa} to its setpoint $T_{\mathrm{sa,set}}$ [42]:

$$u_{\mathrm{val}} = \mathrm{sat}_{[0,\,1]}\left(K_P e_T + K_I \xi\right), \qquad \dot{\xi} = e_T, \qquad e_T = T_{\mathrm{sa,set}} - T_{\mathrm{sa}}. \tag{9}$$

Anti-windup is implemented via back-calculation with constant K_{aw} [41, 42]:

$$\dot{\xi} = e_T + K_{\mathrm{aw}} \left(u_{\mathrm{val}} - u_{\mathrm{val,raw}}\right). \tag{10}$$

When humidity exceeds allowable bounds, supervisory logic decreases $T_{\mathrm{sa,set}}$ (deep cooling) while enabling reheat, allowing latent removal without excessive space cooling.

Tuning. Select K_P and K_I to achieve a phase margin $\geq 45°$ and overshoot $< 5\%$. A practical engineering guideline is [2, 41]:

$$K_P \approx 0.3\, G(0), \qquad K_I \approx \frac{K_P}{\tau_{\mathrm{th}}}, \tag{11}$$

where $G(0)$ is the DC (steady-state) gain from valve position to supply-air temperature, τ_{th} is the dominant thermal time constant of the air-handling process, and the coefficient 0.3 is an empirical tuning factor that provides a conservative proportional gain, corresponding to roughly 30% of the open-loop gain margin. This ensures adequate damping and robustness while maintaining a fast settling response without overshoot.

Supply Fan, Ducts, and Delivery Assuming fan similarity laws, the power and pressure rise scale with volumetric flow as [2, 42]:

$$P_{\mathrm{fan}} \;\approx\; P_{\mathrm{nom}} \left(\frac{\dot{V}_{\mathrm{SA}}}{\dot{V}_{\mathrm{nom}}}\right)^3, \qquad \Delta p_{\mathrm{fan}} \;\approx\; \Delta p_{\mathrm{nom}} \left(\frac{\dot{V}_{\mathrm{SA}}}{\dot{V}_{\mathrm{nom}}}\right)^2. \tag{12}$$

where P_{nom} and Δp_{nom} are the nominal design power and static pressure rise at the rated volumetric flow $\dot{V}_{\mathrm{nom}}$, and $\dot{V}_{\mathrm{SA}}$ is the actual supply-air flow rate. According to turbomachinery affinity laws, the fan pressure rise scales with the square of the flow ratio, while the required shaft power scales with the cube. Consequently, a small reduction in fan speed significantly reduces electrical demand, making variable-speed drives (VSDs) an effective energy-saving strategy under part-load operation. The available static pressure Δp_{fan} must overcome duct friction and terminal device losses, while the fan control signal is further adjusted by the T_{SA} loop and supervisory ventilation logic to maintain adequate airflow and temperature delivery.

2.3 Thermal Zone and IAQ (Purple)

The zone is modeled as a well-mixed volume with effective thermal capacitance C_z and air volume V. Sensible and latent internal gains Q_{int} and $\dot{m}_{\text{gen},\text{H}_2\text{O}}$, as well as CO_2 emissions $\dot{m}_{\text{gen},\text{CO}_2}$, are occupancy-dependent. The sensible energy balance is [32]:

$$C_z \frac{dT_z}{dt} = \dot{m}_{\text{SA}}(h_{\text{SA}} - h_z) + Q_{\text{int}} + UA\,(T_{\text{amb}} - T_z) + Q_{\odot}, \tag{13}$$

where UA is the envelope conductance and $Q_{\odot}$ denotes solar and aperture gains.

Moisture dynamics follow a mass balance on the humidity ratio [2]:

$$V \frac{d\omega_z}{dt} = \dot{m}_{\text{SA}}(\omega_{\text{SA}} - \omega_z) + \dot{m}_{\text{gen},\text{H}_2\text{O}}, \tag{14}$$

and CO_2 concentration [2] evolves as:

$$V \frac{dc_{\text{CO}_2,z}}{dt} = \dot{m}_{\text{SA}}(c_{\text{SA}} - c_{\text{CO}_2,z}) + \dot{m}_{\text{gen},\text{CO}_2}. \tag{15}$$

Comfort and IAQ Constraints. Thermal comfort compliance is assessed using time-in-band metrics for T_z and ϕ_z. Indoor air quality (IAQ) control enforces

$$c_{\text{CO}_2,z} \leq c_{\max} \tag{16}$$

(e.g., $c_{\max} \approx 1000$ ppm during occupied periods).

Parameter Identification. The envelope conductance UA and capacitance C_z are identified from free-decay tests (night setback periods) using (13). Typical occupant CO_2 emission rates follow standards (e.g., 0.004–0.006 L/s CO_2 per person) and may be refined via Bayesian updates based on measured CO_2 transients.

Numerical Considerations. To maintain physically meaningful moisture and CO_2 states [33], we enforce non-negativity using:

$$\omega_z \leftarrow \max(0, \omega_z), \qquad c_{\text{CO}_2,z} \leftarrow \max(0, c_{\text{CO}_2,z}), \tag{17}$$

or employ positivity-preserving ODE integration schemes when discretizing (14)–(15).

2.4 Return and Exhaust Handling (Purple)

Return air passes through an internal flow path and filter with pressure drop Δp_{RA}. The return-air stream is split by the exhaust damper: a fraction $\dot{m}_{\mathrm{EA}}$ is exhausted by the exhaust fan, and the remainder $\dot{m}_{\mathrm{rec}}$ is recirculated [33]:

$$\dot{m}_{\mathrm{RA}} = \dot{m}_{\mathrm{EA}} + \dot{m}_{\mathrm{rec}}, \qquad 0 \le \dot{m}_{\mathrm{EA}} \le \dot{m}_{\mathrm{RA}}. \tag{18}$$

Supervisory logic coordinates the exhaust damper position y_{EA} and exhaust-fan command u_{fanEA} together with the outdoor-air damper y_{OA} to balance indoor air quality (IAQ) and energy efficiency. Rate limiters on $y_{\mathrm{RA/EA}}$ prevent abrupt pressure transients and ensure non-negative mass flow across all branches:

$$\dot{m} \ge 0.$$

Commissioning. Minimum ventilation fractions are verified by measuring the outdoor-air ratio using CO_2 or enthalpy balance. Damper end-stops must be adjusted to guarantee $y_{\min}$ during occupied mode to comply with ventilation requirements.

2.5 Chilled-Water Loop (Cyan)

The hydronic loop comprises chiller → pump → control valve → cooling coil → chiller, with an expansion vessel and a differential-pressure (DP) controller regulating Δp [15]. Coil heat transfer satisfies

$$Q_{\mathrm{coil}} = \dot{m}_{\mathrm{CHW}}\, c_{p,w}\, (T_{\mathrm{in}} - T_{\mathrm{out}}) = UA_{\mathrm{coil}}\, \Delta T_{\mathrm{lm}}, \tag{19}$$

where ΔT_{lm} is the log-mean temperature difference. The air-side approximation linking (19) and the air-side balance (Sec. C) holds when bypass and leakage are negligible.

Chiller Power and COP. Chiller electrical power relates to evaporator load [2] through:

$$P_{\mathrm{chiller}} = \frac{|Q_{\mathrm{eva}}|}{\mathrm{COP}}, \qquad Q_{\mathrm{eva}} \simeq Q_{\mathrm{coil}}. \tag{20}$$

The coefficient of performance depends on temperature lift and part load. A practical performance map [2] is:

$$\mathrm{COP}(T_{\mathrm{evap}}, T_{\mathrm{cond}}, \mathrm{PLR}) = a_0 + a_1 L + a_2 \mathrm{PLR} + a_3 L^2 + a_4 \mathrm{PLR}^2 + a_5 L \cdot \mathrm{PLR}, \tag{21}$$

where $L = T_{\mathrm{cond}} - T_{\mathrm{evap}}$ is the lift and PLR is the part-load ratio.

Pump Power. Under DP control, the pump power [2] is:

$$P_{\mathrm{pump}} \approx \frac{\dot{V}_{\mathrm{CHW}}\, \Delta p}{\eta_{\mathrm{pump}}}, \qquad \Delta p \approx \Delta p_{\mathrm{set}} \ \text{(DP controller)}. \tag{22}$$

Valve Authority and Stability. Control valve authority [2] is defined as:

$$N = \frac{R_{\mathrm{val}}}{R_{\mathrm{val}} + R_{\mathrm{rest}}}, \qquad N \in [0.3,\, 0.7]. \tag{23}$$

Small N yields sluggish response; excessively large N increases risk of oscillation. The DP setpoint is tuned to maintain N across operating conditions.

Freezing Protection. To prevent coil icing [2, 15],

$$T_{\text{coil,out}} \geq T_{\min}, \qquad \dot{m}_{\text{CHW}} \geq \dot{m}_{\min}. \tag{24}$$

Violations trigger controlled airflow reduction and valve opening while issuing alarms.

2.6 Control Architecture and Constraint Handling (Gray/Black)

The control system is organized hierarchically. Two inner feedback loops regulate supply-air temperature and hydronic differential pressure, while a supervisory layer manages ventilation/IAQ and occupancy-dependent behavior [2]:

$$\begin{aligned} \text{Temperature loop:} &\quad T_{\text{sa,set}} \xrightarrow{\text{PI}} (u_{\text{val}},\, u_{\text{fan}}) \;\to\; T_{\text{sa}}, \\ \text{IAQ supervisory loop:} &\quad (c_{\text{CO}_2,z},\, \phi_z,\, T_z) \xrightarrow{\text{logic}} (y_{\text{OA}},\, y_{\text{EA}},\, u_{\text{fanEA}}), \\ \text{Hydronic DP loop:} &\quad \Delta p_{\text{set}} \xrightarrow{\text{PI}} u_{\text{pump}} \;\to\; \Delta p. \end{aligned} \tag{25}$$

Bumpless transfer is enforced when reheat is enabled to prevent abrupt transitions. All actuators are saturated and slew-rate limited to ensure safe, physically feasible motion [2]:

$$u[k] = \text{sat}_{[0,1]}\Big(u[k-1] + \text{clip}\Big(\Delta u,\ -r_{\downarrow} T_s,\ r_{\uparrow} T_s\Big)\Big), \tag{26}$$

where $r_{\uparrow}$ and $r_{\downarrow}$ are the up/down rate limits and T_s is the sampling period. Sensor outputs are low-pass filtered prior to the PI blocks to break algebraic loops, and anti-windup follows (10).

Setpoint Coordination. When IAQ degrades (e.g., $c_{\text{CO}_2,z} \uparrow$), the supervisory layer temporarily increases y_{OA} (and, if needed, u_{fan}). Simultaneously, $T_{\text{sa,set}}$ may be reduced to increase latent removal capacity. Once IAQ recovers, the setpoints return to nominal with hysteresis to avoid short-cycling.

Integrated Process Flow. The full HVAC system comprises:

- Weather and boundary data modules defining external conditions.
- Outdoor air handling and mixing with return air for energy recovery.
- Cooling and heating coils conditioned by a PID-based temperature loop.
- Supply fan and filtration before delivery to the thermal zone.
- Zone sensing of T_z, ϕ_z, and $c_{\text{CO}_2,z}$ for IAQ control.
- Return/exhaust pathway regulating pressure balance and ventilation fraction.
- Hydronic loop (chiller–pump–valve–coil) supplying chilled water for heat extraction.

Time-based occupancy scheduling (e.g., 08:00–17:00) modulates $T_{\text{sa,set}}$, y_{OA}, and chiller/fan load. Indoor air quality is maintained in accordance with ASHRAE 62.1 by ensuring $c_{\text{CO}_2,z} \leq 1000$ ppm during occupied periods.

Overall, the supervisory layer ensures energy-efficient operation while maintaining zone thermal comfort, humidity, and ventilation air quality within prescribed limits.

2.7 Modeling Assumptions and Numerical Considerations

- **Air.** Low-Mach, ideal-gas moist air. Duct dynamics are lumped; leakage is neglected.
- **Water.** Chilled-water modeled as single-phase, incompressible flow. Density ρ_w and specific heat $c_{p,w}$ are treated as weakly temperature-dependent.
- **Mixing.** Perfect mixing at the supply-air mixing node; no thermal or humidity stratification upstream of the cooling coil.
- **Envelope.** Building envelope conduction is represented by a lumped conductance UA, with an optional solar/aperture gain term $Q_{\odot}$.
- **Occupancy.** Occupancy schedule is assumed known. An optional estimator may infer occupancy from CO_2 transients (inverse mass balance).
- **Numerics.** Use implicit integration schemes when stiff dynamics arise (e.g., deep dehumidification). Enforce non-negativity on mass flow $\dot{m}_{\text{flow}}$, humidity ratio ω, and CO_2 concentration c_{CO_2}. Add small ε regularization in denominators to avoid division by near-zero flows. Algebraic loops are avoided using sensor filtering and actuator rate limits.

2.8 Signals and Interfaces (Implementation View)

Measured Outputs.

$$\{\, T_{\text{sa}},\ \text{RH}_{\text{SA}},\ \dot{V}_{\text{SA}},\ T_z,\ \text{RH}_z,\ c_{\text{CO}_2,z},\ \Delta p_{\text{CHW}},\ T_{\text{coil,in}},\ T_{\text{coil,out}},\ \Delta p_{\text{filters}}\,\}.$$

Manipulated Inputs.

$$\{\, u_{\text{val}},\ u_{\text{fan}},\ y_{\text{OA}},\ y_{\text{EA}},\ u_{\text{fanEA}},\ u_{\text{pump}},\ \text{reheat\_enable}\,\}.$$

Setpoints / Disturbances.

$$\{\, T_{\text{sa,set}},\ \Delta p_{\text{set}},\ y_{\text{min}}(\text{occupied}),\ c_{\text{max}},\ \text{weather bus},\ \text{occupancy schedule/estimate}\,\}.$$

Alarms and FDD Hooks.

- **Filter fouling:** Increasing Δp vs. $\dot{V}$ trend on supply or return filters.
- **Damper misalignment:** Discrepancy between commanded and measured y_{OA} or y_{EA}.
- **Coil freezing risk:** $T_{\text{coil,out}} < T_{\text{min}}$.
- **Pump cavitation / instability:** Oscillatory Δp at low $\dot{V}_{\text{CHW}}$.

3. Time Series Forecasting with Deep Learning

3.1 Introduction

Time Series Forecasting (TSF) is a key area of research that aims to predict future states or trends from historical time-stamped data. In this study, we focus on short-term forecasting of outdoor air temperature. The primary objective is to design and evaluate deep learning models capable of accurately predicting temperature in the upcoming hours. These forecasts serve as input to smart HVAC (Heating, Ventilation, and Air Conditioning) systems to optimize their control strategies, improve energy efficiency, and maintain thermal comfort.

We use a univariate time series dataset extracted from an EPW (EnergyPlus Weather) file obtained from an online weather source. The file contains multiple weather-related variables; we selected five essential fields: `Year`, `Month`, `Day`, `Hour`, and `Dry Bulb Temperature`.

- **Target variable:** Dry Bulb Temperature (°C).
- **Temporal resolution:** Hourly.
- **Characteristics:** Exhibits meteorological seasonality (e.g., day–night cycle) and long-term trends. In addition, data recorded before 2023 may exhibit distribution shifts relative to current weather conditions, which must be considered when interpreting model performance and generalization.

3.2 Data Preprocessing Methodology

Data Normalization. Since deep learning models—especially those using sigmoid or tanh activations—are sensitive to data scale, normalization is essential. We employ the MinMaxScaler to map all temperature values into the range $[0, 1]$:

$$X_{\text{scaled}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}} \tag{27}$$

This transformation ensures faster and more stable convergence by preventing large gradient magnitudes.

Time Series Transformation. To transform the TSF task into a supervised learning problem, a sliding window technique is applied with a window length parameter `time_step = 3` (corresponding to 3 hour before).

The formulation is:

$$\begin{aligned} X_t &= \{\text{Temp}_{t-71}, \text{Temp}_{t-70}, \ldots, \text{Temp}_t\}, \\ Y_t &= \{\text{Temp}_{t+1}\} \end{aligned} \tag{28}$$

Choosing a 3-hour window allows the model to capture the fluctuations of Vietnam's weather while also reducing the input processing time of the entire system.

Data Partitioning. Temporal order is preserved to avoid data leakage. The dataset is split sequentially:

- **Training set:** First 70% of data
- **Validation set:** Next 15%
- **Test set:** Final 15%

The validation set is used for early stopping based on validation loss, while the test set is used only once for final performance evaluation.

3.3 Model Architectures

All models were trained with the Adam optimizer and the Mean Squared Error (MSE) loss, which are commonly used in regression problems. This section provides the theoretical background of each deep learning architecture used for temperature time series forecasting.

Long Short-Term Memory (LSTM). Long Short-Term Memory (LSTM) networks [9] are a special class of Recurrent Neural Networks (RNNs) that can capture long-term temporal dependencies. Unlike traditional RNNs, which struggle with vanishing or exploding gradients, LSTMs introduce a *cell state* that acts as a conveyor belt of information, regulated by three gates:

- **Forget gate (f_t):** decides which information from the previous cell state should be discarded.
- **Input gate (i_t):** determines which new information should be stored in the cell.
- **Output gate (o_t):** decides what information to output to the next hidden state.

The mathematical formulation [9] is as follows:

$$\begin{aligned}
f_t &= \sigma(W_f[h_{t-1}, x_t] + b_f)\,, \\
i_t &= \sigma(W_i[h_{t-1}, x_t] + b_i)\,, \\
\tilde{c}_t &= \tanh(W_c[h_{t-1}, x_t] + b_c)\,, \\
c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t, \\
o_t &= \sigma(W_o[h_{t-1}, x_t] + b_o)\,, \\
h_t &= o_t \odot \tanh(c_t).
\end{aligned} \tag{29}$$

Through this gating mechanism, the network learns which temporal patterns are relevant over long periods, making LSTMs particularly effective for meteorological forecasting, where temperature changes follow both short-term and long-term seasonal dynamics.

Convolutional Neural Network (CNN). Convolutional Neural Networks (CNNs) are typically used for spatial feature extraction, but in the one-dimensional (1D) setting, they are powerful tools for learning local temporal dependencies. A 1D CNN applies convolutional filters over sequential inputs, capturing local patterns such as abrupt temperature changes or diurnal oscillations.

Mathematically, a 1D convolution is defined as:

$$y_k = \sigma\left(\sum_{i=1}^{K} w_i \cdot x_{k+i-1} + b\right) \tag{30}$$

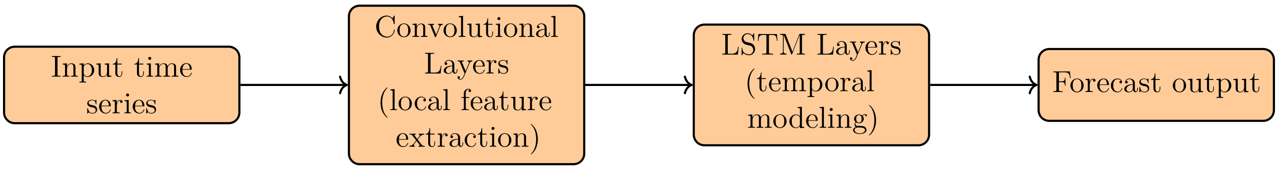
where x is the input sequence, w_i are filter weights, K is the kernel size, and σ is a nonlinear activation function.

After convolution, optional pooling operations (e.g., average or max pooling) can be used to reduce the temporal dimension and retain dominant features. The CNN's translational invariance allows it to detect patterns regardless of their position in time — ideal for identifying recurring phenomena such as daily temperature cycles.

CNN-LSTM Hybrid Model. The CNN-LSTM hybrid model integrates the feature extraction capability of CNNs with the sequential modeling strength of LSTMs. The CNN component processes raw input sequences to identify local temporal features (e.g., short-term trends), and the resulting feature maps are then fed into an LSTM network to learn long-term temporal dependencies.

This two-stage approach effectively decomposes the problem into:

$$\text{Local pattern extraction (CNN)} \rightarrow \text{Temporal modeling (LSTM).}$$


Input time series
Convolutional Layers (local feature extraction)
LSTM Layers (temporal modeling)
Forecast output

This architecture captures both short-term fluctuations and long-term patterns, improving forecasting stability in dynamic temperature environments.

3.4 Performance Evaluation Metrics

Model performance is evaluated on the inverse-transformed temperature predictions (in °C) to ensure physical interpretability. Let y_i denote the ground truth temperature at time i and $\hat{y}_i$ the corresponding model prediction. The following metrics are used to assess prediction accuracy and correlation:

Mean Absolute Error (MAE). MAE measures the average magnitude of prediction errors:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|. \tag{31}$$

Lower MAE values indicate smaller average deviations between predicted and actual temperatures.

Root Mean Squared Error (RMSE). RMSE penalizes larger errors more strongly than MAE due to the squared term:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}. \tag{32}$$

A lower RMSE reflects improved prediction stability and robustness.

