Self-Adaptive Cyber Physical System using Reinforcement Learning

A major project report submitted in partial fulfilment of the requirement for the award of degree of

Bachelor of Technology

in

Computer Science & Engineering / Information Technology

Submitted by

Hari Priya Radhika Sharma (201310)

Anoushka Sud (201146)

Under the guidance & supervision of

Prof. Dr Vivek Kumar Sehgal

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology, Waknaghat, Solan - 173234 (India)

CERTIFICATE

This is to certify that the work which is being presented in the project report titled "Self Adaptive Cyber-Physical Systems using Reinforcement Learning" in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science And Engineering and submitted to the Department of Computer Science And Engineering, Jaypee University of Information Technology, Waknaghat is an authentic record of work carried out by Hari Priya Radhika Sharma 201310 and Anoushka Sud 201146, during the period from August 2023 to May 2024 under the supervision of Prof. Vivek Sehgal, Department of Computer Science and Engineering, Jaypee University of Information Technology, Waknaghat.

Hari Priya Radhika Sharma (201310)

Anoushka Sud (201146)

The above statement made is correct to the best of my knowledge.

Prof. Vivek Sehgal

Professor and Head,

Department of Computer Science & Engineering and Information Technology,

Jaypee University of Information Technology, Waknaghat.

CANDIDATE'S DECLARATION

We hereby declare that the work presented in this report entitled **Self-Adaptive Cyber Physical System using Reinforcement Learning** in partial fulfillment ofthe requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering/Information Technology** submitted in the Department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology,Waknaghat is an authentic record of my own work carried out over a period from August 2023 to May 2024 under the supervision of **Prof. Dr Vivek Kumar Sehgal**(Professor and Head, Dept. of CSE & IT, Department of Computer Science & Engineering and Information Technology).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Student Name: Hari Priya Radhika Sharma
Roll No.: 201310

Student Name: Anoushka Sud
Roll No.: 201146

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Prof. Vivek Sehgal
Professor and Head,
Department of Computer Science & Engineering and Information Technology,
Jaypee University of Information Technology, Waknaghat.
Dated:

ACKNOWLEDGEMENT

Firstly, we express our heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the project work successfully.

We are really grateful and wish our profound indebtedness to Supervisor Prof. Vivek Sehgal, Professor and Head, Department of Computer Science & Engineering and Information Technology. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

We would like to express my heartiest gratitude to Prof. Vivek Sehgal, Professor and Head, Department of Computer Science &Engineering and Information Technology.

We would like to thank Dr. Salman Raju for his guidance and boundless support which helped us getting acquainted with new technologies. I would also like to thank Dr. Vikas Baghel, Mr. Dhinendraand Mr. Kamlesh Kumar Srivastava for their time and guidance throughout the project.

We would also like to thank each one of those individuals who have helped us straightforwardly or in a roundabout way in making this project a win. In this unique situation, Wemight want to thank the various staff individuals, both educating and non instructing, which have developed their convenient help and facilitated our undertaking.

Finally, we must acknowledge with due respect the constant support and patience of my parents and my friends.

Hari Priya Radhika Sharma(201310)
Anoushka Sud (201146)

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS, SYMBOLS OR NOMENCLATURE

CPS - Cyber-Physical Systems
HVAC - Heating, Ventilation, and Air Conditioning
RL - Reinforcement Learning
DHT22 - A specific type of sensor used for reading temperature and humidity data
xbee - A type of communication module
WSNs - Wireless Sensor Networks
IEEE - Institute of Electrical and Electronics Engineers
DAQ - Data Acquisition
LABVIEW - Laboratory Virtual Instrument Engineering Workbench
WirelessHART - Wireless Highway Addressable Remote Transducer
ISA - International Society of Automation
HART - Highway Addressable Remote Transducer
WIA-PA - Wireless Industrial Automation Personal Area Network
MQTT - Message Queuing Telemetry Transport
PLCs - Programmable Logic Controllers
IIoT - Industrial Internet of Things
RTU - Remote Terminal Unit
IoT - Internet of Things
SCADA - Supervisory Control and Data Acquisition
AI - Artificial Intelligence
IWSNs - Industrial Web Sensor Networks
RF - Radio Frequency
ANN - Artificial Neural Network
RBF-ANN - Radial Basis Function Artificial Neural Network
WPT - Wavelet Packets Transform
VAV - Variable-Air-Volume