Pearson Correlation Coefficient (R). The Pearson coefficient evaluates the linear correlation between predicted and observed temperature trends:

$$R = \frac{\sum_i (y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_i (y_i - \bar{y})^2}\sqrt{\sum_i (\hat{y}_i - \bar{\hat{y}})^2}}. \tag{33}$$

Values of R close to 1 indicate strong alignment in temporal variation patterns.

Nash–Sutcliffe Efficiency (NSE). NSE compares model performance to a baseline that always predicts the mean observed temperature:

$$\text{NSE} = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}. \tag{34}$$

An NSE close to 1 indicates excellent predictive capability, whereas values near or below 0 indicate performance no better than the mean baseline model.

Overall, low MAE and RMSE, together with high R and NSE values (approaching 1), indicate that the model accurately captures both the magnitude and temporal dynamics of indoor temperature fluctuations.

3.5 Performance and Evaluation Models

After training the deep learning models, we obtained evaluation results for the forecasting models. For each region, we selected a province or city representative of the climate, and based on the evaluation results, we selected forecasting models suitable for each region and different seasons. Although the forecasting results are not completely accurate, only capturing data trends, the error remains at a low average level and is not yet fully optimized. However, because the data has captured a stable trend, when combined with the DRL agent, we still hope to obtain the best possible results.

Based on the results obtained, the CNN model was chosen for Da Nang and Hanoi, while a hybrid model combining LSTM and CNN was selected for Ho Chi Minh City as the primary model for the predictive data used in the DDPG agent later on.

Table 11: Performance Metrics for Ha Dong Location (March, June, July)

Location	**Metric**	**LSTM**	**CNN**	**LSTM_CNN**
Ha Dong March 3	MAE	0.88	**0.3**	0.36
	MSE	1.18	**0.15**	0.22
	RMSE	1.09	**0.39**	0.47
	R	0.9	**0.98**	**0.98**
	NSE	0.8	**0.97**	0.96
Ha Dong June 6	MAE	0.45	**0.38**	**0.38**
	MSE	0.42	**0.3**	0.31
	RMSE	0.65	**0.55**	0.56
	R	**0.98**	**0.98**	**0.98**
	NSE	0.95	**0.96**	**0.96**
Ha Dong July 7	MAE	0.47	**0.29**	0.32
	MSE	0.38	**0.21**	0.24
	RMSE	0.61	**0.46**	0.49
	R	0.97	**0.98**	**0.98**
	NSE	0.94	**0.97**	0.96

Table 12: Performance Metrics for Nha Be Location (March, May, December)

Location	**Metric**	**LSTM**	**CNN**	**LSTM_CNN**
Nha Be March	MAE	0.34	**0.3**	1.35
	MSE	0.17	**0.14**	2.25
	RMSE	0.41	**0.37**	1.5
	R	**0.98**	**0.99**	0.83
	NSE	**0.97**	**0.97**	0.59
Nha Be May	MAE	0.54	0.58	**0.38**
	MSE	0.45	0.54	**0.27**
	RMSE	0.67	0.73	**0.52**
	R	0.97	**0.98**	**0.98**
	NSE	0.93	0.92	**0.96**
Nha Be December	MAE	0.32	0.29	**0.28**
	MSE	0.2	0.18	**0.17**
	RMSE	0.45	0.43	**0.42**
	R	**0.98**	**0.98**	**0.98**
	NSE	0.96	0.96	**0.96**

Table 13: Performance Metrics for Da Nang Location (March, July, November)

Location	**Metric**	**LSTM**	**CNN**	**LSTM_CNN**
Da Nang March	MAE	0.52	**0.27**	0.29
	MSE	0.38	0.16	**0.14**
	RMSE	0.62	**0.35**	0.37
	R	0.95	0.97	**0.98**
	NSE	0.91	**0.98**	0.97
Da Nang July	MAE	0.58	**0.25**	0.28
	MSE	0.45	**0.13**	0.16
	RMSE	0.67	**0.36**	0.40
	R	0.94	**0.99**	0.98
	NSE	0.89	**0.98**	0.97
Da Nang November	MAE	0.49	0.31	**0.26**
	MSE	0.35	**0.11**	0.15
	RMSE	0.59	0.40	**0.39**
	R	0.96	**0.99**	**0.99**
	NSE	0.92	**0.98**	**0.98**

4. Traditional Controller

Traditional control strategies remain widely used in HVAC systems due to their simplicity, stability, and strong practical reliability. Rule-Based Control (RBC) relies on predefined if-then logic or PID/hysteresis rules that are easy to configure and are already integrated into most Building Management Systems (BMS). As a result, RBC is simple to deploy, inexpensive to maintain, and provides predictable, transparent system behavior an important requirement in safety critical building environments. Meanwhile, Model Predictive Control (MPC) introduces a more advanced optimization based formulation that considers system dynamics and operational constraints, enabling improved comfort energy trade offs while remaining interpretable and structured. Overall, traditional controllers are highly compatible with existing HVAC infrastructures, require minimal modeling or data resources, and often deliver good enough energy efficiency and comfort performance in real-world building operations.

4.1 RBC (Rule-Based Control)

Rule-Based Control (RBC) is the earliest and most straightforward control strategy used in HVAC systems and many other industrial control problems. RBC consists of predefined conditional rules of the form IF–THEN, manually designed based on engineering heuristics, system operational knowledge, and safety constraints. Because the rules are fixed and context-insensitive, RBC is considered static and reactive, meaning the controller responds only to current measurements without forecasting or long-term planning.

In HVAC applications, RBC commonly employs bang-bang or two-position control with a predefined deadband to mitigate short cycling. This ensures that HVAC equipment does not continuously switch between ON and OFF states, which could lead to excessive mechanical wear and energy inefficiency. Mathematically, the cooling action at time t can be expressed as a hysteresis-based policy [7]:

$$a_t = \begin{cases} 1 \text{ (ON, Cooling)} & \text{if } T_{\text{zone},t} > T_{\text{setpoint}} + \delta, \\ 0 \text{ (OFF)} & \text{if } T_{\text{zone},t} < T_{\text{setpoint}} - \delta, \\ a_{t-1} & \text{otherwise.} \end{cases} \tag{35}$$

where

- $T_{\text{zone},t}$ is the measured indoor temperature at time t,
- T_{setpoint} is the desired user comfort temperature,
- δ is the deadband value used to avoid rapid on/off cycling,
- $a_t \in \{0, 1\}$ represents the on/off cooling action (1 = ON, 0 = OFF).

The primary limitation of RBC is its myopia: it only reacts to the current measured state $T_{\text{zone},t}$ without considering predictive factors such as future weather conditions, occupancy schedules, or electricity prices, and it cannot optimize long-term objectives like total energy consumption.

RBC is widely used due to its simplicity, low computational cost, transparency, and reliability. Building management technicians can easily understand and adjust the control logic without requiring advanced optimization or machine learning expertise. Additionally, RBC is straightforward to commission and is robust in environments where system dynamics are relatively consistent.

4.2 MPC (Model Predictive Control)

Model Predictive Control (MPC) is a more advanced control paradigm that addresses the limitations of RBC by incorporating a dynamic model of the system and explicitly optimizing control actions over a future time horizon. Instead of reacting only to the current temperature, MPC anticipates future conditions and selects actions that optimize performance according to a specified objective function.

A general discrete-time state-space model for HVAC thermal dynamics can be written as [38]:

$$x_{k+1} = f(x_k, u_k, d_k) \tag{36}$$

where

- x_k denotes the system state at step k (e.g., zone temperatures, wall thermal masses),
- u_k is the control action (e.g., cooling power, fan speed, damper position),
- d_k represents external disturbances (e.g., ambient temperature, solar heat gain, internal occupancy loads).

At each time step t, MPC solves an open-loop optimal control problem over a finite prediction horizon N. The goal is to find an optimal action sequence $\{u_t^*, \ldots, u_{t+N-1}^*\}$ that minimizes a predefined cost function J [38]:

$$\min_{u_t,\ldots,u_{t+N-1}} J = \sum_{k=0}^{N-1} L(x_{t+k}, u_{t+k}) + \Phi(x_{t+N}), \tag{37}$$

where

- $L(x_k, u_k)$ is the stage cost function, typically defined as:

$$L(x_k, u_k) = w_{\text{energy}} \cdot P_{\text{HVAC},k} + w_{\text{comfort}} \cdot \|T_{\text{zone},k} - T_{\text{setpoint},k}\|^2, \tag{38}$$

- $\Phi(\cdot)$ is the terminal cost promoting stability,
- N is the prediction horizon.

The term $\Phi(x_{t+N})$ serves as the terminal cost in the MPC formulation and plays a crucial role in guaranteeing closed-loop stability. Specifically, it penalizes undesirable system states at the end of the prediction horizon, encouraging the controller to drive the system toward a stable operating condition rather than merely optimizing short-term performance. Without this terminal cost, the controller may select actions that reduce immediate cost but lead to instability or oscillatory behavior in the long run.

A fundamental characteristic of MPC is the *receding horizon control* principle. Although MPC optimizes a sequence of future control actions over a prediction horizon N, only the first control input u_t^* is actually applied to the system. Once this action is executed, the system evolves to a new state x_{t+1}, which is then measured or estimated. The optimization problem is subsequently re-solved using the updated state as the new initial condition. This iterative process allows MPC to naturally incorporate feedback, meaning that if unexpected disturbances occur (e.g., sudden changes in occupancy or fluctuations in outdoor temperature), or if the model does not perfectly capture the real system dynamics, MPC can adjust future actions accordingly at each control step.

The main challenge of MPC lies in the need for an accurate prediction model $f(x, u, d)$ that captures system dynamics and disturbances. Moreover, solving the optimal control problem at each time step can be computationally expensive, particularly for large-scale HVAC systems or long prediction horizons, which may limit real-time applicability.

5. Data Collection and Pre-Processing

The input data used in this study consists of climatic and building parameters required for dynamic energy simulation. The primary climatic data source is the EnergyPlus Weather File (EPW), which provides a comprehensive representation of Vietnam's local weather conditions. The EPW file used here corresponds to a Typical Meteorological Year (TMYx) dataset, developed to statistically represent an average climatic year based on long-term hourly measurements.

5.1 Climatic Data Description

Unlike data from a single observation year, a TMY file aggregates meteorological observations from multiple years to form a representative weather profile for simulation purposes. This approach eliminates the influence of abnormal years and ensures consistent climatic boundary conditions for evaluating building energy performance. TMY datasets are widely adopted in simulation environments such as Modelica and EnergyPlus for analyzing building comfort and HVAC system efficiency.

While TMY refers to the type of weather dataset, EPW (EnergyPlus Weather) refers to the file format used to store such data. A TMY dataset is commonly distributed in EPW format, although EPW files may also contain actual single-year observations.

Each EPW file is organized into several structured sections that describe both site information and hourly weather observations:

- **Location Information:** Geographic and administrative identifiers, including city name, weather-station code, latitude, longitude, elevation, and time zone.
- **Design Conditions:** Extreme values of temperature and humidity used to compute heating and cooling loads.
- **Typical and Extreme Periods:** Representative weeks corresponding to average or extreme seasonal conditions (e.g., peak summer and winter).
- **Ground Temperatures:** Monthly soil temperature profiles at different depths, providing boundary data for ground-coupled heat transfer.
- **Hourly Data:** The core component of the dataset, containing 8,760 hourly records for an entire year. Each record includes:
  - Dry-bulb and dew-point temperatures,
  - Relative humidity,
  - Atmospheric pressure,
  - Global, direct, and diffuse solar radiation,
  - Wind speed and direction,
  - Total sky cover and opaque sky fraction.

5.2 Integration and Data Processing

The EPW dataset is imported into the **Modelica Buildings Library** as the environmental boundary condition for the HVAC simulation. These climatic variables dynamically drive the model, influencing outdoor air temperature, humidity ratio, and solar heat gains that directly affect building thermal loads.

Prior to simulation, the dataset undergoes a validation process to ensure numerical consistency and correct time-step alignment (hourly data from January 1 to December 31). Missing or abnormal entries, if any, are interpolated to maintain temporal continuity. The data is then converted into a Modelica-readable format and linked to the weather data block of the model.

5.3 Building Data

Alongside climatic data, the building model incorporates geometric and thermal parameters representing a standard Vietnamese office building. These include:

- Building envelope characteristics (walls, roof, glazing, insulation properties),
- Occupancy density and activity schedules,
- Internal heat gains from lighting, equipment, and occupants,
- HVAC design capacities and setpoint temperatures.

5.4 Application

By integrating the processed climatic and building datasets, the Modelica simulation accurately reflects the dynamic interaction between outdoor weather conditions and indoor environmental control. This provides a reliable foundation for analyzing energy consumption, system performance, and thermal comfort under Vietnam's tropical climate conditions.

6. Deep Reinforcement Learning Algorithm Design

6.1 State

The agent observes a composite state vector consisting of variables representing outdoor conditions, the conditioned zone, the HVAC system, and energy usage extracted from the FMU model.

The weather group (temperature, humidity, radiation, pressure, wind speed) reflects the external thermal load affecting cooling power demand. The zone variables (temperature, humidity, CO_2 concentration) describe indoor thermal comfort and air quality (IAQ). The air supply and energy variables (T_{SA}, RH_{SA}, P_{fan}, P_{fanEA}, P_{pump}, $P_{chiller}$, P_{heat}) allow the agent to perceive the system's instantaneous response and learn how to balance energy efficiency and occupant comfort. The occupancy signal represents the intensity of internal loads, while the previous action and short-term forecast help handle control delays and improve load prediction capability.

All state variables are normalized to a common scale of $[0, 1]$ or $[-1, 1]$ based on their physical limits, with outliers removed and missing data interpolated. Control variables are filtered through a safety layer to ensure actions remain within valid physical ranges and change smoothly, for example [2, 38]:

$$u_{\text{Fan}} \in [0.3, 1.0], \quad T_{\text{sp,chws}} \in [280, 288]\, K, \quad |\Delta u| < 0.05$$

This structure enables the agent to distinguish environmental noise from its own control effects, maintain simulation stability, and ensure that the learned control policy is practically feasible achieving both energy efficiency and occupant comfort with proper IAQ.

Table 14: Agent State Variables

Group	FMU Name	Description	Unit	Range	Rationale
Weather	TDryBul_in (T_{oa})	Outdoor dry-bulb temperature	K	$\approx$ 270–315	Affects sensible load, coil performance.
	relHum_in (RH_{oa})	Outdoor relative humidity	–	0–1	Influences latent load and dehumidification.
	pAtm_in (p_{atm})	Atmospheric pressure	Pa	$\approx 10^5$	Required for air-property calculations.
	HGloHor_in ($H_{\mathrm{glo,hor}}$)	Global horizontal irradiance	W/m^2	0–1000	Indirectly affects envelope/solar gains.
	HDifHor_in ($H_{\mathrm{dif,hor}}$)	Diffuse horizontal irradiance	W/m^2	0–500	Indirectly affects envelope/solar gains.
	winSpe_in (v_{wind})	Wind speed	m/s	0–15	Affects outdoor air exchange.
Zone	T_zone (T_{zone})	Zone (room) temperature	K	290–302	Primary thermal comfort target.
	RH_zone (RH_{zone})	Zone relative humidity	–	0.3–0.8	Humidity comfort.
	CO2_zone_ppm ($CO_{2,\mathrm{zone}}$)	Zone CO$_2$ concentration	ppm	400–2000+	Core IAQ indicator.
SA	T_SA (T_{SA})	Supply-air temperature	K	280–300	Reflects coil/control effectiveness.
	RH_SA (RH_{SA})	Supply-air relative humidity	–	0–1	Supply-air humidity control.
	Vdot_SA ($\dot{V}_{\mathrm{SA}}$)	Supply-air volumetric flow rate	m^3/s	$\sim$ 0.02–0.2	Tied to ventilation and fan energy.
Energy	P_fan (P_{fan})	Supply-fan power	W	0–$P_{\mathrm{fan,nom}}$	Direct energy component.
	P_fanEA (P_{fanEA})	Exhaust-fan power	W	0–$P_{\mathrm{EA,nom}}$	Exhaust-side ventilation load.
	P_pump (P_{pump})	Chilled-water pump power	W	0–$P_{\mathrm{pump,nom}}$	Hydronic energy; cubic relation with flow.

Group	FMU Name	Description	Unit	Range	Rationale
	`P_chiller` (P_{chiller})	Chiller equivalent power	W	≥ 0	Core cooling energy.
	`P_heat` (P_{heat})	Electric reheat coil power	W	≥ 0	Restores SA temp; heating mode only.
Other	`occupancy` (occ)	Proportion of people present in a room (input)	–	0–1	Gates comfort/IAQ and internal load.
Action History	`uFan_m1`, `uOA_m1`, `Tchws_m1` ($u_{\text{Fan}}^{t-1}, u_{\text{OA}}^{t-1}, T_{\text{chws}}^{t-1}$)	Previous-step actions	– / K	per actuator	Reduces oscillation; Markov augmentation.
Forecast	`T_forecast` (T_{forecast})	Outdoor temperature forecast	K	$\approx$ 270–315	Anticipates load/coil needs.

6.2 Action

The control action vector at time t is defined as:

$$a_t = \Big[uFan_t, \quad uOA_t, \quad u_{chiller}, \quad u_{heater}, \quad uFan_{EA}\Big]$$

- $uFan_t \in [0.1, 1.0]$ — Supply fan speed ratio (later converted to mass flow rate).
- $uOA_t \in [0.2, 1.0]$ — Outdoor air (OA) damper opening ratio.
- $u_{chiller} \in [0.0, 1.0]$ — Chiller capacity control signal.
- $u_{heater} \in [0.0, 1.0]$ — Heater capacity control signal.
- $uFan_{EA} \in [0.1, 1.0]$ — Exhaust fan speed ratio.

The five control variables correspond to the primary actuators influencing air side and water side thermal dynamics. The fan speed directly modulates the air supply rate and system power consumption, the outdoor air damper controls ventilation and CO$_2$ concentration, while the chilled-water temperature governs cooling capacity and coil efficiency. Together, these actions allow the agent to achieve a balance between energy efficiency, thermal comfort, and indoor air quality (IAQ).

The following table details the mapping between actions and FMU components:

Table 15: Action Space Definition

Action	**Symbol**	**FMU Mapping**	**Description**	**Range**	**Meaning & Constraints**
Fan speed control	$uFan$	$uFan \to uFan\_lim \to fanSA.m\_flow\_in$	Controls the air mass flow rate from the supply fan.	$[0.1, 1.0]$	$uFan\_lim$ is clipped to $[0.1, 1.0]$ and multiplied by $\dot{m}_{SA,nom}$ to yield $\dot{m}_{SA}$.
Outdoor air damper control	uOA	$uOA \to yOA\_lim \to damOA.y$	Controls the amount of natural outdoor air intake.	$[0.2, 1.0]$	yOA_lim is clipped (≥ 0.2); return/exhaust air are interpolated: $y_{RA,int} = 1 - uOA$, $y_{EA,int} = uOA$.
Chiller capacity control	$u_{chiller}$	$uChiller \to uChiller\_lim \to chiller.u$	Controls the chiller capacity (0–100%) for cooling the supply air through the cooling coil.	$[0.0, 1.0]$	$uChiller\_lim$ is clipped to $[0.0, 1.0]$; modulates effective evaporator setpoint: $T_{eva,eff} = u \cdot T_{eva,cold} + (1 - u) \cdot T_{eva,off}$.
Heater capacity control	u_{heater}	$uHeater \to uHeater\_lim \to heaterEnable.y \to heaCoil.u$	Controls the electric heater capacity (0–100%) for heating the supply air after the cooling coil.	$[0.0, 1.0]$	$uHeater\_lim$ is clipped to $[0.0, 1.0]$ and multiplied by flow fraction to enable heater only when sufficient airflow exists.
Exhaust fan speed control	$uFan_{EA}$	$uFanEA \to uFanEA\_lim \to fanEA.m\_flow\_in$	Controls the exhaust air mass flow rate to maintain pressure balance and ventilation.	$[0.1, 1.0]$	$uFanEA\_lim$ is clipped to $[0.1, 1.0]$ and multiplied by $\dot{m}_{SA,nom}$ to yield exhaust air mass flow rate.

6.3 Policy Specification

The reward function is designed to balance energy efficiency, indoor air quality (IAQ), and thermal comfort while ensuring smooth and stable system operation. Penalty terms are imposed when environmental or operational variables exceed predefined thresholds.

CO2 concentration During occupied periods, the zone CO$_2$ concentration $C_{\mathrm{CO_2}}$ should remain below the acceptable limit $\beta_{\mathrm{CO_2}}$. For every incremental increase of $\Delta\beta_{\mathrm{CO_2}}$ exceeding this limit, a penalty is applied [7]:

$$P_{\mathrm{CO_2}} = \begin{cases} \alpha_{\mathrm{CO_2}} \left\lfloor \dfrac{C_{\mathrm{CO_2}} - \beta_{\mathrm{CO_2}}}{\beta_{\mathrm{CO_2,scale}}} \right\rfloor, & \text{if occupied and } C_{\mathrm{CO_2}} > \beta_{\mathrm{CO_2}}, \\ 0, & \text{otherwise.} \end{cases} \tag{39}$$

Temperature comfort. The zone air temperature T_z is required to stay within the comfort range $[\beta_{T,\mathrm{low}}, \beta_{T,\mathrm{high}}]$. Temperatures outside this interval incur a penalty proportional to the deviation from the nearest comfort boundary:

$$P_T = \begin{cases} \alpha_T \left| T_z - T_{\mathrm{ref}} \right|, & \text{if } T_z \notin \left[\beta_{T,\mathrm{low}}, \beta_{T,\mathrm{high}}\right], \\ 0, & \text{otherwise,} \end{cases} \tag{40}$$

where $T_{\mathrm{ref}} \in \{\beta_{T,\mathrm{low}}, \beta_{T,\mathrm{high}}\}$ denotes the violated comfort limit.

Relative humidity. Relative humidity RH should remain below the acceptable threshold β_{RH}. Exceeding this threshold yields:

$$P_{RH} = \begin{cases} \alpha_{RH}, & \text{if } RH > \beta_{RH}, \\ 0, & \text{otherwise.} \end{cases} \tag{41}$$

Smoothness of control actions. To reduce actuator wear and maintain system stability, abrupt control variations are penalized (Eq. 42) or a continuous regularization term (Eq. 43).:

$$P_{\mathrm{smooth}} = \alpha_{\mathrm{smooth}} \cdot \mathbf{1}(|\Delta u| > \Delta\beta_u) \tag{42}$$

$$P_{\mathrm{smooth}} = \alpha_{\mathrm{smooth}} \cdot |\Delta u| \tag{43}$$

Energy consumption. $P_{\mathrm{total}}(t)$ denotes the aggregated instantaneous HVAC power consumption, including chillers, pumps, supply and exhaust fans, and reheat coils, and the interval $\left[\beta_{P,\mathrm{low}}, \beta_{P,\mathrm{high}}\right]$ defines the acceptable operating range for power consumption.

$$P_E = \begin{cases} \alpha_E \left(P_{\mathrm{total}}(t) - \beta_{P,\mathrm{high}}\right), & \text{if } P_{\mathrm{total}}(t) > \beta_{P,\mathrm{high}}, \\ \alpha_E \left(\beta_{P,\mathrm{low}} - P_{\mathrm{total}}(t)\right), & \text{if } P_{\mathrm{total}}(t) < \beta_{P,\mathrm{low}}, \\ 0, & \text{otherwise.} \end{cases} \tag{44}$$

These β parameters represent comfort and operational thresholds empirically derived from thermal comfort studies and IAQ standards relevant to the building's climatic context. They ensure that the reinforcement learning agent learns to maintain comfort and air quality within acceptable limits while optimizing energy efficiency.

Overall penalty and instantaneous reward. The overall penalty at time step t is defined as the additive combination of individual penalty components addressing indoor air quality, thermal comfort,energy consumption, and control smoothness:

$$P_t = P_{\mathrm{CO_2}} + P_T + P_{RH} + P_{\mathrm{smooth}} + P_E \tag{45}$$

This additive formulation provides a transparent and interpretable reward structure,allowing the contribution of each constraint to be independently tuned via its corresponding weighting factor. The instantaneous reinforcement learning reward is then defined as the negative of the total penalty, encouraging the agent to minimize constraint violations while achieving energy-efficient and comfort-compliant control.

6.4 Reward system

The objective of the Deep Reinforcement Learning (DRL) agent is to minimize the overall energy consumption while maintaining indoor thermal comfort, humidity, and air quality (IAQ) within acceptable ranges. The total reward R_t at each timestep is formulated as a weighted sum of sub-rewards associated with energy usage, comfort, and control smoothness.

$$\begin{aligned} R_t = &- \alpha_E R_E(t+1) - \alpha_T f(T_{\mathrm{zone}}(t+1)) - \alpha_{RH} f(\phi_{\mathrm{za}}(t+1)) \\ &- \alpha_{CO_2} f(CO_{2,\mathrm{za}}(t+1)) - \alpha_{\mathrm{smooth}} f(u(t+1)) \end{aligned} \tag{46}$$

where

$$\begin{aligned} R_E(t) &= P_{\mathrm{fan}}(t) + P_{\mathrm{fanEA}}(t) + P_{\mathrm{pump}}(t) + P_{\mathrm{chiller}}(t) + P_{\mathrm{heat}}(t), && (47)\\ f(T_{\mathrm{zone}}(t)) &= [T_{\mathrm{zone}}(t) - \beta_{T,\mathrm{high}}]_+ + [\beta_{T,\mathrm{low}} - T_{\mathrm{zone}}(t)]_+, && (48)\\ f(\phi_{\mathrm{za}}(t)) &= [\phi_{\mathrm{za}}(t) - \beta_{RH}]_+, && (49)\\ f(CO_{2,\mathrm{za}}(t)) &= \begin{cases} 0, & \text{if } CO_{2,\mathrm{za}}(t) < \beta_{CO_2}, \\ 1, & \text{if } CO_{2,\mathrm{za}}(t) \geq \beta_{CO_2}, \end{cases} && (50)\\ f(u(t)) &= (\Delta\tilde{u}_{\mathrm{Fan}})^2 + (\Delta\tilde{u}_{\mathrm{OA}})^2 + (\Delta\tilde{T}_{\mathrm{chws}})^2. && (51) \end{aligned}$$

The total reward consists of an energy term, multiple comfort-related penalties (activated only when the space is occupied), and a smoothness penalty to prevent abrupt control actions.

Energy Consumption The energy consumption term penalizes the instantaneous total power usage of the HVAC system, encouraging the agent to minimize energy consumption:

$$R_E = P_{\mathrm{fan}}(t) + P_{\mathrm{fanEA}}(t) + P_{\mathrm{pump}}(t) + P_{\mathrm{chiller}}(t) + P_{\mathrm{heat}}(t) \tag{52}$$

Each component is defined as follows:

- **Supply fan power:**

$$P_{\mathrm{fan}} = P_{\mathrm{fan,nom}} \left(\frac{\dot{m}_{SA}}{\dot{m}_{SA,\mathrm{nom}}} \right)^3 \tag{53}$$

- **Exhaust fan power:** modeled identically to the supply fan:

$$P_{\text{fanEA}} = P_{\text{fanEA,nom}} \left(\frac{\dot{m}_{EA}}{\dot{m}_{EA,\text{nom}}} \right)^3 \tag{54}$$

- **Chilled-water pump power:**

$$P_{\text{pump}} = P_{\text{pump,nom}} \left(\frac{\dot{m}_{CW}}{\dot{m}_{CW,\text{nom}}} \right)^3 \tag{55}$$

- **Chiller power:**

$$P_{\text{chiller}} = \frac{|\dot{Q}_{Eva}|}{\text{COP}_{\text{chiller}}} \tag{56}$$

- **Reheat coil power:**

$$P_{\text{heat}} = \max(0, \dot{Q}_{\text{heater}}) \tag{57}$$

By aggregating these components, R_{E} captures the complete electrical demand of the HVAC system, including cooling, heating, and ventilation subsystems. A larger coefficient α_E places stronger emphasis on energy efficiency, driving the DRL agent to discover control strategies that optimally balance comfort and total power consumption.

Thermal Comfort

$$T_c(t) = T_{\text{zone}}(t) - 273.15 \tag{58}$$

$$f\Big(T_{\text{zone}}(t)\Big) = \Big([T_c(t) - \beta_{T,\text{high}}]^+ + [\beta_{T,\text{low}} - T_c(t)]^+ \Big) \tag{59}$$

This component enforces the thermal comfort range when the zone is occupied. Deviations above $\beta_{T,\text{high}}$ (upper comfort threshold) or below $\beta_{T,\text{low}}$ (lower comfort threshold) are penalized, guiding the controller to maintain acceptable indoor temperature.

Humidity Comfort

$$f\Big(\phi_{\text{za}}(t)\Big) = [\, \phi_{\text{za}}(t) - \beta_{RH} \,]^+ \tag{60}$$

$\phi_{\text{za}}(t)$ is the relative humidity of the zone air at time t, and β_{RH} is the upper humidity threshold. The operator $[x]^+ = \max(x, 0)$ applies a penalty only when $\phi_{\text{za}}(t)$ exceeds β_{RH}.

Indoor Air Quality (CO$_2$)

$$f\Big(CO_{2,\text{za}}(t)\Big) = \frac{[CO_{2,\text{za}}(t) - \beta_{CO_2}]^+}{\beta_{CO_2,\text{scale}}} \tag{61}$$

This term discourages CO$_2$ concentrations above β_{CO_2}, promoting sufficient outdoor air intake and ventilation quality, especially under high occupancy.

Combo Penalty This penalty handles particularly undesirable situations where temperature and humidity are jointly outside the comfort zone in opposite directions:

$$P_{\text{combo}} = \begin{cases} 2.0, & \text{if } T_{z,t} < T_{\text{low}} \text{ and } \mathrm{RH}_{z,t} > \mathrm{RH}_{\text{high}}, \\ 1.5, & \text{if } T_{z,t} > T_{\text{high}} \text{ and } \mathrm{RH}_{z,t} < \mathrm{RH}_{\text{low}}, \\ 0, & \text{otherwise.} \end{cases} \tag{62}$$

Here:

- $T_{z,t}$ is the zone air temperature at timestep t (°C).
- $\mathrm{RH}_{z,t}$ is the zone relative humidity at timestep t (fraction: 0–1).
- $T_{\text{low}}, T_{\text{high}}$ are the temperature comfort bounds.
- $\mathrm{RH}_{\text{low}}, \mathrm{RH}_{\text{high}}$ are the humidity comfort bounds.

A higher penalty (2.0) is applied in cold-humid conditions (low T, high RH), which are perceived as clammy and uncomfortable, while a medium penalty (1.5) is applied in hot-dry conditions (high T, low RH), which can cause dryness and discomfort.

Action Extreme Penalty To prevent the agent from repeatedly selecting actions at the boundaries (near 0 or 1), which may be unsafe for equipment or lead to unstable operation:

$$P_{\text{extreme}} = \lambda_{\text{extreme}} \cdot \sum_{i=1}^{n_{\text{action}}} \nVdash_{(u_i < \epsilon_{\text{low}} \text{ or } u_i > \epsilon_{\text{high}})} \tag{63}$$

where:

- $\lambda_{\text{extreme}} = 0.2$ is the penalty weight.
- $n_{\text{action}} = 5$ is the number of action dimensions $(u_{\text{Fan}}, u_{\text{OA}}, u_{\text{Chiller}}, u_{\text{Heater}}, u_{\text{FanEA}})$.
- $\epsilon_{\text{low}} = 0.05$ and $\epsilon_{\text{high}} = 0.95$ define the extreme action region.
- $\nVdash_{(\cdot)}$ is the indicator function, returning 1 if the condition is true and 0 otherwise.

The penalty is accumulated for each action component that violates the bounds, encouraging the agent to keep actions within $[0.05, 0.95]$ for safer operation.

Action Change Bonus This bonus encourages the agent to adjust its actions in a reasonable way, avoiding being stuck at a nearly constant policy:

$$B_{\text{change}} = \begin{cases} \lambda_{\text{change}}, & \text{if } \epsilon_{\text{change,low}} < \Delta \bar{u}_t < \epsilon_{\text{change,high}}, \\ -0.1, & \text{if } \Delta \bar{u}_t < \epsilon_{\text{stuck}}, \\ 0, & \text{otherwise,} \end{cases} \tag{64}$$

where the mean action change is:

$$\Delta \bar{u}_t = \frac{1}{n_{\text{action}}} \sum_{i=1}^{n_{\text{action}}} |u_{i,t} - u_{i,t-1}| \tag{65}$$

Parameters:

- $\lambda_{\text{change}} = 0.3$ is the positive bonus.
- $\epsilon_{\text{change,low}} = 0.05$ and $\epsilon_{\text{change,high}} = 0.3$ define the desirable change range.
- $\epsilon_{\text{stuck}} = 0.02$ is the threshold for detecting a stuck policy (less than 2% change).
- $\Delta \bar{u}_t$ measures the average action change between two consecutive timesteps.

A positive bonus is granted when the action change lies within a reasonable range (5–30%), while a small negative penalty is applied if the change is very small, indicating that the agent might be stuck.

Comfort Stability Bonus To encourage maintaining comfort conditions over longer periods instead of achieving comfort only temporarily:

$$B_{\text{stability}} = \begin{cases} \lambda_{\text{comfort}} + \lambda_{\text{stability}}, & \text{if } C_t = 1 \text{ and } n_{\text{comfort}} \geq n_{\text{threshold}}, \\ \lambda_{\text{comfort}}, & \text{if } C_t = 1 \text{ and } n_{\text{comfort}} < n_{\text{threshold}}, \\ 0, & \text{if } C_t = 0, \end{cases} \tag{66}$$

where the comfort indicator C_t is defined as:

$$C_t = \nVdash_{(T_{\text{low}} \leq T_{z,t} \leq T_{\text{high}} \text{ and } \text{RH}_{\text{low}} \leq \text{RH}_{z,t} \leq \text{RH}_{\text{high}} \text{ and } \rho_t \geq \rho_{\text{threshold}})} \tag{67}$$

Parameters:

- $\lambda_{\text{comfort}} = 2.0$ is the basic comfort bonus.
- $\lambda_{\text{stability}} = 0.5$ is the additional bonus for long-term stability.
- n_{comfort} is the number of consecutive timesteps with $C_t = 1$.
- $n_{\text{threshold}} = 10$ is the minimum number of consecutive comfort steps to earn the stability bonus (2.5 hours at $\Delta t = 15$ minutes).
- ρ_t is the occupancy level at timestep t.
- $\rho_{\text{threshold}} = 0.3$ is the minimum occupancy level to consider comfort relevant.

A higher bonus is given when comfort is maintained for at least 10 consecutive timesteps during occupied periods, promoting stable rather than oscillatory control.

Temperature Stability Bonus An additional bonus rewards temperature stability between consecutive timesteps:

$$B_{T,\text{stable}} = \begin{cases} \lambda_{T,\text{stable}}, & \text{if } |T_{z,t} - T_{z,t-1}| < \delta_T \text{ and } T_{\text{low}} \leq T_{z,t} \leq T_{\text{high}}, \\ 0, & \text{otherwise.} \end{cases} \tag{68}$$

Here:

- $\lambda_{T,\text{stable}} = 0.3$ is the stability bonus.
- $\delta_T = 0.5$ °C is the allowable temperature variation between timesteps.
- $T_{z,t-1}$ is the zone air temperature at the previous timestep.

This term encourages the agent to keep temperature variations small (below 0.5 °C) while remaining within the comfort band, avoiding rapid fluctuations that could disturb occupants.

Updated Total Reward Function The overall reward function is updated to include the non-hierarchical components:

$$R_t = R_{\text{base}} - P_{\text{combo}} - P_{\text{extreme}} + B_{\text{change}} + B_{\text{stability}} + B_{T,\text{stable}} \tag{69}$$

where R_{base} is the base reward from Eq. (46):

$$R_{\text{base}} = 2.0 - (C_T + C_{\text{RH}} + C_E + P_{\text{smooth}} + P_{\text{severe}}) \tag{70}$$

The final reward is clamped to the interval $[R_{\min}, R_{\max}] = [-10.0, 5.0]$:

$$R_t^{\text{final}} = \max\left(R_{\min}, \min\left(R_{\max}, R_t\right)\right) \tag{71}$$

to ensure stable training and prevent gradient explosion during learning.

Action Smoothness (Regularization)

$$f\Big(u(t)\Big) = \Big[(\Delta\tilde{u}_{\text{Fan}}(t))^2 + (\Delta\tilde{u}_{\text{OA}}(t))^2 + (\Delta\tilde{T}_{\text{chws}}(t))^2\Big] \tag{72}$$

where

$$\Delta\tilde{u}_{\text{Fan}}(t) = \tilde{u}_{\text{Fan}}(t) - \tilde{u}_{\text{Fan}}(t-1), \quad \Delta\tilde{u}_{\text{OA}}(t) = \tilde{u}_{\text{OA}}(t) - \tilde{u}_{\text{OA}}(t-1), \quad \Delta\tilde{T}_{\text{chws}}(t) = \tilde{T}_{\text{chws}}(t) - \tilde{T}_{\text{chws}}(t-1).$$

Here, each Δ term represents the discrete change in the corresponding control input between two consecutive timesteps, capturing the smoothness of the control signal over time. This formulation penalizes rapid fluctuations in the fan speed, outdoor-air ratio, and chilled-water setpoint to ensure actuator longevity and stable HVAC dynamics.

Coefficient and Parameter Settings

$$\alpha_E = 4.0, \quad \alpha_T = 5.0, \quad \alpha_{RH} = 2.5, \quad \alpha_{CO2} = 0.001, \quad \alpha_{\text{smooth}} = 0.001 \tag{73}$$

In the case that someone is in the room, all variables are normalized to a set of weights as shown above, obtained from testing the DRL agent with a hierarchical reward system in Table 18. Regarding the weighting of CO_2 and smoothness, a very small weight will be established, as these two factors, across multiple tests, do not affect the performance of the models, ensuring arithmetic stability during training. The overall design allows the DRL agent to adapt by balancing energy efficiency and user comfort, responding flexibly to changes in user numbers and weather conditions in Vietnam's hot and humid climate.

Especially when no one is in the room, we set a new weighting system that prioritizes penalizing wasted energy and reduces the importance of temperature and humidity.

$$\alpha_E = 6.0, \quad \alpha_T = 1.5, \quad \alpha_{RH} = 0.5, \quad \alpha_{CO2} = 0.001, \quad \alpha_{\text{smooth}} = 0.001 \tag{74}$$

Table 16: Comfort and IAQ threshold parameters for reward and policy

Symbol	Meaning	Typical Value
$\beta_{T,high}$	Upper temperature comfort limit	28°C
$\beta_{T,low}$	Lower temperature comfort limit	25.5°C
β_{RH}	Maximum acceptable relative humidity	0.70
β_{CO2}	CO_2 concentration threshold (ppm)	1000 ppm
$\beta_{CO2,scale}$	Scaling factor for normalization	500

Constrains Selection Rationale The specific threshold parameters (β) defined in Table 11 represent a strategic balance between energy conservation and occupant satisfaction, tailored specifically to Vietnam's tropical hot-humid climate context. Recent studies [16] [17] [10] [23] show that Vietnamese people have a slightly higher tolerance for weather conditions than the world average because Vietnam is located near the equator, has a tropical monsoon climate, and high humidity. Based on various articles, it can be seen that Vietnamese people feel comfortable with temperatures ranging from 24°C to 30°C and humidity from 40% to 90%. However, given the severe climate change, the melting ice at the poles causing alarming weather changes, and the alarming levels of fine dust pollution in major cities like Hanoi, we have lowered the comfortable temperature ranges in the rooms. Specifically, the temperature will be maintained between 25.5°C and 28°C degrees Celsius, and the humidity between 40% and 70%. This ensures that the HVAC system operates effectively to remove fine dust, adapt to the temperature better, and ensure good respiration for workers. The selection criteria are grounded in established standards:

- **Thermal Comfort ($\beta_T \in [25.5, 28.0]$°C):** This range is selected based on the *Adaptive Thermal Comfort* model outlined in ASHRAE Standard 55 [2] and aligned with TCVN 5687:2010 (Vietnam Building Code) [21].

In tropical climates, occupants are acclimatized to higher ambient temperatures. By elevating the upper comfort limit to 28°C (compared to the conventional 24°C setpoint), the cooling load on the chiller is significantly reduced without compromising perceived comfort. This wider deadband provides the DRL agent with greater flexibility to exploit passive coasting strategies for energy savings.

- **Relative Humidity ($\beta_{RH} = 70\%$):** High humidity is a critical challenge in Vietnam. The 70% threshold is derived from ASHRAE 55 recommendations to control latent heat.

  Maintaining RH below this level is essential to prevent mold growth, ensuring hygiene, and avoiding the sensation of "stuffiness" that degrades perceived air quality, even at comfortable temperatures.

- **Indoor Air Quality ($\beta_{CO2} = 1000$ ppm):** This limit follows ASHRAE Standard 62.1 [2] for acceptable indoor air quality. It serves as a safety constraint; minimizing energy (by reducing ventilation) is prioritized only as long as CO_2 levels remain safe. Exceeding 1000 ppm triggers a penalty to force outdoor air intake, ensuring occupant health and cognitive performance take precedence over energy savings.

6.5 RL Algorithms

DDPG (Deep Deterministic Policy Gradient) Deep Deterministic Policy Gradient (DDPG) is a model-free, off-policy, Actor–Critic RL algorithm. It was developed to handle continuous action spaces, a common requirement for control problems like HVAC (e.g., setting a precise temperature setpoint such as 25.5 °C). DDPG learns a deterministic policy $a = \mu(s|\theta^{\mu})$ instead of a stochastic one.