FPGA - Field Programmable Gate Array

RTUs - Remote Terminal Units

CPU - Central Processing Unit

ML - Machine Learning

DRL - Deep Reinforcement Learning

DNN - Deep Neural Networks

POD - Probability of Detection

ARIMA - AutoRegressive Integrated Moving Average

CNN - Convolutional Neural Network

ALE - Arcade Learning Environment

SVM - Support Vector Machines

ARX - AutoRegressive with exogenous input

AC - Actor-Critic

MAE - Multiactor Networks Ensemble

GPU - Graphics Processing Unit

IDE - Integrated Development Environment

XCTU - Xbee Configuration and Test Utility

PPO - Proximal Policy Optimization

ABSTRACT

The integration of advanced technologies in Cyber-Physical Systems (CPS) has significantly enhanced various industrial and consumer applications, including Heating, Ventilation, and Air Conditioning (HVAC) systems. Conventional HVAC control systems, which are dependent on preprogrammed schedules and setpoints, frequently lead to poor performance, high energy usage, and restricted flexibility in response to changing environmental conditions. This project consists of two main parts, a simulation environment for Reinforcement Learning (RL) and developing a hardware prototype.

The CPS framework's physical foundation for data processing, actuation, and acquisition is provided by the hardware prototype. It consists of communication modules for wireless data transmission, microcontrollers for data processing, and sensors to monitor the surrounding environment. This configuration offers real-time information that is essential for optimizing HVAC systems.

Simultaneously, the dynamic interactions between the CPS and its surroundings are modeled by the RL simulation environment. The simulation environment uses reinforcement learning (RL) algorithms to help with decision-making and control strategies based on system constraints and environmental data. The RL agent optimizes HVAC management in response to shifting conditions and user demands by iteratively learning control policies.

The project is important because it has the potential to improve the flexibility, adaptability, and efficiency of traditional CPS models by overcoming some of their major limitations. This project advances the field of autonomous industrial systems and lays the groundwork for more productive and sustainable industrial operations, while also demonstrating the advantages of integrating RL with CPS. Reducing energy use, improving comfort, and fostering innovation in HVAC system optimization are the ultimate objectives.

CHAPTER 1: INTRODUCTION

1.1 INTRODUCTION

The integration of advanced technologies in Cyber-Physical Systems (CPS) has revolutionised various industrial and consumer domains, promising increased efficiency, automation, and adaptability. HVAC (heating, ventilation, and air conditioning) systems are among the many uses of CPS that are essential to maintaining comfort, efficiency, and energy conservation in constructed environments. However, traditional HVAC control systems frequently depend on preset schedules and predefined setpoints, which results in suboptimal performance, wasteful energy use, and restricted flexibility to changing environmental circumstances. This project addresses these issues by incorporating Reinforcement Learning (RL) approaches into a self-adaptive CPS framework, which presents a relatively new method of HVAC system control. By continuously responding to changing conditions and customer requirements, RL shows the potential to optimise HVAC operations in real-time.

The primary objective of this project is to create a working prototype of a self-adaptive CPS framework that can learn from its surroundings and optimise HVAC management strategies in response to system limitations and environmental data. This framework comprises both hardware and software components, including sensors for data collection, microcontrollers for data processing and actuation, communication modules for wireless data transmission, and an RL simulation environment for decision-making.

The project's significance lies in its potential to address key challenges faced by traditional CPS systems, including suboptimal efficiency, flexibility, and adaptability. The proposed methodology aims to use reinforcement learning (RL) to improve overall system performance, lower energy consumption, and increase comfort levels across a range of usage scenarios and environmental circumstances.

1.2 PROBLEM STATEMENT

Cyber-Physical Systems (CPS) play a pivotal role in modern industrial processes, seamlessly integrating physical components with computational systems to monitor, control, and optimise various operations. HVAC (heating, ventilation, and air conditioning) systems, which maintain ideal interior climatic conditions in commercial and industrial settings, are one of some famous application domains for CPS. Currently, the foundation of many CPS models follow deterministic approaches, which ignore the inherent uncertainties and dynamic character of actual industrial environments. As a result, these deterministic models frequently exhibit subpar performance and energy inefficiencies due to their inability to adjust to changing circumstances.

In the realm of autonomous industries, deterministic modelling approaches pose significant challenges. These sectors rely on flexible and effective systems to manage complex jobs with minimal assistance from humans. Deterministic models, on the other hand, are rigid and cannot account for the uncertainties and variations present in actual industrial processes in real time.