DDPG maintains two pairs of neural networks (four networks in total):

- **Actor** ($\mu(s|\theta^{\mu})$): the policy network, which directly maps a state s to an action a.
- **Critic** ($Q(s, a|\theta^{Q})$): the value network, which learns the Q-value to evaluate the pair (s, a) proposed by the Actor.
- **Target Actor** ($\mu'(s|\theta^{\mu'})$) and **Target Critic** ($Q'(s, a|\theta^{Q'})$): two target networks used for stabilization.

The update process also uses Experience Replay. The Critic network is updated by minimizing the MSE loss, but the target value y is calculated using the action from the target Actor μ':

$$y = r + \gamma\, Q'\Big(s', \mu'(s'; \theta^{\mu'}); \theta^{Q'}\Big) \tag{75}$$

$$L(\theta^{Q}) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\left[\Big(y - Q(s, a; \theta^{Q})\Big)^2\right] \tag{76}$$

The Actor network is updated using the *policy gradient.* The Actor's goal is to find an action a that maximizes $Q(s, a)$. Therefore, the Actor is updated in the direction of the gradient of the Q-value (provided by the Critic) with respect to the action, based on the DPG theorem and the chain rule:

$$\nabla_{\theta^\mu} J \approx \mathbb{E}_{s\sim\mathcal{D}} \left[\nabla_a Q(s, a; \theta^Q)\Big|_{a=\mu(s)} \nabla_{\theta^\mu} \mu(s; \theta^\mu) \right] \tag{77}$$

Here, $\nabla_a Q(\cdot)$ is the gradient from the Critic (indicating how the action should change to increase Q), and $\nabla_{\theta^\mu} \mu(\cdot)$ is the gradient of the Actor (indicating how the parameters θ^μ should change to produce that action).

The target networks are *soft updated* as:

$$\theta' \leftarrow \tau\theta + (1-\tau)\theta' \tag{78}$$

where $\tau \ll 1$, ensuring smooth convergence and stability during training.

Algorithm 1 DDPG-based HVAC Control Policy Learning

```
Randomly initialize critic $Q(s, a \mid \theta^Q)$ and actor $\mu(s \mid \theta^\mu)$ with weights $\theta^Q$ and $\theta^\mu$
Initialize target networks $Q'(s, a \mid \theta^{Q'})$ and $\mu'(s \mid \theta^{\mu'})$ with $\theta^{Q'} \leftarrow \theta^Q$, $\theta^{\mu'} \leftarrow \theta^\mu$
Initialize replay buffer $\mathcal{D}$

for episode = 1 to $N_{\text{episodes}}$ do
  Initialize Ornstein–Uhlenbeck process $\mathcal{N}_t$ for action exploration
  Initialize FMU-based environment and obtain initial observation $s_1$
  for time step $t = 1$ to $T_{\max}$ do
    Select action $a_t = \mu(s_t \mid \theta^\mu) + \mathcal{N}_t$
    Apply $a_t$ to FMU, simulate one control interval $\Delta t$
    Observe reward $r_t$ and next state $s_{t+1}$
    Store transition $(s_t, a_t, r_t, s_{t+1})$ in replay buffer $\mathcal{D}$
    Sample random minibatch of $N$ transitions $(s_i, a_i, r_i, s'_i)$ from $\mathcal{D}$
    Compute target value:
        $y_i = r_i + \gamma Q'\Big(s'_i, \mu'(s'_i \mid \theta^{\mu'}) \mid \theta^{Q'}\Big)$
    Update critic by minimizing:
        $L(\theta^Q) = \frac{1}{N} \sum_i \Big(y_i - Q(s_i, a_i \mid \theta^Q)\Big)^2$
    Update actor using the sampled policy gradient:
        $\nabla_{\theta^\mu} J \approx \frac{1}{N} \sum_i \nabla_a Q(s, a \mid \theta^Q)\Big|_{s=s_i,\, a=\mu(s_i)} \nabla_{\theta^\mu} \mu(s \mid \theta^\mu)\Big|_{s=s_i}$
    Soft-update target networks:
        $\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'}, \quad \theta^{\mu'} \leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'}$
  end for
end for
```

The DDPG agent consists of four key components:

1. Actor Network: generates a continuous action (e.g., control signal or temperature setpoint) for the current state.

2. Critic Network: evaluates how good this action is by estimating its Q-value.

3. Replay Buffer: stores past experiences (s, a, r, s') to break temporal correlations, enabling more stable learning.

4. Target Networks: slowly updated copies of the Actor and Critic that provide stable targets for learning, preventing feedback oscillations.

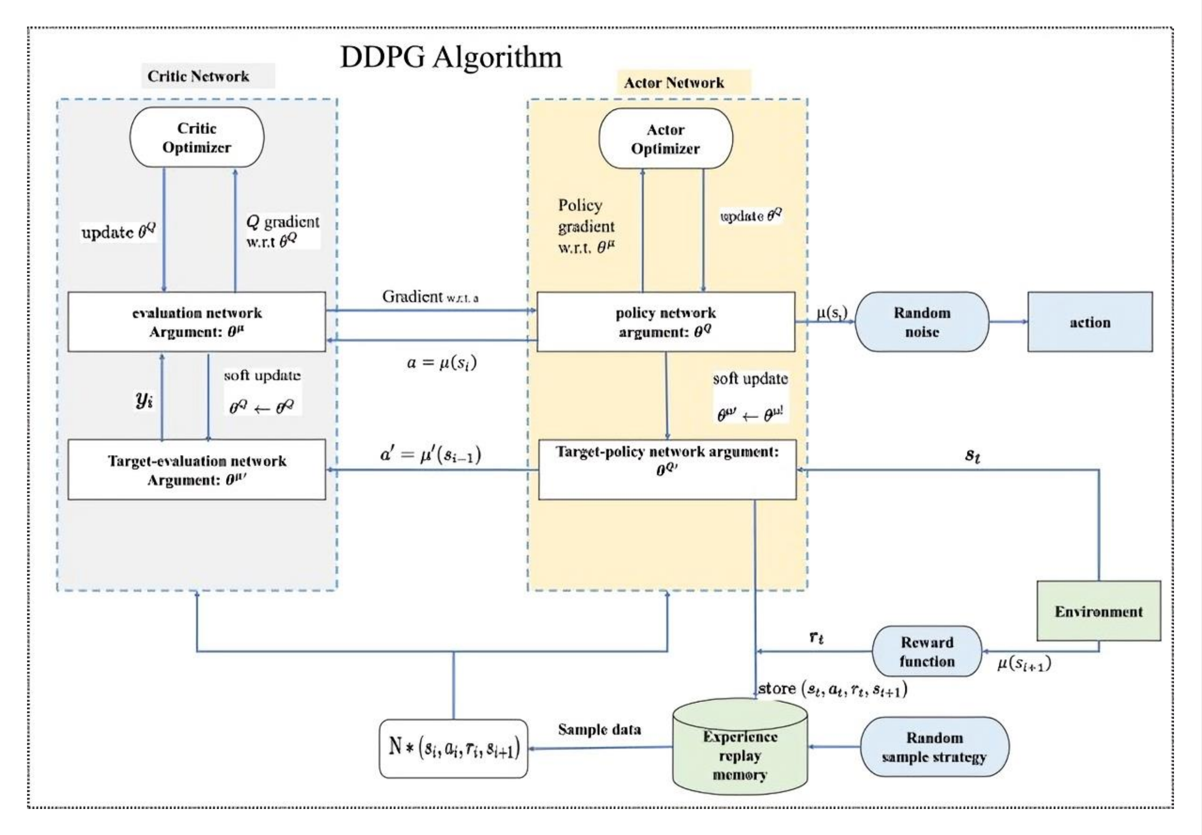

DDPG Algorithm
Critic Network
Critic Optimizer
update θ^Q
Q gradient w.r.t θ^Q
evaluation network Argument: θ^μ
y_i
soft update $\theta^Q \leftarrow \theta^Q$
Target-evaluation network Argument: $\theta^{\mu'}$
Gradient w.r.t. a
$a = \mu(s_i)$
$a' = \mu'(s_{i-1})$
Actor Network
Actor Optimizer
Policy gradient w.r.t. θ^μ
update θ^Q
policy network argument: θ^Q
soft update $\theta^{\mu'} \leftarrow \theta^{\mu!}$
Target-policy network argument: $\theta^{Q'}$
$\mu(s_t)$
Random noise
action
s_t
Environment
r_t
Reward function
$\mu(s_{i+1})$
store (s_t, a_t, r_t, s_{t+1})
Sample data
$N * (s_i, a_i, r_i, s_{i+1})$
Experience replay memory
Random sample strategy

Figure 7: Architecture of the Deep Deterministic Policy Gradient (DDPG) algorithm.

During training, the agent interacts with the environment as follows:

1. Observe current state s_t and use the Actor to output an action $a_t = \mu(s_t|\theta^\mu) + \mathcal{N}_t$, where $\mathcal{N}_t$ is exploration noise (e.g., Ornstein–Uhlenbeck process).

2. Execute a_t, receive reward r_t and next state s_{t+1}.

3. Store (s_t, a_t, r_t, s_{t+1}) in the replay buffer.

4. Sample a batch from the buffer and update:

  - The Critic using the Bellman target $y = r + \gamma Q'(s', \mu'(s'))$.
  - The Actor using the policy gradient derived from the Critic's feedback.

5. Soft-update target networks.

This combination allows DDPG to handle continuous and high-dimensional control problems efficiently, making it highly suitable for HVAC systems where actions (e.g., fan speed, valve position, or setpoint) are continuous variables.

6.6 Training and Evaluation Strategy

All experiments are conducted using a custom `HVACEnvironment` that interfaces with an FMU building model via the `pyfmi` library. Before each episode, the environment executes a mandatory 7-day warm-up using fixed control inputs to drive the thermal, humidity, and CO_2 states toward physically consistent conditions; the final FMU outputs serve as the initial state s_0.

Each episode simulates 60 days of operation (5760 steps at 15-minute intervals). At each timestep, the agent receives a 15-dimensional state vector including FMU outputs, current weather, a 1-hour-ahead outdoor temperature forecast, time features, occupancy, and the previous action. The agent outputs a 5-dimensional continuous control action, and the FMU advances one step.

Training proceeds for up to 50 episodes with mini-batches of 512 transitions and soft target updates. A forecast-aware hierarchical reward encourages predictive control actions. Evaluation is performed on unseen weather profiles to assess generalization, comfort robustness, and energy efficiency.

6.7 Implementation Details

The simulation and learning framework is implemented using the following toolchain:

- **Modeling:** HVAC system built in *Modelica Buildings Library 12.0.0*, exported as an FMU (Model Exchange mode).
- **Simulation:** *PyFMI* library for FMU integration, solver set to CVode (BDF/Newton, tolerance = 10^{-4}).
- **DRL Framework:** *Stable-Baselines3* (DDPG for benchmarking).
- **Data Processing:** *NumPy, Pandas, and Matplotlib* for preprocessing and analysis.
- **Environment:** Python 3.10, Conda environment `pyfmi_env`, executed on an AMD Ryzen 7 with 32 GB RAM.

The training loop interacts directly with the FMU simulation through the Gym-like interface, where each simulation step corresponds to a 5-minute control interval. All state variables are normalized to $[0, 1]$, and control actions are bounded within physical safety limits before being applied to the FMU. This setup ensures stable learning and reproducible results for subsequent evaluation.

V. System Design and Implementation

1. AI Model Integration

The proposed system adopts a supervisory control architecture in which a Deep Reinforcement Learning (DRL) agent operates on top of a physics-based hybrid simulation environment. The overall architecture consists of three principal components: (i) the hybrid physical model representing the HVAC equipment, thermal envelope (e.g: wall, roof..), and indoor air-quality (IAQ) dynamics (Modelica simulation environment, (ii) the DRL supervisory controller, and (iii) the environment data and forecasting module.

The DRL agent is integrated into the simulation as an external supervisory decision-making layer. At each simulation step, the hybrid physical model provides a compact state vector describing outdoor conditions, indoor thermal and IAQ measurements, and operational context (HVAC status, energy + previous action + forecast + occupancy). This state is passed to the actor network, which computes continuous HVAC control actions such as airflow rate, damper position, and chilled-water temperature setpoints. These actions are injected directly into the digital twin, replacing the conventional supervisory control logic while still respecting equipment-level safety constraints handled by local PI loops inside the physical model.

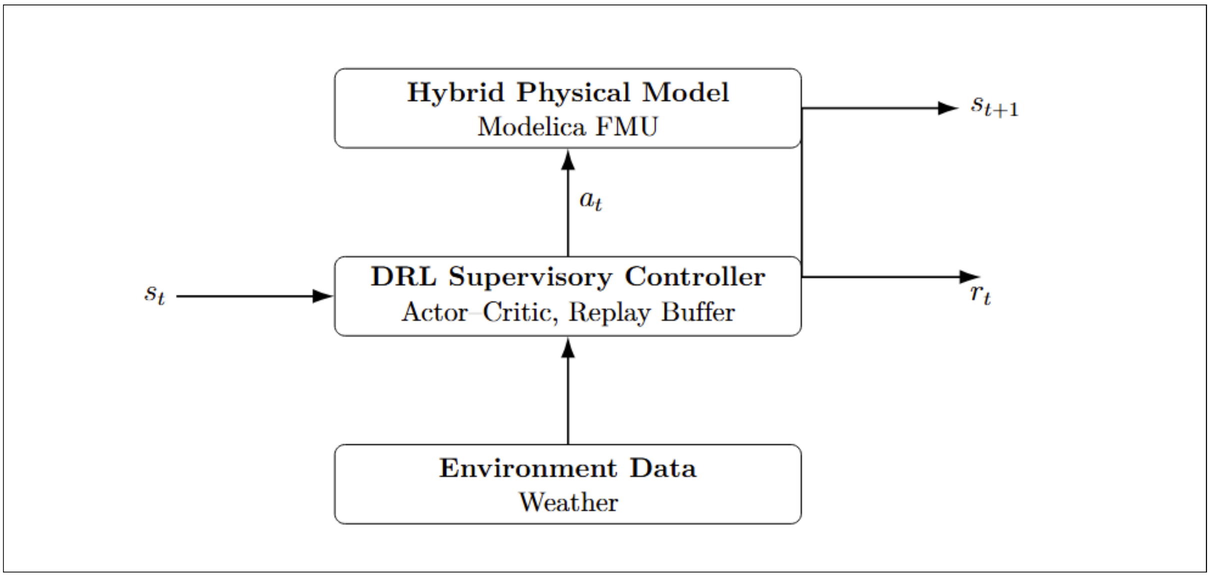


Figure 8: DRL integration with the hybrid physical model.

Within the data-flow pipeline, the DRL agent occupies the central closed-loop interaction point between state sensing and environment response. After executing the control actions, the hybrid model updates the thermodynamic and IAQ states, calculates system energy consumption, evaluates a multi-objective reward, and returns both the next state and reward signal to the agent. This completes the RL interaction cycle, enabling iterative learning of an optimal policy that coordinates ventilation, cooling, and filtration to jointly improve energy efficiency, thermal comfort, and indoor air quality.

2. Data Flow and Processing Pipeline

The data-processing pipeline follows the closed-loop interaction between the DRL controller and the hybrid physical simulation environment described in Section IV. The system operates in discrete 15-minute control intervals, where data continuously flows from the digital twin to the DRL agent and back. Fig. 10 illustrates the overall data flow, while the steps below detail the complete RL processing pipeline, including data inputs, transformations, control computation, environment feedback, and learning updates.

Below is the complete processing pipeline:

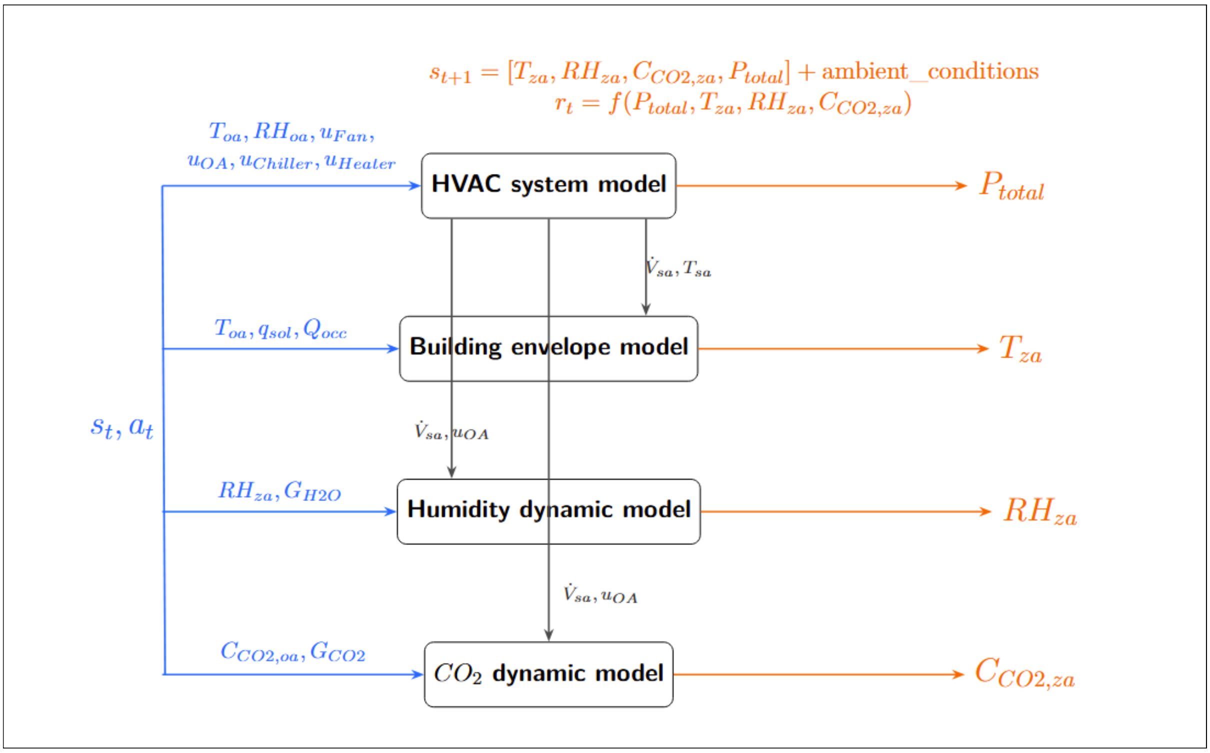


Figure 9: Data-flow and processing pipeline

At each timestep, the hybrid model generates a complete environmental state vector. This state is passed to the DRL agent, which computes control commands that are fedback into the simulation. The environment then updates its internal states based on HVAC physics, envelope dynamics, humidity, CO_2 model, returning the next state and a multi-objective reward. The RL agent stores each state transition tuple for training and updates its policy periodically.

2.1 Initialization

At the beginning of each simulation episode, the system initializes both the hybrid simulation environment and the DRL controller. The HVACEnvironment loads the FMU model, weather dataset, disturbance profiles, and internal configuration parameters, and then performs a mandatory 7-day warm-up period to drive the thermal, humidity, and air-quality states toward physically consistent initial conditions. This warm-up procedure simulates the FMU with fixed control inputs and constant weather boundary conditions, after which the final FMU outputs (zone temperature, humidity, CO_2, supply-air conditions, airflow rate, and power) are stored as the initial physical state.

Simultaneously, the DDPG agent initializes its actor and critic networks, constructs the corresponding target networks, allocates the prioritized replay buffer, and activates the adaptive Ornstein–Uhlenbeck noise process for exploration. All learning-rate schedulers, tracking variables, and exploration parameters are reset at the beginning of each episode. Once the warm-up is completed, the environment generates the first observation vector s_0 by assembling FMU outputs, weather conditions, occupancy level, time features, and the previous action statistics. This state serves as the input that begins the closed-loop interaction cycle between the DRL controller and the hybrid HVAC–IAQ digital twin.

2.2 State Generation (Modelica → DRL Agent)

At each simulation step, the hybrid model computes a complete state representation based on the FMU outputs and external conditions. The state vector s_t represents the 14-dimensional observation space and is assembled from the following components:

- **HVAC Subsystem State (FMU outputs):**
    - Supply-air conditions: Temperature (T_{SA}), intermediate cooling temperature, and Relative Humidity (RH_{SA}),
    - System operation variables: Supply airflow rate ($\dot{V}_{SA}$) and total power consumption (P_{total}).
- **Zone Thermal & IAQ State:**
    - Indoor air temperature (T_{zone}), reflecting envelope and thermal mass dynamics,
    - Indoor relative humidity (RH_{zone}),
    - Indoor CO_2 concentration (C_{CO2}).
- **External Boundary Conditions:**
    - Meteorological variables: Outdoor dry-bulb temperature (T_{drybulb}), outdoor relative humidity (RH_{out}),
    - Atmospheric pressure (P_{atm}) and wind speed,
    - Solar radiation components: Direct Normal Irradiance (DNI) and Diffuse Horizontal Irradiance (DHI).
- **Contextual Features:**
    - Temporal features: hour-of-day and day-of-week (encoding occupancy patterns),
    - Previous control actions (smoothed average), providing temporal continuity for the controller.