Enhancing CPS models using stochastic modelling techniques and reinforcement learning (RL) algorithms hold immense significance in addressing the challenges faced by autonomous industries. Increased ability to capture and adapt to the ever-changing and uncertain nature of industrial surroundings can be achieved by adding stochastic aspects to CPS models. This enhanced adaptability enables autonomous systems to make informed decisions in real-time, leading to improved performance, energy efficiency, and overall productivity. Furthermore, by highlighting the value of integrating latest technologies like reinforcement learning with the field of autonomous industrial systems advances and opens the door to future industrial operations that are more sustainable and productive.

1.3 OBJECTIVES

The primary objective of this project is to develop a prototype encompassing both hardware and software components for optimising Heating, Ventilation, and Air Conditioning (HVAC) systems, a prominent example of a self-adaptive Cyber-Physical System (CPS). This involves two main components: the hardware prototype and the RL simulation environment.

The hardware part serves as the physical infrastructure responsible for data acquisition, processing, and actuation within the CPS framework. It includes sensors, microcontrollers, and communication modules, all integrated to facilitate seamless interaction and data flow between all the components. The DHT22 sensor reads the data of environmental attributes, temperature and humidity. The microcontrollers process this data and communicate it wirelessly using communication modules, xbee modules. This hardware setup forms the foundation of the CPS framework, providing real-time insights into the environmental conditions crucial for HVAC system optimization.

On the other hand, the RL simulation environment provides a base for decision-making and control strategies based on environmental data and system limitations.RL simulation environment is developed using RL algorithms and software libraries. The simulation environment models the dynamic interactions between the CPS and its surroundings. Through iterative learning and adaptation, the RL agent within the simulation environment refines control policies to optimise HVAC management strategies in response to changing conditions and user requirements.

The hardware prototype and RL simulation environment represent distinct entities, they are intrinsically linked within the broader CPS framework. This integration will enable the CPS framework to adapt and optimise HVAC system operation autonomously, leading to improved energy efficiency, comfort levels, and overall system performance.

By developing these two components, our project aims to demonstrate the feasibility and effectiveness of integrating advanced technologies such as RL within self-adaptive CPS frameworks for HVAC system optimization.

1.4 SIGNIFICANCE AND MOTIVATION OF PROJECT WORK

The significance and motivation of this project work extend beyond mere technological innovation. It addresses fundamental challenges faced by contemporary industrial processes.It comes from the pressing need to address challenges inherent in traditional HVAC system optimization methods within industrial and commercial environments. In today's industrial landscape, sustainability and energy efficiency have emerged as critical concerns, necessitating the urgent need for innovative solutions.

As significant energy users in commercial and industrial buildings, HVAC systems are essential to reaching sustainability and energy efficiency goals. Nevertheless, conventional control approaches frequently fail to adjust to the changing and unpredictable industrial environments. Conventional control strategies often rely on preset schedules and fixed setpoints, leading to suboptimal performance and energy wastage. The significance of this project in developing a self-adaptive Cyber-Physical System (CPS) framework designed especially for HVAC system optimization is highlighted by this inadequacy.

In autonomous industries, where real-time performance and energy consumption optimization are critical, emphasises the importance of this project. The framework allows hardware prototypes to serve as a base for understanding CPSs and Reinforcement Learning (RL) to add the autonomous attribute to it. This integration can lead CPS systems the ability to respond quickly and adaptable to changing user preferences and environmental conditions.

The project's motivation is rooted in the pursuit of enhancing energy efficiency, improving comfort levels, and driving innovation in the field of HVAC system optimization, contributing to a more sustainable and resilient future for industrial operations.

1.5 ORGANIZATION OF PROJECT REPORT

CHAPTER 1: INTRODUCTION

Chapter 1 introduces the integration of advanced technologies in Cyber-Physical Systems (CPS) and their impact on various industrial and consumer domains. It highlights the importance of CPS in HVAC (heating, ventilation, and air conditioning) systems, which are crucial for maintaining comfort, efficiency, and energy conservation in built environments. The chapter discusses the limitations of traditional HVAC control systems and presents the project's objective of incorporating Reinforcement Learning (RL) into a self-adaptive CPS framework to optimize HVAC operations in real-time.

CHAPTER 2: LITERATURE REVIEW

Chapter 2 provides an overview of relevant research and existing solutions in the field of CPS and HVAC systems. It examines previous work on RL and its applications, identifying key gaps that this project aims to fill. The review highlights the limitations of current deterministic models and underscores the need for stochastic modeling techniques and RL algorithms to enhance CPS models' adaptability and efficiency.

CHAPTER 3: PROJECT DESIGN AND ARCHITECTURE

Chapter 3 details the development of the self-adaptive CPS framework for HVAC systems. It includes a comprehensive analysis of the requirements for both hardware and software components and describes the architecture of the CPS framework. The chapter explains how physical components like sensors, microcontrollers, and communication modules are integrated with advanced control strategies based on RL algorithms to optimize system performance.

CHAPTER 4: TESTING STRATEGY AND IMPLEMENTATION

Chapter 4 outlines the methodologies used to validate the project's components. It covers unit testing for hardware and software modules, integration testing to ensure seamless communication and data exchange, functional testing for accuracy and reliability, and performance testing to assess the effectiveness of RL algorithms in optimizing HVAC system control. This comprehensive approach ensures the system's reliability and responsiveness.

CHAPTER 5: RESULTS AND DISCUSSION

Chapter 5 presents the findings from the hardware prototype and RL simulation environment. It discusses the performance of the CPS framework in real-time HVAC optimization, highlighting its flexibility and responsiveness. The analysis covers the framework's scalability, robustness, and potential for practical applications. A comparison with existing solutions demonstrates the advantages of the proposed approach, particularly its ability to continuously learn and adapt, leading to improved energy efficiency and user comfort.

CHAPTER 6: CONCLUSION AND FUTURE WORK

Chapter 6 summarizes the project's outcomes and key findings, emphasizing the successful integration of RL techniques with hardware components to create a self-adaptive CPS framework. It acknowledges the project's limitations and discusses its contributions to CPS technology and HVAC system optimization. The chapter also outlines potential future work, including the integration of the hardware prototype with the RL simulation environment for real-time interaction, further research on refining RL algorithms, and testing in real-world industrial settings to enhance system reliability and efficiency.

CHAPTER 2: LITERATURE SURVEY

2.1 OVERVIEW OF RELEVANT LITERATURE

[1] The paper illuminates ZigBee as a standout transceiver standard for WSNs operating over the IEEE 802.15.4 specification. The paper provides a thorough analysis of ZigBee wireless technology, elucidating its key features, specifications, and architectural components, such as its protocol stack architecture and underlying communication mechanisms. With a low throughput of approximately 250kbps, ZigBee finds use cases in scenarios demanding low data rates, as evidenced by various examples such as greenhouse monitoring, multi-level parking vacancy monitoring, intelligent warehouse management, and environmental monitoring. Additionally, ZigBee's versatility extends to standard applications like home automation, automatic meter reading, building automation, personal healthcare, and automotive solutions, underlining its pivotal role in enabling efficient and reliable wireless communication for a broad range of monitoring and control applications across diverse domains.

[2]In order to enable remote monitoring of crucial parameters like length filtering, ground vibration sensing, and electricity monitoring, the study introduces a method employing ZigBee embedded systems for real-time industrial measurements and monitoring. By offering centralized control and real-time monitoring capabilities, this method enhances safety and improves operational efficiency. Through the use of digital signal conversion from analog signals from several sensors, the system architecture allows for both wireless and wired transmission via DAQ and ZigBee, allowing for synchronized measurement and monitoring. By guaranteeing the smooth integration and centralized management of sensor signals, the LABVIEW program design lowers labor costs, boosts operational effectiveness, and improves overall safety in industrial settings.The study concludes, the integration of ZigBee embedded systems holds immense potential to improve efficiency and safety in a variety of industrial contexts by delivering real-time insights and enabling preventative safety measures.

[3] The research paper offers a thorough analysis of four prominent industrial wireless sensor network standards: ZigBee, WirelessHART, ISA100.11a, and WIA-PA, focusing on their design, protocol architectures, uniqueness, and suitability for industrial applications. The paper clarifies the unique features, constraints, and architecture of every standard, highlighting how crucial it is to match particular needs with network attributes in order to make well-informed decisions. Although WirelessHART, ISA100.11a, and WIA-PA are specifically designed for process automation and address issues with ZigBee, ZigBee is still preferred for low-traffic and noncritical applications because of its simplicity and lower power consumption. The paper lists the technical similarities and differences between the standards,highlightingWirelessHART's suitability for integration with existing HART control systems and ISA100.11a's flexibility but increased complexity. Additionally, it addresses upcoming obstacles that hinder wireless solutions from being widely adopted in industrial settings and suggests solutions that would hasten deployment and acceptance.