The resulting vector s_t provides a comprehensive snapshot of the thermal, hygric, and energetic status of the building, serving as the direct input to the DRL agent's policy network.

2.3 Sensor and Data Pre-processing:

Before being fed into the DRL agent, the raw physical outputs from the hybrid model are processed through a streamlined pipeline to construct the state observation:

- **Signal Extraction:** Key operational variables—including zone temperature (T_{zone}), relative humidity (RH_{zone}), CO_2 concentration, supply-air metrics, and total power consumption—are extracted directly from the FMU simulation outputs at each timestep.

- **Feature Construction:** These physical signals are assembled into a 14-dimensional observation vector that integrates external weather boundary conditions (temperature, humidity), occupancy status, and temporal context features (hour of day, day of week).

- **Temporal Augmentation:** To mitigate the non-Markovian effects caused by thermal inertia and system lag, the state vector is augmented with the previous control action (smoothed average), providing short-term historical context.

- **Input Binding:** The final NumPy state vector is converted into a `PyTorch FloatTensor`. Explicit input scaling is intentionally omitted; instead, feature normalization is handled dynamically through *Layer Normalization* embedded within both the Actor and Critic networks to ensure gradient stability during training.

2.4 Policy Evaluation

Given the constructed state vector s_t, the DRL agent computes the control action a_t through the following four-stage procedure:

- **Actor Forward Pass:** The state vector is fed into the Actor network to evaluate the deterministic policy $\pi(s_t)$, which outputs a continuous control vector in the range $[0, 1]$ using a Sigmoid activation function.

- **Exploration Strategy (Training Phase):** To prevent premature convergence and ensure sufficient exploration of the state–action space, an *Adaptive Ornstein–Uhlenbeck* noise process is superimposed onto the deterministic action. As defined in the `AdaptiveOUNoise` module, the noise magnitude decays over time to balance exploration and exploitation.

- **Action Vector Assembly:** The resulting action vector a_t represents the normalized control signals for the five key HVAC actuators defined in the simulation environment:

  - Supply Fan speed (u_{Fan}),

  - Outdoor Air Damper position (u_{OA}),

  - Chiller valve opening / load (u_{Chiller}),

  - Heater valve opening (u_{Heater}),
  - Exhaust Air Fan speed (u_{FanEA}).
- **Bound Enforcement:** The combined control signals are clipped to the feasible operating range $[0, 1]$ to satisfy the physical constraints of the actuators before being transmitted to the FMU simulation environment.

2.5 Action Execution

The action a_t is applied to the hybrid model through the following interface:

- **Direct Control Injection:** The DRL agent overrides the baseline control logic, directly modulating the actuators via normalized signals $[0, 1]$.
- **FMU Input Mapping:** The action vector is mapped to specific FMU input ports: fan speeds (u_{Fan}, u_{FanEA}), damper position (u_{OA}), and thermal valve openings $(u_{Chiller}, u_{Heater})$.
- **Physics Resolution:** The FMU solver resolves the differential-algebraic equations describing the HVAC dynamics and envelope thermal response.
- **Environment Advance:** The simulation propagates the digital twin forward by one timestep ($\Delta t = 15$ minutes).

2.6 Environment Update

After an action is executed at time step t, the integrated HVAC–envelope–IAQ environment advances to the next state through a coupled physical simulation. The HVAC module computes the thermodynamic response to the applied control, updating supply-air conditions and power consumption. The building envelope model propagates heat transfer through the structure while accounting for solar and outdoor disturbances to update zone temperature. In parallel, IAQ dynamics update indoor relative humidity and CO_2 concentration using mass-balance formulations that incorporate ventilation, infiltration, and occupancy-driven internal gains. The resulting physical states, together with temporal and exogenous variables, are assembled into the next observation s_{t+1}.

2.7 Learning Update

Given the transition tuple (s_t, a_t, r_t, s_{t+1}), the Improved DDPG agent performs a prioritized learning update. The transition is stored in the replay buffer with high priority and sampled using TD-error–based weighting with importance-sampling correction. Twin Critic networks are optimized by minimizing the weighted Bellman error to mitigate overestimation bias, while the Actor is updated via deterministic policy gradients to maximize the primary Critic's value. Target networks are softly updated using $\theta' \leftarrow \tau\theta + (1 - \tau)\theta'$ to ensure training stability.

2.8 Continuous Loop Operation

This interaction loop continues until the episode horizon, typically covering a 24-hour or multi-day simulation. After each episode, key performance indicators—including cumulative reward, comfort violations, and energy consumption—are logged to assess learning progress. A curriculum strategy progressively tightens comfort constraints, and an adaptive learning-rate scheduler reduces step sizes when performance plateaus. Model parameters corresponding to the best episode return are retained as checkpoints.

3. Deployment Strategy

3.1 System Architecture Diagram

The deployment architecture follows a three-tier design separating frontend, API, and computation layers:

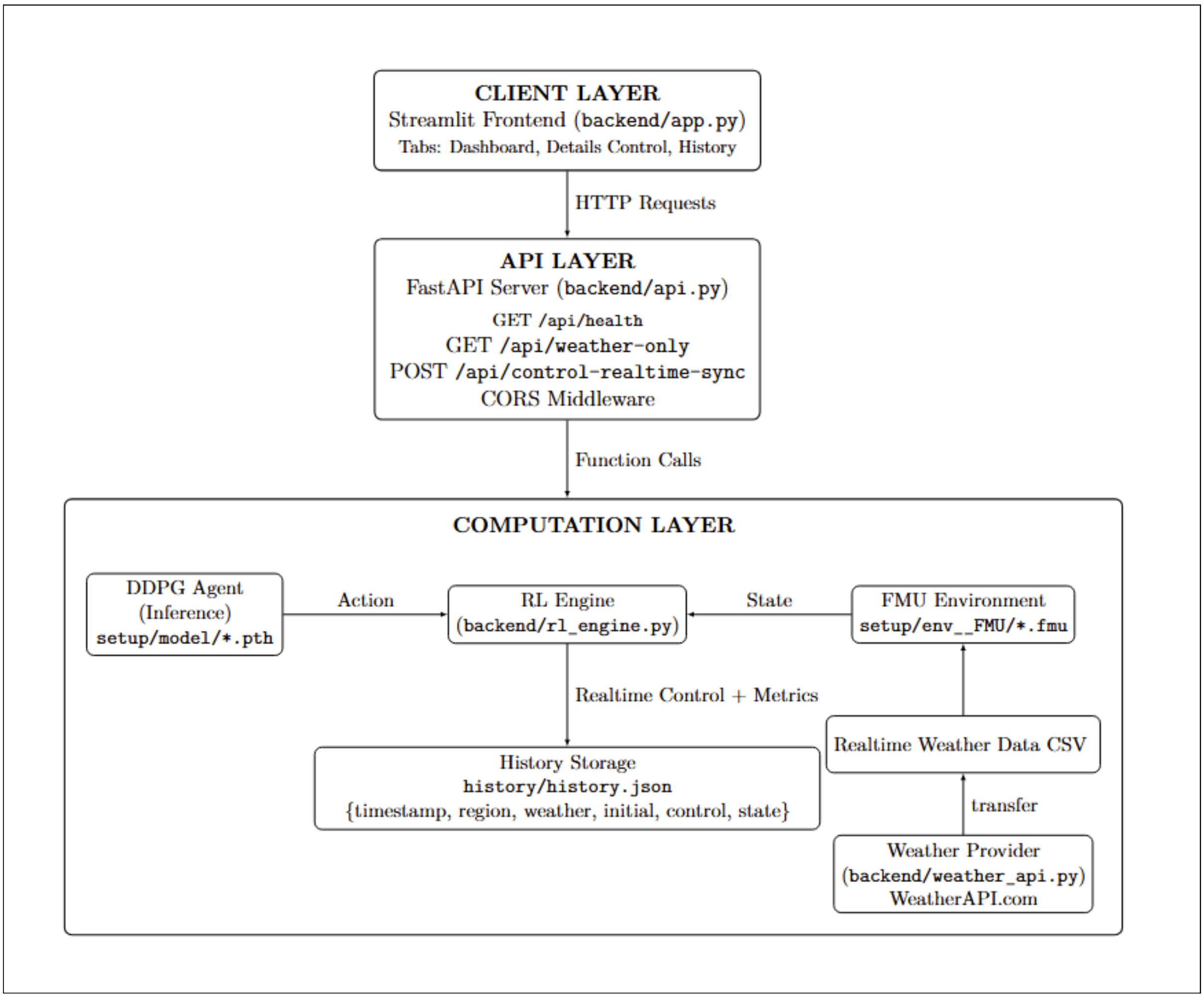
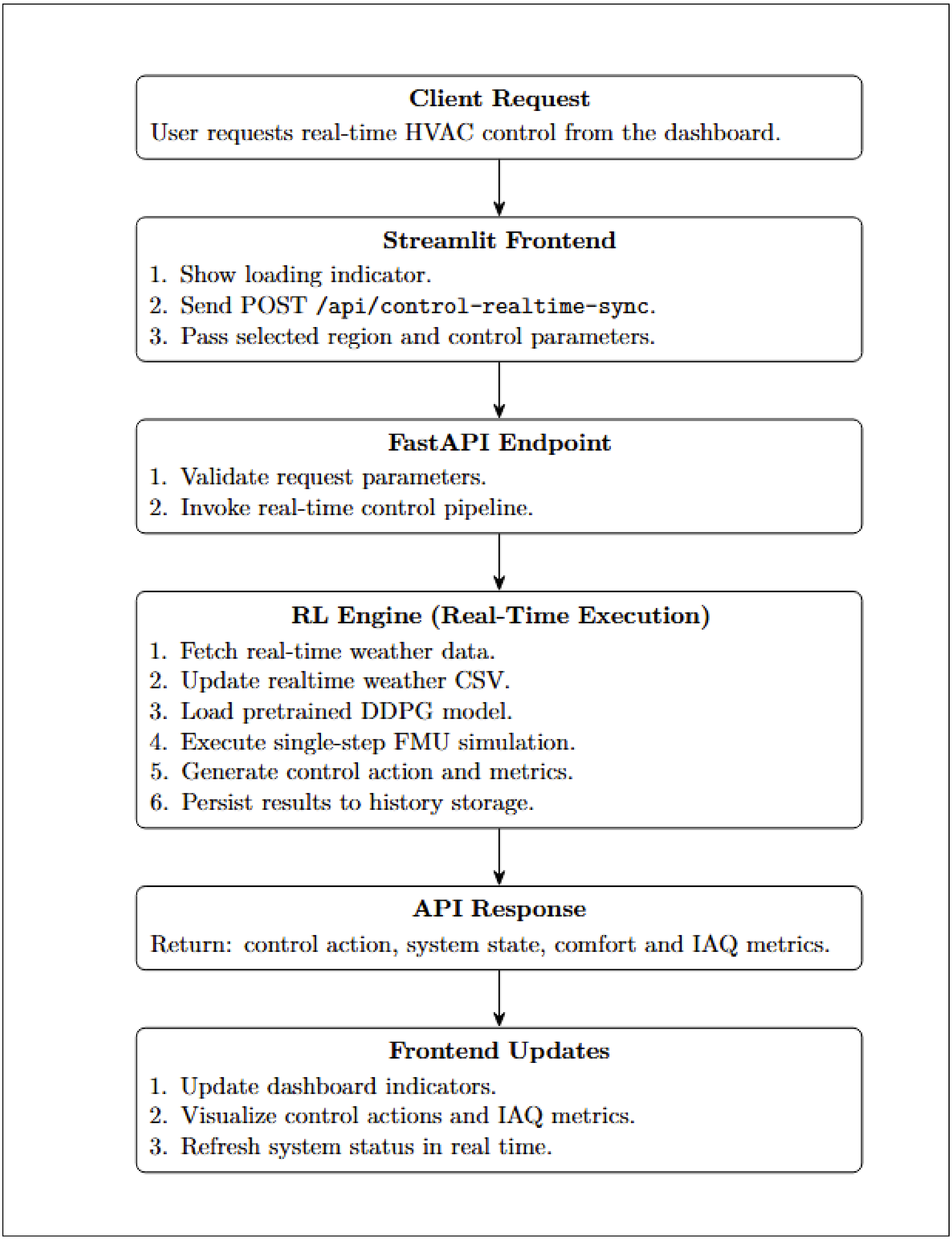


Figure 10: System deployment architecture for the HVAC RL With FMU real time.

This architecture is designed to support scalable and reliable real-time HVAC control. Reinforcement learning inference and FMU-based simulation are isolated within the computation layer, preventing performance bottlenecks at the API level. The stateless API enables horizontal scaling, while the client layer remains lightweight, focusing on supervisory control and real-time visualization. Overall, the architecture provides a robust foundation for deploying reinforcement-learning-based HVAC control in production environments.

3.2 Request-Response Flow

The API handles simulation requests through the following flow:



Client Request
User requests real-time HVAC control from the dashboard.

Streamlit Frontend
1. Show loading indicator.
2. Send POST `/api/control-realtime-sync`.
3. Pass selected region and control parameters.

FastAPI Endpoint
1. Validate request parameters.
2. Invoke real-time control pipeline.

RL Engine (Real-Time Execution)
1. Fetch real-time weather data.
2. Update realtime weather CSV.
3. Load pretrained DDPG model.
4. Execute single-step FMU simulation.
5. Generate control action and metrics.
6. Persist results to history storage.

API Response
Return: control action, system state, comfort and IAQ metrics.

Frontend Updates
1. Update dashboard indicators.
2. Visualize control actions and IAQ metrics.
3. Refresh system status in real time.

Figure 11: Request–response flow for the real-time HVAC RL control pipeline.

The proposed deployment architecture adopts a stateless, API-first design that decouples real-time control logic from the user interface, thereby enabling horizontal scalability and modular system evolution. The backend is implemented using FastAPI, providing lightweight RESTful endpoints for real-time HVAC control and weather data access.

At initialization, the pre-trained DDPG Actor network is loaded into the RL engine and set to evaluation mode to ensure deterministic inference. For each incoming request, a dedicated FMU-based HVAC environment is instantiated, ensuring isolation between sessions and preventing state leakage under concurrent execution.

The primary endpoint triggers a single-step real-time control pipeline, in which current weather data are retrieved, the FMU model is evaluated, and the DDPG agent computes the corresponding control actions. Resulting control signals and indoor environmental metrics are returned immediately and persisted for monitoring and post-analysis.

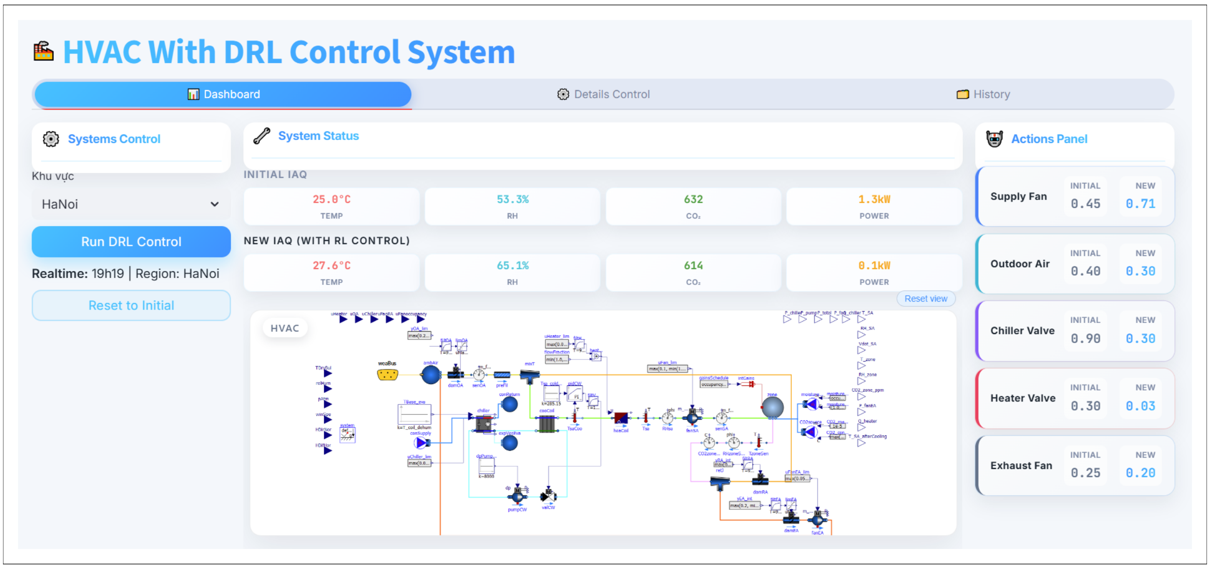
Overall, the stateless processing model supports efficient load balancing across multiple API instances and provides a robust foundation for deploying reinforcement-learning-based HVAC control in real-world applications.

3.3 Streamlit Front-End for Monitoring and Supervisory Control

UI Layout and Component Structure

The graphical user interface (GUI) of the HVAC with RL Control System (Fig. 12) is designed as an integrated supervisory environment that unifies monitoring, visualization, and control within a single operational framework. The interface is organized into three coordinated regions to support both real-time supervision and analytical evaluation.

The Streamlit interface is organized into a responsive dashboard layout:



HVAC With DRL Control System

Khu vực: HaNoi

Realtime: 19h19 | Region: HaNoi

INITIAL IAQ

TEMP	RH	CO₂	POWER
25.0°C	53.3%	632	1.3kW

NEW IAQ (WITH RL CONTROL)

TEMP	RH	CO₂	POWER
27.6°C	65.1%	614	0.1kW

	INITIAL	NEW
Supply Fan	0.45	0.71
Outdoor Air	0.40	0.30
Chiller Valve	0.90	0.30
Heater Valve	0.30	0.03
Exhaust Fan	0.25	0.20

Figure 12: UI Layout and Component Structure

The left panel provides access to core control functions, including execution of the DDPG agent, environment reset, and state synchronization. The central dashboard presents key indoor air quality (IAQ) indicators—temperature, relative humidity, CO_2, and power consumption—allowing direct comparison between baseline conditions and RL-controlled states. A schematic HVAC representation is embedded to facilitate visual interpretation of system dynamics during control execution. The right panel displays baseline and RL-generated actuator commands, enabling transparent inspection of control decisions and their impact on system behavior.

Prototype Deployment Using Secure Tunneling

In the current capstone demonstration, the system is deployed primarily as a local-hosted prototype. The backend (FastAPI + RL engine + FMU execution) is executed on a local machine, while the Streamlit interface runs locally as the supervisory layer. This deployment choice is driven by practical constraints of the dependency stack: the FMU simulation layer relies on `pyfmi`, which is difficult to build and install reliably on managed platforms such as Streamlit Community Cloud due to native/compiled requirements and environment limitations. Therefore, instead of cloud hosting, the project exposes the local service externally using ngrok tunneling to support remote access for demonstration purposes.

Operationally, the demo starts two local services: (i) the FastAPI server (e.g., `python -m uvicorn backend.api:app -host 0.0.0.0 -port 8000`) and (ii) the Streamlit UI (e.g., `streamlit run backend/app.py -server.port 8501`). Ngrok is then used to publish the localhost endpoints to temporary public URLs, allowing external users to access the UI and trigger control requests without requiring full cloud deployment. This approach enables a practical "personal-project deployment" suitable for presentation and short-term validation while preserving the native PyFMI/FMU execution environment locally.

Recommendation and Limitations

Although ngrok is convenient for rapid demos, it is not recommended for long-term or production deployment. Public tunnel URLs are temporary, operational stability depends on the tunneling session, and security hardening (authentication, rate-limiting, secrets management) is limited compared to standard cloud/VM deployments. For a more robust deployment path, the recommended direction is to containerize the backend (or run it on a dedicated VM/server where PyFMI can be installed properly), place the API behind a stable reverse proxy/load balancer, and host the frontend on a managed platform that does not require native FMU dependencies. In summary, ngrok is retained only as a short-term workaround for capstone demonstration when Streamlit Cloud deployment is blocked by PyFMI constraints, and should be phased out in future production-oriented iterations.

4. Scalability and Maintenance

Table 17: Scalability and maintenance-oriented design strategies of the proposed system

Aspect	Design Strategy	Scalability Impact	Maintenance Benefit
Frontend–Backend Separation	The Streamlit frontend communicates with the control logic exclusively through a RESTful FastAPI layer.	Enables independent horizontal scaling of UI and backend services without shared state coupling.	UI updates and bug fixes can be applied without modifying control or simulation logic.
Modular RL Engine	The reinforcement learning agent is encapsulated as an independent decision-making module.	Multiple agents or control strategies can be evaluated or deployed in parallel.	New algorithms (e.g., PPO, SAC) can replace the existing agent with minimal system changes.
FMU-Based Physical Modeling	HVAC dynamics are encapsulated within standardized FMU interfaces.	Supports reuse across buildings and scenarios without reimplementing physics models.	FMU updates or calibration do not affect the RL or frontend layers.
API-Driven Control Interface	Control actions and state queries are exposed via stateless API endpoints.	Allows multiple concurrent clients or supervisory tools to access the system.	Clear interface contracts simplify debugging and future refactoring.
Model Versioning Strategy	Best-performing and intermediate RL models are stored using versioned checkpoints.	Facilitates safe rollback and comparative evaluation under increased system complexity.	Model updates can be deployed incrementally without retraining from scratch.