[4] In this research paper, industrial communication protocols are examined with a specific emphasis on the application of the Message Queuing Telemetry Transport (MQTT) protocol to PLCs in industrial environments. The goal of the study is to increase factory efficiency by addressing the need for better PLC inter-communication within the Industrial Internet of Things (IIoT) framework. With the help of Siemens gateways Simatic IoT2020 that use the Modbus Remote Terminal Unit (RTU) protocol, the study models an experimental setup in which two PLCs control an industrial plant. The study emphasizes how important it is to address the communication issues facing PLCs and suggests MQTT as a workable substitute for fieldbus protocols currently in use.The paper underscores the significance of tackling PLCs' communication challengesand suggests MQTT as a workable substitute for fieldbus protocols currently in use. In its concluding remarks, the paper acknowledges the feasibility of employing Modbus RTU and MQTT for PLC interconnection while highlighting the necessity for supplementary tools to streamline PLC variable updates when employing MQTT. In addition, it recommends that future studies investigate additional industrial protocols that are compatible, like OPC UA, in order to improve system response time and automation application flexibility.

[5] The project focuses on integrating Internet of Things (IoT) technology into industrial processes to enable real-time monitoring and intelligent decision-making, aiming to enhance traditional industrial systems reliant on Programmable Logic Controllers (PLCs) and Supervisory Control and Data Acquisition (SCADA) systems. The project proposes developing a web-based real-time PLC data monitoring system to automatically monitor industrial applications, generate alerts or alarms, and make intelligent decisions. The integration of artificial intelligence (AI) and the Internet of Things (IoT) is crucial to this project because it allows for intelligent monitoring and control through the analysis of data from sensors and actuators to identify anomalies, anticipate failures, and maximize performance. The project's ultimate goal is to provide a comprehensive solution for industrial appliance monitoring and control, improving daily operations' performance while lowering manual labor costs and raising operational efficiency.

[6] The study emphasizes the critical need for smart and cost effective industrial automation solutions in light of the mounting pressure on businesses to increase operational effectiveness, while adhering to environmental regulations, and meet financial goals. The paper explores the design principles and technical difficulties that underlying the creation of IWSNs, scratching on subjects like standards, energy-harvesting methods, radio technologies, and cross-layer design. The study explores several unexplored open research topics, such as effective deployment models, analytical performance assessment, sensor-node deployment optimization, localization, security, and interoperability issues. Additionally, handling RF interference and dynamic wireless-channel conditions in industrial settings presents significant challenges that could be resolved by cognitive radio paradigms and channel hand-off mechanisms. Through enabling self-organization, rapid deployment, adaptability, and built-in intelligent processing capabilities, Industrial Web Sensor Networks (IWSNs) support extremely dependable and self-repairing industrial ecosystems that react quickly to events occurring in real time. The paper seeks to contribute to to the realization of intelligent and effective industrial automation systems by advancing the field of Industrial Wireless Sensor Networks, encouraging innovation, and educating decision-making processes.

[7] This study investigates how wireless sensor networks (WSNs) can revolutionize data collection procedures for both terrestrial and spaceflight applications. WSNs provide benefits like scalability, easy sensor installation in difficult-to-reach places, and weight and cost savings by doing away with the need for large wiring trunks. The real-world implementation of WSNs poses certain challenges though, which new developments in standards-based protocols designed for industrial control applications—like ISA100.11a and WirelessHART aim to address. The paper gives a summary of these protocols, highlighting their salient characteristics and benefits for industrial applications. It also presents the architecture of a new standards-based sensor node intended to support applications research and networking in the framework of these protocols. The article aims to offer insights into resolving real-world issues related to WSN deployments by utilizing standardized WSN protocols. This will eventually open the door to better data collection capabilities and increased productivity in a variety of fields.

[8] The research paper introduces a novel methodology for crack detection in rotating shafts, employing a combination of Wavelet Packets transform energy and Artificial Neural Networks with Radial Basis Function architecture (RBF-ANN). The study aims to maximize the success rates of the detection approach by analyzing vibration signals obtained from a rig under different crack conditions. The 'Daubechies 6' wavelet function is used to process experimental data with varying speeds and crack conditions using Wavelet Packets transform energy and the resultant energies are then used to train multiple RBF-ANNs. The study achieves promising results by fine-tuning parameters of both the ANNs and the WPT decomposition level. Notably, features extracted at a specific decomposition level from signals acquired at higher speeds offer optimal performance. With a minimal false alarm rate of 1.77% and detection probabilities approaching 100%, Probability of Detection (POD) curves show reliable detection even at crack levels as low as 4% of the shaft diameter. The suggested methodology promises improved reliability in crack detection to avert catastrophic failures and lower expensive repairs. It also has the potential to be integrated into industrial equipment for improved condition monitoring.