VI. Results and Discussion

1. Experiment Setting

1.1 DDPG Implementation and Environment

All experiments are conducted using the Python implementation provided in the training script, utilizing a custom HVACEnvironment class that wraps the FMU model via the pyfmi library. The environment loads the FMU model, weather datasets (CSV format), and configuration parameters at startup.

Initialization and Warm-up Before any episode begins, the environment performs a mandatory 7-day warm-up procedure to drive the thermal, humidity, and CO_2 states toward physically consistent initial conditions. As implemented in the warmup() method, this procedure simulates the FMU using fixed control inputs (e.g., $u_{\text{Fan}} = 0.6$, $u_{\text{OA}} = 0.5$) under constant weather boundary conditions derived from the first timestep of the dataset. After the warm-up phase, the final FMU outputs serve as the initial state s_0 for the main simulation.

Interaction Loop The control interval is set to 900 seconds (15 minutes). At each timestep, the agent receives the current 14-dimensional state vector, computes a continuous action vector, and the environment advances the hybrid physical model by one simulation step. Disturbances—including occupancy levels, metabolic heat gains, moisture generation, and metabolic CO_2 emissions—are injected dynamically based on the simulation time.

Improved DDPG Agent The reinforcement learning controller is implemented as an Improved DDPG agent, incorporating several state-of-the-art enhancements:

- **Twin Q-Networks:** The critic employs two parallel Q-networks (Q_1, Q_2) to mitigate value overestimation bias.
- **Prioritized Experience Replay (PER):** A replay buffer with a capacity of 200,000 transitions stores tuples $(s_t, a_t, r_t, s_{t+1}, \text{done})$. During training, minibatches are sampled according to TD-error priority rather than uniform sampling.
- **Adaptive Mechanisms:** The agent incorporates:
  - an *Adaptive Learning Rate Scheduler*, which decreases the learning rate when episode rewards plateau for three consecutive episodes;
  - an *Adaptive Ornstein–Uhlenbeck* noise process whose magnitude decays over time to smoothly transition from exploration to exploitation.
- **Hierarchical Reward System**: Prioritizing in this order:
  - *Temperature (Highest Priority)*: If the room temperature is within the range from 25 °C to 28 °C, a bonus of +5 points is assigned. If the temperature is outside this range, a heavy penalty is applied, since human thermal comfort is considered paramount.
  - *Energy Consumption (Second Priority)*: Energy saving is encouraged, but only when both temperature and humidity are within their comfort ranges and remain stable. Comfort is never sacrificed in exchange for lower energy consumption.
  - *Humidity (Third Priority)*: Relative humidity between 40% and 70% is considered good. However, humidity is only optimized when the temperature is already stable; if temperature is unstable, the controller prioritizes temperature regulation first.

Table 18: Comparison of DDPG agents with different weight sets optimized using a hierarchical reward system

Metric	**Model 1**	**Model 2**	**Model 3**	**Model 4 (Best)**
Average Temperature (°C)	28.378	28.112	27.812	26.709
Temperature Std Dev (°C)	2.705	2.366	1.803	0.551
Temperature Comfort Ratio (%)	35.834	37.665	63.832	96.436
Average Power Consumption (kW)	1.108	1.082	1.033	0.984
Total Energy Consumption (kWh)	1139.73	1011.97	1100.87	1464.51
Average Humidity (%)	63.395	64.041	60.754	58.847
Humidity Std Dev (%)	13.54	12.021	9.341	3.426
Humidity Comfort Ratio (%)	61.737	60.105	88.903	99.378
Average Reward	-0.124	0.164	1.143	2.858
Total Reward per Episode	-1201.856	60.253	3600.73	16996.43
Action Diversity	0.402	0.39	0.504	1.0
Action Changes per Episode	2787.105	2640.92	7054.0	12792.0
Combined Comfort Ratio (%)	22.031	22.607	60.012	95.836
Episodes in Full Comfort (%)	60.526	66.0	4.163	96.015

With model 1, which has weights of (2, 1, 0.5), the test results did not meet expectations, with comfortable temperature and humidity levels, although energy consumption was somewhat optimized. Model 2, with weights of (3, 1.5, 1), showed a slight improvement in performance, but it was negligible. Increasing the weights further optimized the model's performance. Next, model 3, with weights of (4, 3, 2), showed relatively good performance, indicating that this Hierarchical Reward System was effective. Finally, model 4, with weights of (5, 4, 2.5), showed extremely good agent performance. This represents a significant step forward for our learning model.

1.2 DDPG with Weather Forecasting Implementation

All experiments are conducted using the Python implementation provided in the training script, utilizing a custom HVACEnvironment class that wraps the FMU model via the pyfmi library. The environment loads weather data in CSV format with an additional forecast column (TDryBul-forecast) that provides 1-hour-ahead temperature predictions, enabling the agent to make proactive control decisions based on anticipated weather conditions.

Forecast-Augmented State Representation Unlike the standard DDPG implementation with 14-dimensional state, this enhanced version incorporates weather forecasting through a **15-dimensional state vector**:

$$\mathbf{s}_t = [\underbrace{T_{\text{zone}}, \text{RH}_{\text{zone}}, \text{CO}_2, T_{\text{SA}}, \ldots, P_{\text{total}}}_{\text{8 FMU outputs}}, \underbrace{T_{\text{dry}}, \text{RH}_{\text{outdoor}}}_{\text{2 current weather}}, \underbrace{\boldsymbol{T}_{\text{forecast}}}_{\text{1 future weather}}, \underbrace{h, d}_{\text{2 time features}}, \text{occ}, \bar{a}_{t-1}] \tag{79}$$

The key innovation is the inclusion of T_{forecast}—a 1-hour-ahead outdoor temperature prediction that allows the controller to anticipate thermal load changes and adjust HVAC operations preemptively. This forecasting capability distinguishes the proposed method from reactive control strategies.

Initialization and Warm-up Before any episode begins, the environment performs a mandatory 7-day warm-up procedure to drive the thermal, humidity, and CO_2 states toward physically consistent initial conditions. During this phase, the FMU is simulated with fixed control inputs (e.g., $u_{\text{Fan}} = 0.6$, $u_{\text{OA}} = 0.5$) under constant weather boundary conditions derived from the first timestep of the dataset. Importantly, the forecast value is also initialized during warm-up to ensure continuity. After the warm-up phase, the final FMU outputs and the forecasted temperature serve as the initial state s_0 for the main simulation.

Interaction Loop with Forecast Integration The control interval is set to 900 seconds (15 minutes). At each timestep, the agent receives the current 15-dimensional state vector (including the 1-hour forecast), computes a continuous action vector, and the environment advances the hybrid physical model by one simulation step. Weather disturbances—including current outdoor temperature, humidity, and the forecasted temperature for the next hour—are interpolated from the CSV file and injected dynamically based on simulation time.

This is a visualization of the best weather forecast data we used in conjunction with DDPG. Because the time series data used to run our forecasts was not entirely accurate, requiring us to combine additional data, the results obtained for the forecasts were not optimal, even though previous forecasting models yielded good results on the training datasets.

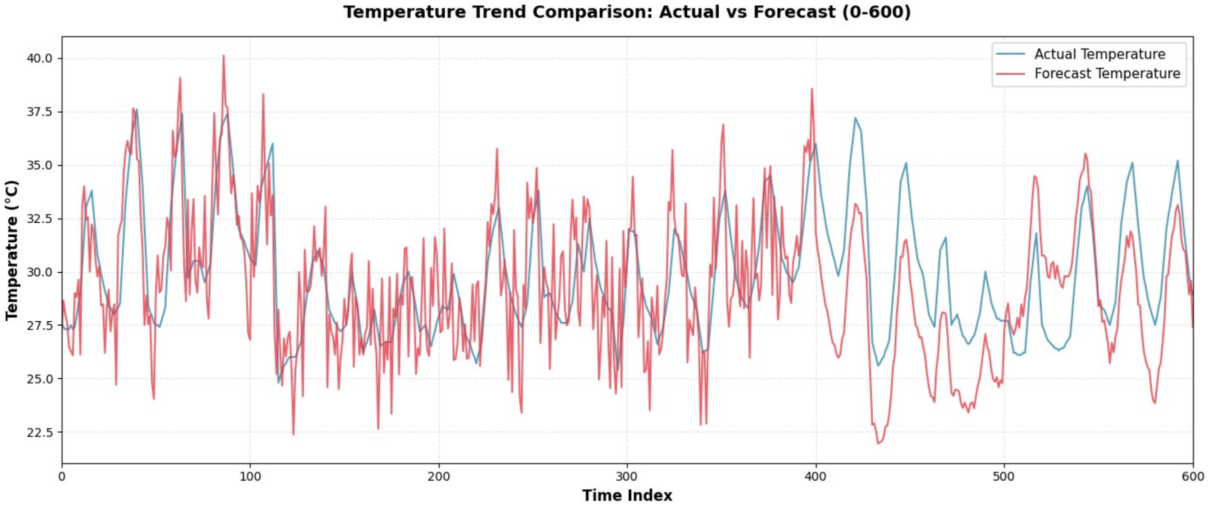


Figure 13: Visualize prediction results with time series data.

Improved DDPG Agent with Forecast-Aware Policy The reinforcement learning controller is implemented as an Improved DDPG agent that leverages the forecast-augmented state space. The agent incorporates several state-of-the-art enhancements:

- **Twin Q-Networks:** The critic employs two parallel Q-networks (Q_1, Q_2) to mitigate value overestimation bias, taking the minimum Q-value for target computation.
- **Prioritized Experience Replay (PER):** A replay buffer with a capacity of 200,000 transitions stores tuples $(s_t, a_t, r_t, s_{t+1}, \text{done})$. During training, minibatches are sampled according to TD-error priority with importance sampling weights, ensuring the agent focuses on informative transitions.
- **Forecast-Aware Architecture:** Both the actor and critic networks process the full 15-dimensional state, including T_{forecast}. The actor is a 4-layer feedforward network with LayerNorm (input $\rightarrow 512 \rightarrow 512 \rightarrow 256 \rightarrow 5$ output actions), while each critic Q-network concatenates state and action before passing through a similar architecture. This design allows the policy to explicitly condition on future weather predictions.
- **Adaptive Mechanisms:** The agent incorporates:
  - an *Adaptive Learning Rate Scheduler*, which decreases the actor and critic learning rates when episode rewards plateau for three consecutive episodes;
  - an *Adaptive Ornstein–Uhlenbeck* noise process whose magnitude decays from $\sigma_{\text{start}} = 0.7$ to $\sigma_{\text{end}} = 0.15$ over 200,000 steps, smoothly transitioning from exploration to exploitation while maintaining sufficient stochasticity.

- **Enhanced Exploration Strategy:** The agent uses epsilon-greedy exploration with decay ($\epsilon_{\min} = 0.2$, decay rate 0.9997) combined with OU noise. In early episodes (force_explore mode), the agent biases exploration toward mid-range actions ([0.15, 0.85]) to avoid extreme actuator settings, accelerating learning convergence.
- **Action Diversity Monitoring:** The agent tracks the standard deviation of recent actions (last 50 steps) to detect policy collapse. Low action diversity triggers warnings and is penalized in the hierarchical reward function, encouraging the agent to explore diverse control strategies rather than converging to suboptimal fixed actions.

Hierarchical Reward with Forecast-Driven Bonuses The reward function prioritizes thermal comfort over humidity and energy efficiency, with curriculum learning progressively tightening comfort bands over 30 episodes (initial: 24–29°C, final: 25–28°C).

Crucially, the reward calculator encourages **proactive forecast utilization** by:

- Penalizing excessive action changes (smoothness penalty) unless justified by forecasted weather shifts;
- Rewarding stable comfort maintenance when the forecast indicates favorable conditions;
- Providing action diversity bonuses to prevent the agent from ignoring forecast information and falling back to reactive control.

Training runs for up to 50 episodes with early stopping, saving checkpoints, per-episode CSV logs (including forecasted vs. actual temperatures), and comprehensive training plots. The best model is evaluated on new weather data with different forecast sequences to validate generalization and forecast-awareness.

In summary, the integration of 1-hour-ahead weather forecasting into the DDPG state space, combined with forecast-aware reward shaping and enhanced exploration, enables the agent to learn predictive HVAC control policies that outperform both rule-based controllers and standard reactive RL agents in terms of comfort, energy efficiency, and robustness.

2. Parameter Setting

Parameters are extremely important for the good operation of machine learning models, deep learning or RL agents. With DRL, these parameters are even more important, as with DDPG, there are parameters discount factor γ, Exploration noise ($\sigma_{start} \to \sigma_{end}$), Noise decay steps that play an essential role in both the training and optimization process, besides PPO also has parameters GAE parameter λ, PPO clip ratio ϵ, Target KL divergence that play a similar role. After determining the structure for the agents, below is a list of important parameters that greatly affect the learning speed and training process of the agents during our fine-tuning process.

3. Evaluation Metrics

The evaluation framework adheres to the hierarchical multi-objective design of the proposed method, assessing controller performance across energy efficiency, thermal comfort,

Table 19: Hyperparameters and Experimental Settings

Model parameter	**Value**
Simulation time step Δt	15 minutes
Warm-up duration	7 days
GAE parameter λ	0.95
Target KL divergence	0.01
Adaptive LR patience	3 episodes
Actor learning rate	1×10^{-4}
Critic learning rate	3×10^{-4}
Optimizer	Adam
Hidden layer units	512
Batch size	512
Replay buffer capacity	200,000
Soft update coefficient τ	0.005
Exploration noise ($\sigma_{start} \to \sigma_{end}$)	0.7 → 0.15 (Adaptive)
Noise decay steps	200,000
Target Temperature Band (Final)	25.5°C – 28.0°C
Target Humidity Band	40% – 70%

humidity regulation, and operational stability [38]. Based on the logged simulation data, the primary metrics include:

- **Total Energy Consumption (kWh):** Computed from the time integral of the total HVAC electrical power output extracted from the FMU.

$$E_{\text{total}} = \sum_{t=1}^{T} P_{\text{total}}(t)\, \Delta t \tag{80}$$

- **Thermal Comfort Violations:** Quantified as the cumulative deviation of the zone temperature T_{zone} from the adaptive comfort band.

$$V_T = \sum_{t=1}^{T} \left(\Big[T_{\text{zone}}(t) - \beta_{T,\text{high}}\Big]_+ + \Big[\beta_{T,\text{low}} - T_{\text{zone}}(t)\Big]_+ \right) \tag{81}$$

  where $[x]_+ = \max(x, 0)$.

- **Humidity Control Performance:** Expressed as the cumulative deviation when indoor relative humidity falls outside the target range (45%–65%).

$$V_{RH} = \sum_{t=1}^{T} \left(\Big[RH_{\text{zone}}(t) - 0.65\Big]_+ + \Big[0.45 - RH_{\text{zone}}(t)\Big]_+ \right) \tag{82}$$

- **Indoor Air Quality (CO_2):** Measured by the extent to which indoor CO_2 concentration exceeds the recommended threshold (e.g., 1000 ppm).

$$V_{\mathrm{CO}_2} = \sum_{t=1}^{T} \Big[C_{\mathrm{CO}_2}(t) - 1000\Big]_+ \tag{83}$$

- **Actuator Smoothness:** Operational stability is evaluated using the squared Euclidean distance between consecutive action vectors.

$$S = \sum_{t=1}^{T} \|\mathbf{a}_t - \mathbf{a}_{t-1}\|_2^2 = \sum_{t=1}^{T}\sum_{i=1}^{N_a} \Big(a_{i,t} - a_{i,t-1}\Big)^2 \tag{84}$$

- **Overall Multi-objective Reward:** The total episode reward aggregates prioritized objectives, including penalties for constraint violations and incentives for stable control.

$$R_{\mathrm{episode}} = \sum_{t=1}^{T} \Big(R_{\mathrm{base}} - \Big(w_T\mathcal{L}_T(t) + w_{RH}\mathcal{L}_{RH}(t) + w_E\mathcal{L}_E(t) + w_S S(t)\Big)\Big) \tag{85}$$

4. Compared Methods

Three control strategies were compared:

4.1 Rule-Based Controller (RBC).

As a baseline reference, the Rule-Based Controller (RBC) is evaluated first. RBC reflects the conventional threshold-based logic deployed in most existing HVAC systems, making it a natural benchmark for assessing the improvements offered by advanced controllers in Figure 14.

The simulation results show that the RBC system achieves very good temperature control performance, with values ranging from 22–25.5°C, which does not guarantee the optimal temperature threshold for cold days. In addition, the relative humidity shows significant fluctuations, ranging from 70% to 100% during the 60 simulation days, with a tendency to exceed the upper threshold of the comfort zone throughout the entire period. This shows that the RBC system does not do a good job in controlling humidity.

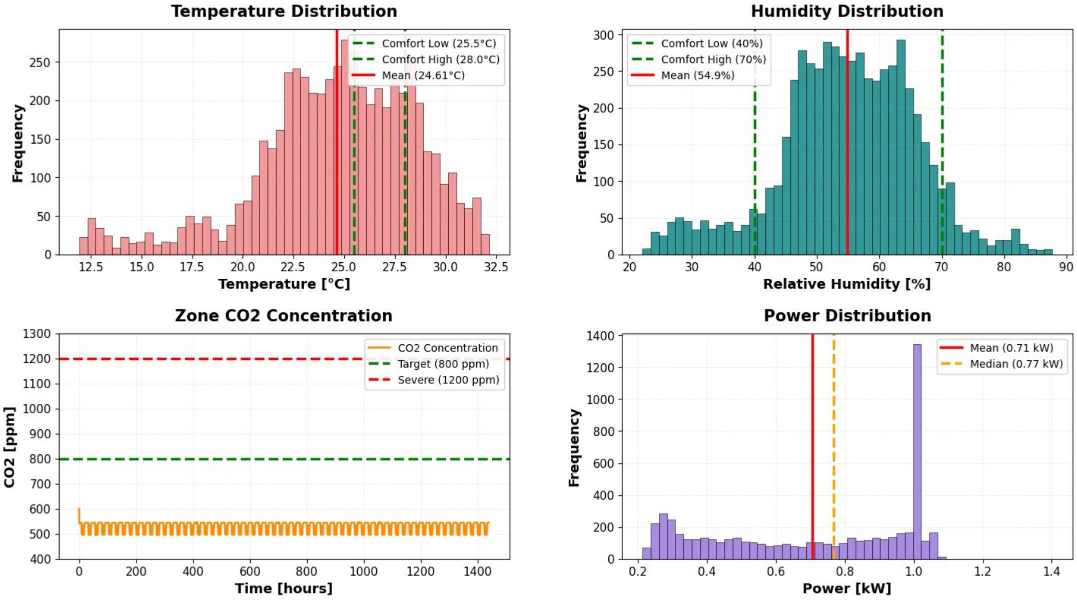


Figure 14: Performance of Rule-Based Controller (RBC) for HVAC systems in the Vietnamese climate context.

The CO_2 concentration is maintained at a stable low level (below 600 ppm), deep in the safe zone (below 800 ppm), proving that the ventilation system is operating effectively. However, the power consumption shows large and continuous fluctuations, ranging from 0.25 to 1.75 kW with high power peaks appearing frequently. This reflects the typical on/off control strategy of RBCs, which leads to suboptimal energy consumption and equipment wear due to continuous actuator operation.

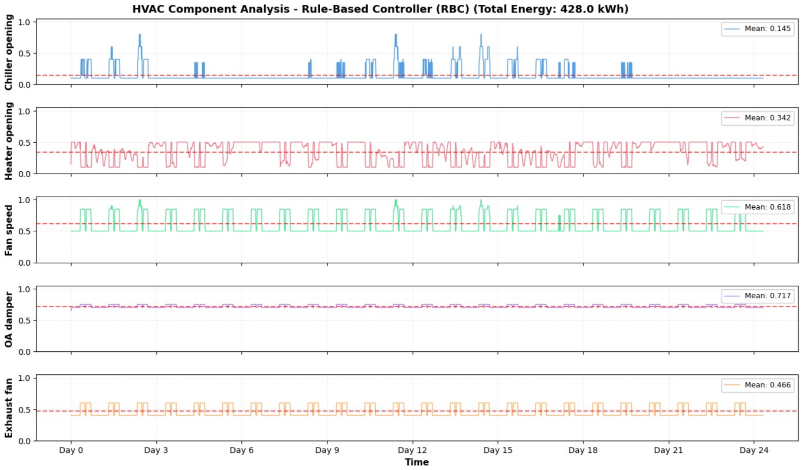


Figure 15: Components performance of Rule-Based Controller (RBC) for HVAC systems in the Vietnamese climate context.

RBCs are known for the highly chaotic operation of the system's components, processing actions through predefined rules. The image below easily illustrates the chaos and instability throughout the HVAC system's operation, leading to excessive energy consumption and negative impacts on both human comfort and the environment.

Overall, RBCs provide good temperature and air quality (CO_2), but have significant limitations in humidity control and energy efficiency. The large power fluctuations suggest the need for smarter control methods such as deep reinforcement learning (DDPG/PPO) to optimize both comfort and energy efficiency simultaneously.

4.2 Model Predictive Control (MPC).

A supervisory controller optimizing control actions over a prediction horizon using reduced-order models. While effective under accurate modeling assumptions, MPC suffers from nonlinear optimization complexity and performance degradation under unmeasured disturbances.

The Model Predictive Control (MPC) system demonstrated superior control over the RBC in many aspects in Figure 16. In terms of temperature, the MPC maintained an average value of 24.30°C with a standard deviation of ±1.80°C, although compliance was only 26.5% due to the tight setpoint, but the amplitude of the oscillation was significantly reduced compared to the RBC and followed the outdoor conditions more intelligently.

The most prominent strength of the MPC was its ability to control humidity, with an average value of 78.4% and low compliance (1.5%), showing that humidity was still high but the oscillation was much smoother and more predictable than the RBC (RBC oscillated completely irregularly from 70–100%). The MPC minimized extreme humidity peaks and generated a continuous control trajectory, reflecting the ability to optimize multiple objectives.

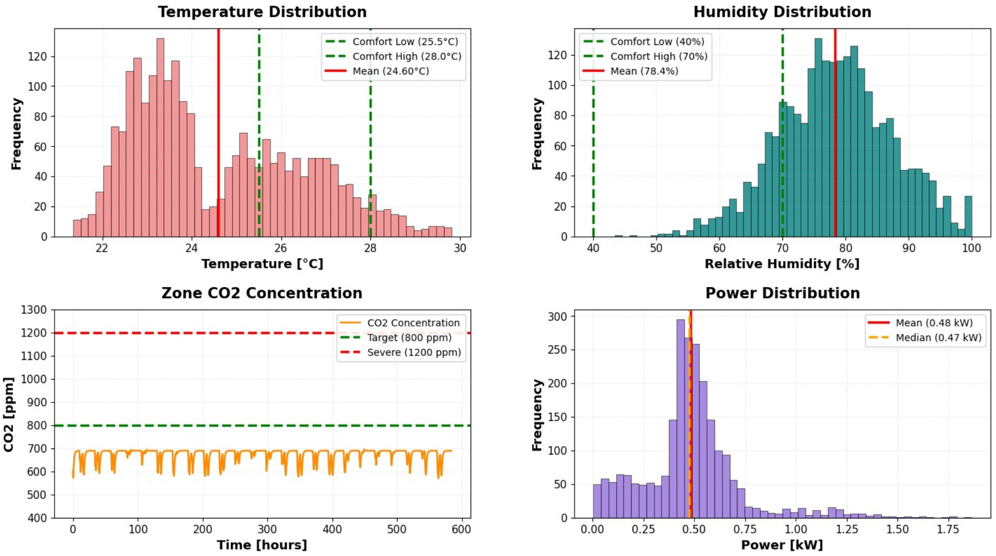


Figure 16: Performance of Model Predictive Control (MPC) for HVAC systems in the Vietnamese climate context.

The MPC's CO_2 control was near perfect, maintaining an average concentration of 680 ppm and fluctuating slightly around the 500 ppm setpoint, indicating that the ventilation system was effectively optimized. Most importantly, the MPC's energy consumption was significantly smoother, with better controlled power peaks (max 2.0 kW) and a lower energy baseline than the RBC (0.5–0.7 kW vs 0.05–0.35 kW continuous fluctuation). Despite the higher power peaks, the MPC performed more consistently thanks to its prediction and horizon optimization strategy, reducing equipment wear.

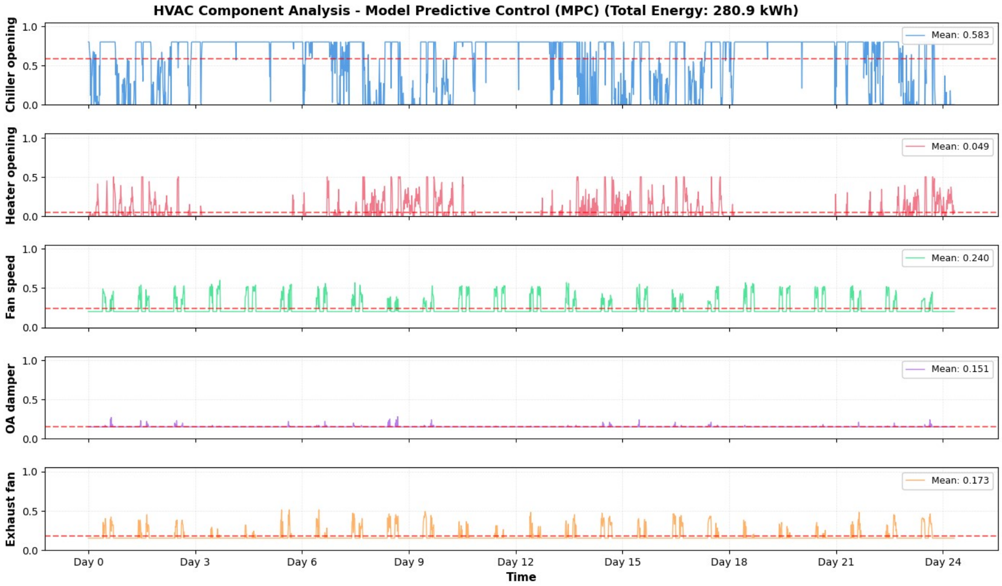


Figure 17: Components performance of Model Predictive Control (MPC) for HVAC systems in the Vietnamese climate context.

MPC is rated better than RBC, and this has been demonstrated during testing. The image below easily shows its greater stability compared to RBC throughout the HVAC system's operation. This reduces energy consumption, benefiting the environment; however, the system still fails to ensure a comfortable working environment for people.

Overall, the MPC outperformed the RBC in terms of prediction, multivariate optimization, and smooth control, but still fell short of fully optimizing the temperature and humidity comfort range.

4.3 Deep Deterministic Policy Gradient (DDPG).

A DDPG-based controller that learns directly from data, enabling coordinated multi-variable control. The agent balances energy, comfort, CO_2, without requiring explicit control-oriented models.

Through visualizations of the model results in Figure 18, it can be seen that the DDPG agent is doing a very good job of ensuring comfort with an optimal amount of energy. The agent has made the system maintain comfort almost throughout the simulation time. The comfort threshold is almost always guaranteed in the range 25.5°C to 28.0°C, in building people always feel comfortable and pleasant. With humidity, similarly, with the ability to learn through each episode, the agent has helped the HVAC system maintain a comfortable humidity threshold of 40% to 70% most of the time. Regarding CO_2, the agent is learning how to balance correctly with the minimum CO_2 threshold, to ensure human concentration. Finally, regarding energy, with an average hourly consumption of 0.15 to 2.4 kW, this is an optimal number for a working room of less than 15 people. In summary, the DDPG agent we built has been ensuring the balance of two factors: comfort and energy optimization.

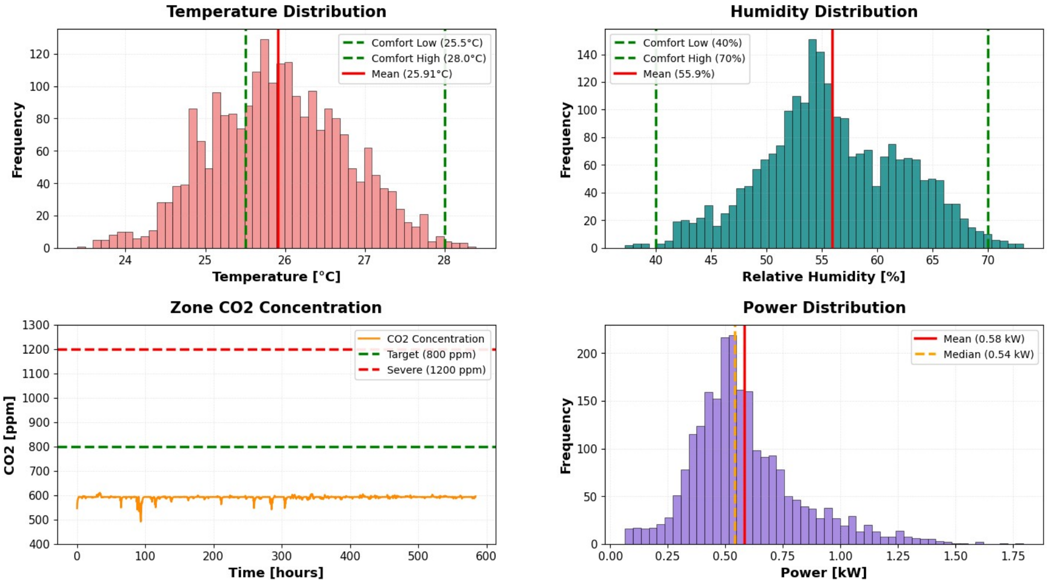


Figure 18: Performance of basic DDPG agent for HVAC systems in the Vietnamese climate context.

With the DDPG agent, it is considered superior to traditional methods, and our testing has demonstrated this. The image below easily shows the greater stability compared to traditional control models throughout the HVAC system's operation. However, to ensure human comfort in the workplace, the system requires continuous fine-tuning, resulting in a significant overall energy consumption for the HVAC system.

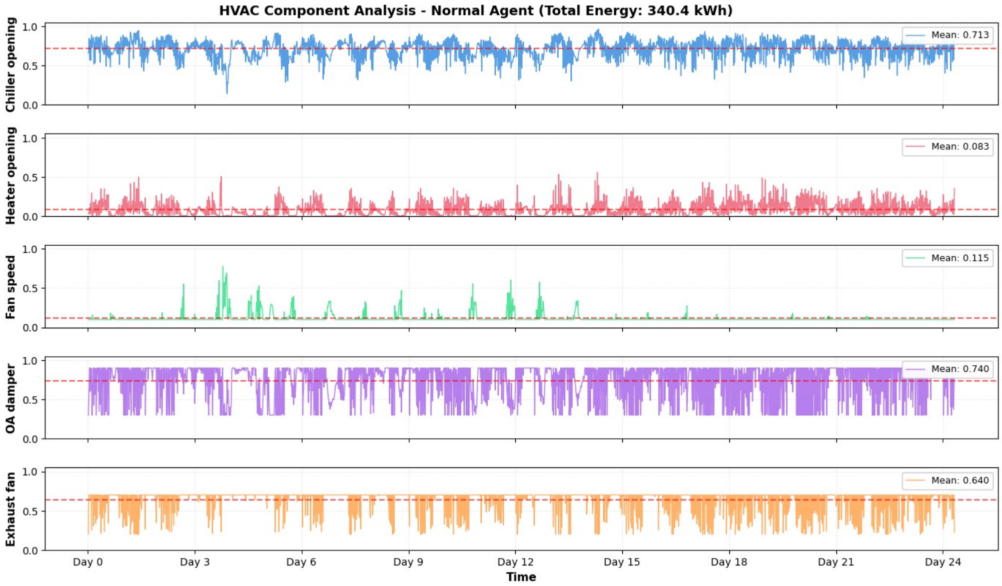


Figure 19: Components performance of basic DDPG for HVAC systems in the Vietnamese climate context.

In addition to the original DDPG structure, we also have a DDPG structure combined with forecasted data based on a Deep Learning model. Through the first testing step, we obtained results on 4 factors of temperature, humidity, CO_2 and energy as shown below.

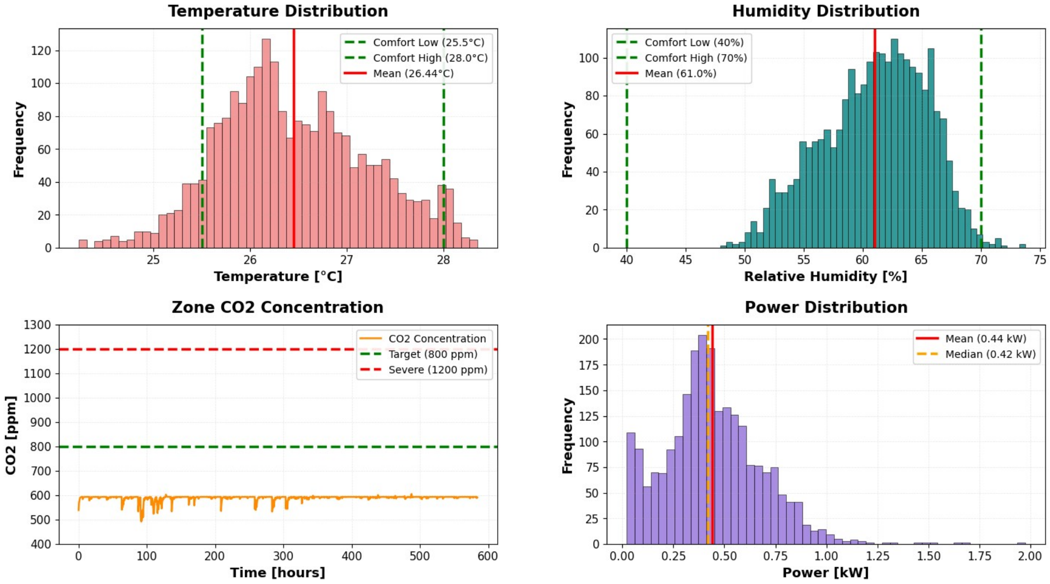


Figure 20: Performance of DDPG agent with forecast data for HVAC systems in the Vietnamese climate context.

Through the results in Figure 20 without optimization of the DDPG agent combined with the forecasted data, we can easily see that the agent is making progress when it can control humidity more optimally, always maintaining well within the comfortable threshold. As for temperature, there is still no clear optimization when compared with a basic DDPG, the deviations from the comfort zone are more. But with an agent that has not been tweaked too much, such results are very much in line with our expectations.

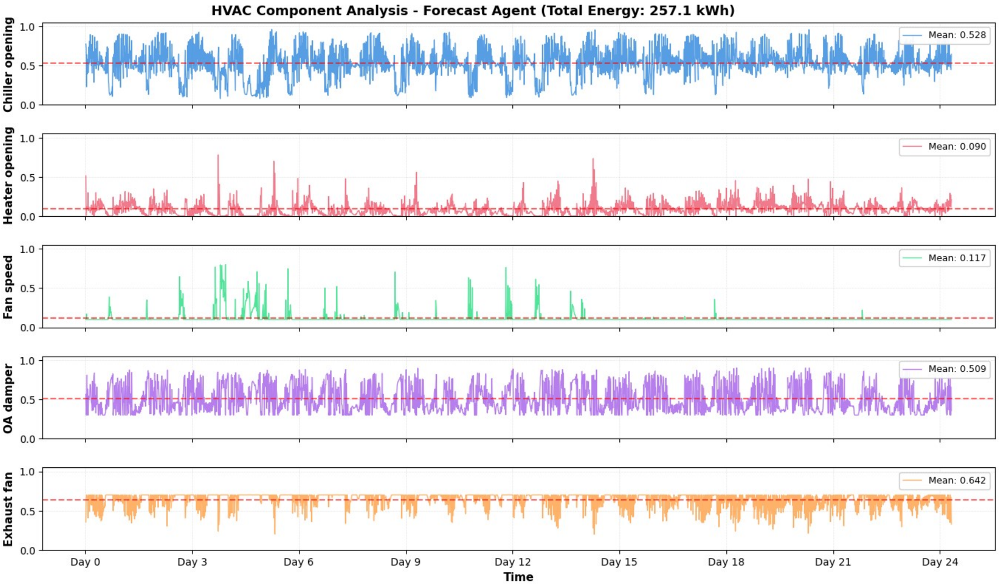


Figure 21: Components performance of basic DDPG for HVAC systems in the Vietnamese climate context.

Recognizing the inefficiency of conventional DDPGs in terms of energy consumption, we combined DDPGs with forecast data so the system could anticipate future conditions and make early adjustments. The result is an HVAC system that maintains stability while balancing both comfort and energy efficiency.

To better illustrate the differences between the DDPG agent and the DDPG agent combined with predictive data, the following are two tables containing key criteria for comparing the two agents. Table 20 highlights the percentage of temperatures within the comfort zone for temperature and humidity, the percentage of the optimal temperature zone for Vietnamese people (threshold > 27°C), and criteria related to temperature standard deviation and energy consumption. Table 14 shows the performance of the HVAC system components between the two agents. Table compares key metrics such as mean, standard deviation, uptime rate, and control signal change rate between Normal DDPG and DDPG with Forecast. These metrics show the stability, operating load, and "dynamic" level of each component in each control strategy. This allows readers to identify which components are controlled stably, which are more aggressively controlled, and the strategic differences between the two agents in the use of fans, fresh air, chillers, heaters, and exhaust fans.

Table 20: Performance Comparison of Two DDPG Agents for HVAC Control System

Metric	**DDPG Normal**	**DDPG with forecast**
Average Temperature (°C)	25.915	26.444
Temperature Std Dev (°C)	0.844	0.777
Temperature Comfort Ratio (%)	67.837	87.794
Temperature Stability for Vietnamese (%)	11.520	23.983
Average Humidity (%)	55.878	60.973
Humidity Std Dev (%)	6.349	4.427
Humidity Comfort Ratio (%)	98.801	99.400
Average Power Consumption (kW)	0.583	0.440
Total Energy Consumption (kWh)	340.366	257.094
Average Reward	1.718	2.751
Total Reward	4010.828	6424.372
Action Diversity	1.000	1.000
Action Changes	7951.000	8802.000
Combined Comfort Ratio (%)	67.024	87.268
Full Comfort Ratio (%)	67.024	87.238

The comparative analysis in Table 20 reveals that the DDPG with Forecast agent demonstrates superior overall performance across most evaluation metrics. With an average reward of 2.751 compared to 1.718 for the Normal agent, and achieving 87.24% full comfort ratio versus 67.02%, the forecast-augmented approach successfully maintains stable thermal conditions with lower temperature variability (std dev: 0.777°C vs 0.844°C). Most impressively, the Forecast agent achieves 87.79% temperature comfort ratio, representing a 20% improvement over the Normal agent's 67.84%. However, the Normal agent maintains higher temperature stability for Vietnamese climate preferences (11.52% time above 27°C). Notably, the Forecast agent demonstrates remarkable energy efficiency with 24% energy savings (257.09 kWh vs 340.37 kWh), while maintaining superior humidity control (99.40% comfort ratio vs 98.80%).

In contrast, the Normal agent exhibits more conservative control behavior with fewer total action changes (7,951 vs 8,802), suggesting a more stable but less adaptive control strategy. Despite this reduced activity, the Normal agent achieves lower overall comfort ratios, indicating that the forecast-based proactive adjustments are essential for optimal performance.

Table 21 provides deeper insights into the operational strategies of both agents. The most significant difference lies in the Outside Air Damper (uOA) control: the Normal agent maintains a higher setting (mean: 0.740, std: 0.211) compared to the Forecast agent's more moderate approach (mean: 0.509, std: 0.168), demonstrating the Forecast agent's ability to anticipate outdoor conditions and reduce unnecessary ventilation by 31%. Similarly, the Chiller operation shows substantial differences, with the Normal agent operating at 71.3%

average load compared to the Forecast agent's optimized 52.8%, resulting in significant cooling energy savings. The Forecast agent achieves better stability in the Exhaust Air Fan (uFanEA) operation (std: 0.093 vs 0.134), indicating more predictable ventilation management.

Overall, the DDPG with Forecast integration proves to be a superior solution that excels in both comfort maintenance and energy efficiency, achieving 20% improvement in comfort metrics while delivering 24% energy savings. The forecast-augmented approach represents a significant advancement in HVAC control, making it highly suitable for practical deployment. Meanwhile, the DDPG Normal agent, while demonstrating reasonable performance, serves as a solid baseline that validates the benefits of weather prediction integration for proactive control optimization.

Table 21: HVAC components operation comparison between two models

Component	**Metric**	**DDPG Normal**	**DDPG with forecast**
uFan	Mean	0.115	0.117
	Std Dev	0.058	0.073
	Operating Ratio (%)	11.478	9.979
	Change Rate (%)	15.803	13.533
uOA	Mean	0.740	0.509
	Std Dev	0.211	0.168
	Operating Ratio (%)	100.000	100.000
	Change Rate (%)	84.026	97.302
uChiller	Mean	0.713	0.528
	Std Dev	0.115	0.183
	Operating Ratio (%)	100.000	99.572
	Change Rate (%)	99.957	99.957
uHeater	Mean	0.083	0.090
	Std Dev	0.085	0.082
	Operating Ratio (%)	29.893	35.589
	Change Rate (%)	99.957	99.957
uFanEA	Mean	0.640	0.642
	Std Dev	0.134	0.093
	Operating Ratio (%)	100.000	100.000
	Change Rate (%)	40.771	66.210

From a broader perspective, in Figure 22 shows the energy efficiency of the DDPG when combined with forecast data for optimal performance, thanks to its ability to pre-calculate external temperature fluctuations and adjust HVAC system equipment accordingly. Furthermore, the temperature benchmarks are optimized to be closer to 28°C to better suit the weather throughout all four seasons in Vietnam.

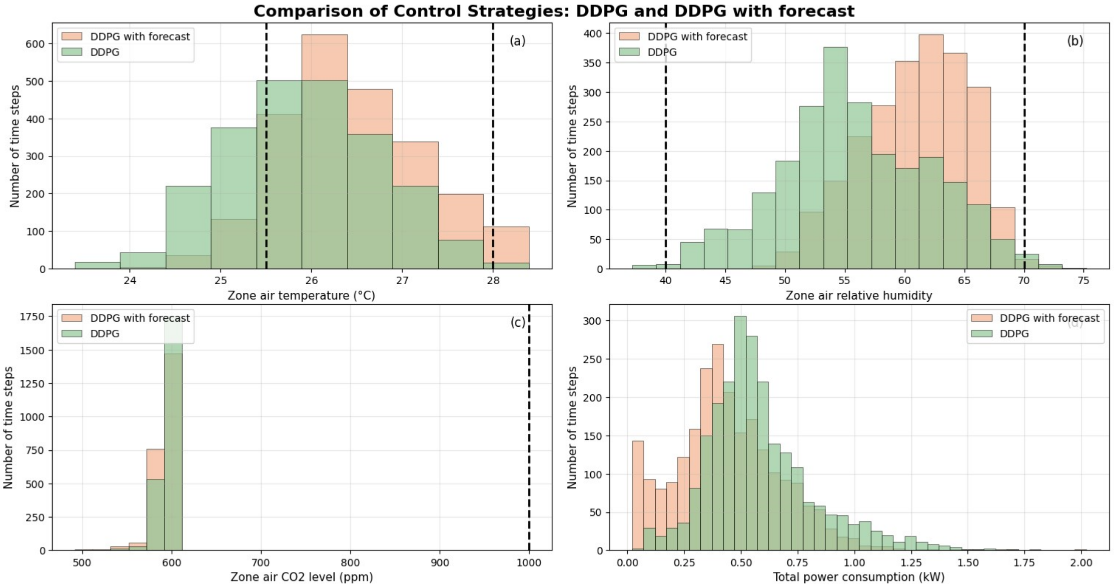


Figure 22: Comparison of temperature, humidity, CO_2 and energy factors of DDPG normal and DDPG with forecast data.

4.4 Comparison with State-of-the-Art Methods

Compared to the most common and typical HVAC control strategies currently in use, including RBC (Rest Control Monitoring) based on fixed rules and MPC (Data-Driven Predictive Control), DRL (Double-Reach Line) is the best performing option. Specifically, the DDPG and DDPG combined with predictive data that we have developed demonstrate several advantages:

- **Multi-objective coordination**: simultaneous optimization of thermal comfort, humidity, CO_2 concentration, and total energy consumption for the HVAC system, which few previous works have addressed holistically.

- **Adaptability**: DRL agents allow for handling nonlinearities and conflicting objectives without the need for conventional control-oriented model development in a rule-optimized manner.

- **Robustness**: superior performance under unmeasured disturbances and disturbances compared to MPC, RBC.

- **Computational efficiency**: energy costs have been optimized compared to MPC and RBC, while still ensuring the comfort of Vietnamese people.

- **Superior optimization**: by using forecast data combined with DRL, exceptionally good results have been achieved in terms of both thermal comfort and maximum energy optimization compared to all other methods.

- **Device performance**: figure 24 show that the performance of the devices in the HVAC system under both DDPG agents is very stable, especially with the DDPG agent combined with forecast data, ensuring optimal device operation.

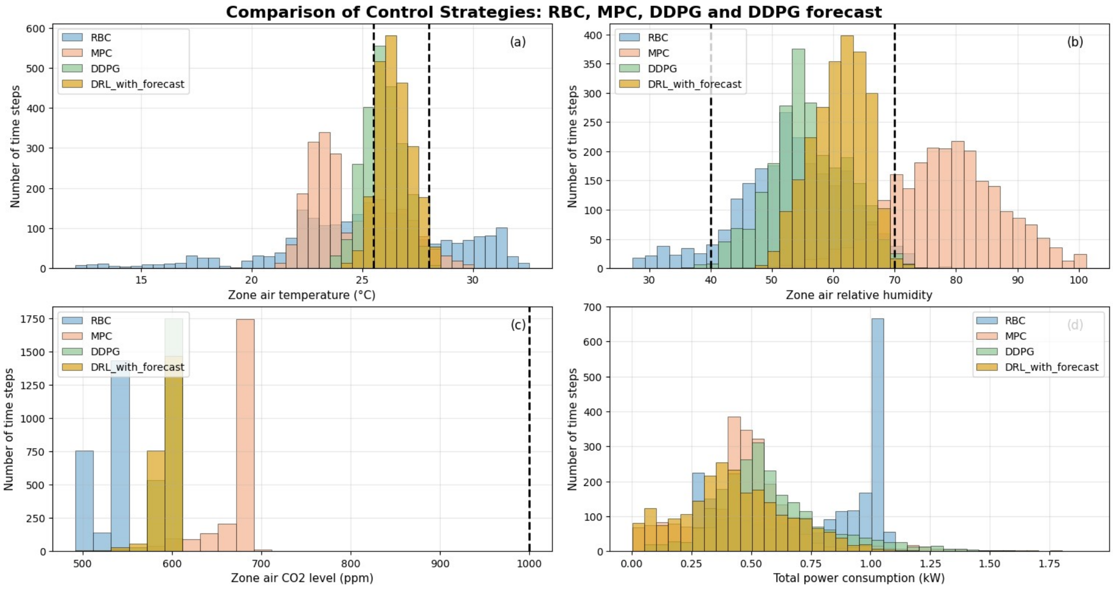


Figure 23: Comparison of temperature, humidity, CO_2 and energy factors of two agent DDPG with RBC, MPC.

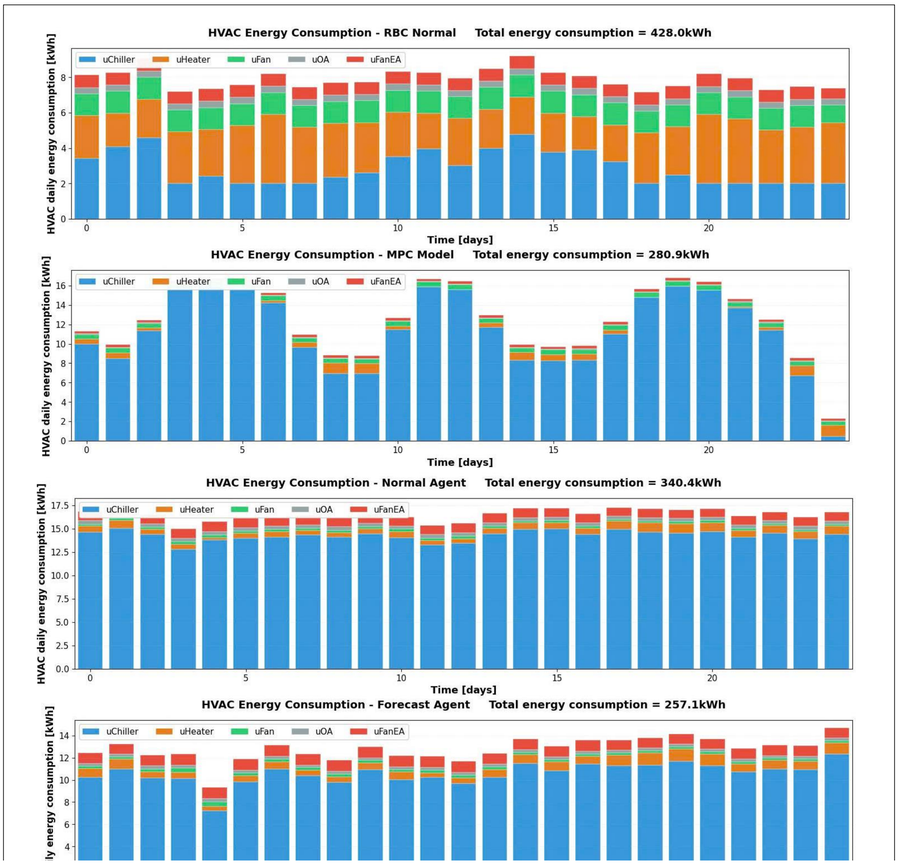


Figure 24: Comparison of device performance of two agent DDPG with RBC, MPC.

Overall, the two DDPG models offer a flexible and scalable alternative to model-based methods and superior optimization of the predictive data combination approach, especially for complex central HVAC systems with coupled dynamics and multi-objective constraints.

5. Recommendations

Based on the findings of this study, several recommendations are proposed to guide future development and deployment of RL-based HVAC control systems.

- **Hybrid Deployment:** RL controllers should initially operate alongside existing RBC or MBC strategies to ensure safety and provide a reliable fallback.
- **Simulation Environment Selection:** During the simulation process, we observed limitations in Modelica due to limited documentation and unstable libraries; therefore, we recommend initially testing control strategies in well-supported tools such as EnergyPlus before transitioning to Modelica for more detailed modeling.
- **Improve Sensor Quality:** Accurate and well-maintained sensors for temperature, occupancy, and airflow are essential to support stable RL performance.
- **Enable Periodic Retraining:** As building conditions change over time, RL policies should be periodically updated or fine-tuned to maintain optimal performance.
- **Align Rewards With Operational Goals:** Reward weights should reflect the building's priorities (e.g., comfort or energy savings) to produce desired behaviors.
- **Conduct Pilot Testing:** RL controllers should be tested in limited zones before full-scale deployment to reduce operational risk.
- **Ensure Safety and Monitoring:** Constraint filters, anomaly detection, and manual override mechanisms should be implemented to guarantee safe real-world operation.

6. Simulation to reality

6.1 Refine the simulation model with real-world data

After achieving good simulation results, data from the real-world HVAC system needs to be collected to calibrate the Modelica model parameters. This process includes adjusting building thermodynamic parameters such as thermal resistance, heat mass, and HVAC system characteristics so that the simulation model more accurately reflects real-world behavior.

6.2 Deploy testing and benchmarking

Deploy the trained DRL controller to the real system with a fail-safe mechanism to automatically revert to default control mode if an error occurs. Simultaneously, establish a benchmarking process by running the DRL controller and the standard controller in parallel or alternately to compare performance in terms of energy consumption and thermal comfort.

6.3 Integrate hardware and software

Connect the DRL controller to the existing Building Management System via standard communication protocols. This requires developing an interface so that the DRL can read sensor data (indoor/outdoor temperature, humidity, solar radiation) and send control commands to HVAC devices including valves, dampers, or temperature setpoints.

6.4 Continuous Monitoring and Refinement

After deployment, continuously monitor performance and collect data to refine the DRL agent through transfer learning or online learning.

VII. Conclusion

This study demonstrates the strong potential of reinforcement learning (RL) as an effective and adaptive control strategy for HVAC systems. Using a high-fidelity Modelica simulation environment, the proposed RL framework—evaluated through DDPG, PPO, consistently outperformed traditional Rule-Based Control (RBC) and Model-Based Control (MBC) across key metrics including energy consumption, thermal comfort, and control smoothness. The RL agents successfully learned complex building dynamics, anticipated disturbances, and generated intelligent control actions without requiring explicit system models.

The results highlight several advantages of RL-based HVAC control, such as improved comfort maintenance, reduced energy usage, minimized actuator wear, and robust operation under sensor noise and partial system faults. These findings indicate that RL can support more efficient, resilient, and occupant-centric building operation.

Despite these benefits, challenges remain regarding training stability, reward design, interpretability, and scalability. Addressing these limitations, alongside real-world validation will be essential for enabling broader adoption. Overall, the outcomes of this research suggest that RL offers a promising and flexible foundation for next-generation smart building management systems.

Future research can further enhance the robustness and applicability of the proposed RL-based HVAC control framework. A key direction is the exploration of real-building deployment and improved simulation-to-reality transfer through techniques such as transfer learning, domain randomization, and online fine-tuning.

Another promising avenue is the use of multi-agent and hierarchical RL to improve scalability in large or multi-zone buildings, enabling coordinated decision-making across complex HVAC networks. Incorporating occupant feedback and preference learning may also enhance comfort personalization while maintaining energy efficiency.

Future work should additionally consider integrating richer sensing modalities, including indoor air quality metrics, CO_2 levels, and real-time weather data, to improve situational awareness. Hybrid approaches that combine RL with physics-informed or model-based components may also increase control stability and interpretability.

Finally, advancing explainability and safety mechanisms such as interpretable RL policies, formal safety constraints, and fault-tolerant control strategies—will be essential to ensuring reliable and trustworthy deployment in real-world building environments.

References

[1] T. Al Mindeel, E. Spentzou, and M. Eftekhari. Energy, thermal comfort, and indoor air quality: Multi-objective optimization review. *Renewable and Sustainable Energy Reviews*, 202:114682, 2024.

[2] ASHRAE. *ASHRAE Handbook – HVAC Systems and Equipment*. ASHRAE, 2021.

[3] ASHRAE. Ansi/ashrae/ies standard 90.1: Energy standard for buildings except low-rise residential buildings, 2022.

[4] Y. Boutahri and A. Tilioua. Reinforcement learning for hvac control and energy efficiency in residential buildings with boptest simulations and real-case validation. *Discover Computing*, 28:45, 2025.

[5] A. Campoy-Nieves, A. Manjavacas, J. Jiménez-Raboso, M. Molina-Solana, and J. Gómez-Romero. Sinergym – a virtual testbed for building energy optimization with reinforcement learning. *Energy and Buildings*, 307:113641, 2024.

[6] Richard C. Dorf and Robert H. Bishop. *Modern Control Systems*. Pearson, 13th edition, 2017.

[7] G. Fangzhou, H. W. Sang, D. Kim, and H. J. Moon. Deep reinforcement learning control for co-optimizing energy consumption, thermal comfort, and indoor air quality in an office building. *Applied Energy*, September 2024.

[8] G. Gao, J. Li, and Y. Wen. Energy-efficient thermal comfort control in smart buildings via deep reinforcement learning. *IEEE Internet of Things Journal*, 7(9):8472–8484, 2020.

[9] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[10] N. V. Hung et al. Assessing daily maximum heat index in the context of climate change: Case study hanoi, vietnam. *Applied Ecology and Environmental Research*, 22(5), 2024. RH analysis for Hanoi climate 2024-2028.

[11] R. Jia, M. Jin, K. Sun, T. Hong, and C. Spanos. Advanced building control via deep reinforcement learning. *Energy Procedia*, 158:6158–6163, 2019.

[12] Ruoxi Jia, Ming Jin, Kaiyu Sun, Tianzhen Hong, and Costas Spanos. Advanced building control via deep reinforcement learning. *Energy Procedia*, 158:6158–6163, 2019.

[13] K. Kadamala, D. Chambers, and E. Barrett. Enhancing hvac control systems through transfer learning with deep reinforcement learning agents. *Energy and Buildings*, 2024.

[14] H. Kazmi, J. Suykens, A. Balint, and J. Driesen. Multi-agent reinforcement learning for modeling and control of thermostatically controlled loads. *Applied Energy*, 238:1022–1035, 2019.

[15] Lawrence Berkeley National Laboratory. *Buildings Library Documentation*, 2023. Cooling and Dehumidifying Coil Models.

[16] D. Le et al. Thermal comfort in mixed-mode cooled houses: A field study in the hot-humid climate of danang, vietnam. *Building and Environment*, March 2025.

[17] Dung Le. *Energy-saving measures for existing houses in Da Nang, Vietnam.* PhD thesis, Yokohama National University, 2024. Neutral temperature: 30.4°C SET*.

[18] C. Lee, D. Kim, and J. Park. Deep reinforcement learning control for co-optimizing energy consumption, thermal comfort, and indoor air quality in an office building. *Applied Energy*, 350:120150, 2024.

[19] J. Li, W. Zhang, G. Gao, Y. Wen, G. Jin, and G. Christopoulos. Toward intelligent multizone thermal control with multiagent deep reinforcement learning. *IEEE Internet of Things Journal*, 8(4):2678–2689, 2021.

[20] P. Lissa, M. Schukat, and E. Barrett. Transfer learning applied to reinforcement learning-based hvac control. *SN Computer Science*, 1(3):127, 2020.

[21] Ministry of Science and Technology of Vietnam. TCVN 5687:2010: Ventilation – Air Conditioning – Design Standards. Vietnam Standard, 2010. (in Vietnamese: Thong gio – Dieu hoa khong khi – Tieu chuan thiet ke).

[22] A. Nagy, H. Kazmi, F. Cheaib, and J. Driesen. Deep reinforcement learning for optimal control of space heating. *arXiv preprint arXiv:1805.03777*, 2018.

[23] A. T. Nguyen et al. Analysis of passive cooling and heating potential in hot-humid climate of vietnam. In *CISBAT 2011*, 2011. Comfort at 28.5-29.5°C with 90% RH for Vietnamese.

[24] Katsuhiko Ogata. *Modern Control Engineering.* Prentice Hall, 5th edition, 2010.

[25] T. Peirelinck, F. Ruelens, and G. Deconinck. Using reinforcement learning for optimizing heat pump control in a building model in modelica. In *2018 IEEE International Energy Conference (ENERGYCON)*, pages 1–6, 2018.

[26] Thijs Peirelinck, Frederik Ruelens, and Geert Deconinck. Using reinforcement learning for optimizing heat pump control in a building model in modelica. In *2018 IEEE International Energy Conference (ENERGYCON)*, pages 1–6, 2018.

[27] Minjae Shin, Sungsoo Kim, Youngjin Kim, Ahhyun Song, Yeeun Kim, and Ha Young Kim. Development of an hvac system control method using weather forecasting data with deep reinforcement learning algorithms. *Building and Environment*, 248:111069, 2024.

[28] J.C. Solano, E. Caamaño-Martín, L. Olivieri, and D. Almeida-Galárraga. Hvac systems and thermal comfort in buildings climate control: An experimental case study. *Energy Reports*, 7:269–277, 2021.

[29] F. Spada, E. D. Castronuovo, and T. Parisini. Reinforcement learning for hvac control and energy efficiency in residential buildings with boptest simulations and real-case validation. *Discover Computing*, 28(1):1–21, 2025.

[30] Y. Tian, L. Zhang, Y. Yang, and Q. Xu. Employing federated learning for training autonomous hvac systems. *Energy and Buildings*, 2025.

[31] U.S. Department of Energy. *EnergyPlus Engineering Reference*, 2024. Cooling and Heating Coil Energy Balance Models.

[32] U.S. Department of Energy. *EnergyPlus Engineering Reference*, 2024. Zone model and HVAC component descriptions.

[33] U.S. Department of Energy. *EnergyPlus Engineering Reference*, 2024. Zone Moisture Balance and Zone Contaminant Transport Models.

[34] Marshall Wang, John Willes, Thomas Jiralerspong, and Matin Moezzi. A comparison of classical and deep reinforcement learning methods for hvac control. In *2023 IEEE Smart World Congress (SWC)*, pages 1–7, 2023.

[35] T. Wei, Y. Wang, and Q. Zhu. Deep reinforcement learning for building hvac control. In *54th ACM/EDAC/IEEE Design Automation Conference (DAC)*, pages 1–6, 2017.

[36] Tianshu Wei, Yanzhi Wang, and Qi Zhu. Deep reinforcement learning for building hvac control. In *2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC)*, pages 1–6, 2017.

[37] T. Yang, L. Zhao, W. Li, J. Wu, and A. Y. Zomaya. Towards healthy and cost-effective indoor environment management in smart homes: A deep reinforcement learning approach. *Applied Energy*, 300:117335, 2021.

[38] M. Zhang et al. Modelling building hvac control strategies using a deep reinforcement learning approach. *Energy and Buildings*, 353:113441, 2024.

[39] Z. Zhang and K. P. Lam. Practical implementation and evaluation of deep reinforcement learning control for a radiant heating system. In *Proceedings of the 5th Conference on Systems for Built Environments (BuildSys '18)*, pages 148–157, New York, NY, USA, 2018. ACM.

[40] Dian Zhuang, Vincent J.L. Gan, Zeynep Duygu Tekler, Adrian Chong, Shuai Tian, and Xing Shi. Data-driven predictive control for smart hvac system in iot-integrated buildings with time-series forecasting and reinforcement learning. *Applied Energy*, 338:120936, 2023.

[41] Karl J. Åström and Tore Hägglund. *PID Controllers: Theory, Design, and Tuning.* Instrument Society of America, 2nd edition, 1995.

[42] Karl J. Åström and Richard M. Murray. *Feedback Systems.* Princeton University Press, 2010.