[9] This study critically reviews prevailing modeling techniques in HVAC systems, aiming to assess their applicability, practical acceptance, strengths, weaknesses, applications, and performance. Through a comprehensive analysis, the study provides insights into the efficacy and outcomes of developed models deployed in real-world HVAC systems, highlighting inherent shortcomings in nearly every approach.The study also highlights the necessity of enhancing the performance of building HVAC systems by illuminating these limitations. Overall, it emphasizes how important modeling methods are to maximizing HVAC performance and energy economy, providing insightful information to guide future studies and improve the effectiveness of HVAC control strategies in practical settings.

[10] With an emphasis on Variable-Air-Volume (VAV) systems, the study attempts to improve the modeling efficiency of HVAC systems by creating modular models for HVAC components using the state-space method. These modular models provide a uniform representation of relationships among input, state, and output signals. They are based on state-space modeling and graph theory, and have been validated through transient response experiments. This approach enables the efficient development of dynamic models for real air-conditioning systems. The study also looks into the use of an ARIMA-based predictive Proportional-Integral (PI) controller for controlling room temperature, demonstrating superior performance compared to traditional PI controllers through model simulations and experimental validation.

[11] This study explores the design of Supervisory Control and Data Acquisition (SCADA) systems for energy management in areas experiencing energy shortages. Specifically, the design of Remote Terminal Units (RTUs) is examined, as these are an essential component that enable data transfer between field devices and the SCADA system. The research suggests two different RTU designs for third-generation SCADA systems: one that uses a Field Programmable Gate Array (FPGA) as the CPU and the other that uses a Programmable Logic Controller (PLC) as the CPU.The paper shows the superiority of FPGA-based RTUs through a

comparative analysis, providing improved features like support for encryption, radio support, and increased memory capacity. The performance of each design is assessed with the use of simulation tools, and FPGA-based RTUs turn out to be the best option overall, especially for energy management applications that call for the deployment of wireless SCADA. In addition, the study explores the optimization of wireless communication links connected to RTUs, addressing factors such as RF spectrum utilization to make it easier to design an optimal wireless link for the deployment of low-cost RTUs. Overall, the suggested FPGA-based RTU design offers a more potent and reconfigurable solution for wireless SCADA execution, overcoming the drawbacks of traditional PLC-based systems and showing promise for deployment in energy-deficient areas to improve energy distribution and management efficiency.

[12] The paper shows the superiority of FPGA-based RTUs through a comparative analysis, providing improved features like support for encryption, radio support, and increased memory capacity. The performance of each design is assessed with the use of simulation tools, and FPGA-based RTUs turn out to be the best option overall, especially for energy management applications that call for the deployment of wireless SCADA. In addition, the study explores the optimization of wireless communication links connected to RTUs, addressing factors such as RF spectrum utilization to make it easier to design an optimal wireless link for the deployment of low-cost RTUs. Overall, the suggested FPGA-based RTU design offers a more potent and reconfigurable solution for wireless SCADA execution, overcoming the drawbacks of traditional PLC-based systems and showing promise for deployment in energy-deficient areas to improve energy distribution. The paper highlights the potential integration of deep learning into cloud computing infrastructure for more convenient and on-demand computational services in smart manufacturing applications. It also identifies emerging research efforts, future trends, and challenges associated with deep learning in manufacturing processes. These developments promise to further advance deep learning capabilities within the manufacturing domain.

[13] The paper highlights how manufacturing businesses must change to keep up with the intricacies of contemporary technology and competitive environments, which calls for a move toward digitalization and intelligence in automated manufacturing. The revolutionization of industrial processes can be attributed to the potential of artificial intelligence (AI) and machine learning (ML), specifically reinforcement learning (RL). The paper aims to analyze reinforcement learning's performance in industrial contexts and identify opportunities and challenges for further integrating it into automated manufacturing through a thorough review of recent practical applications. The study performs a hierarchical analysis of RL's application across a variety of industrial domains, from industrial process systems to human-machine interaction and process monitoring, drawing on extensive literature reviews. The study offers valuable insights into the changing field of intelligent manufacturing technologies by clarifying RL's applicability and potential to improve the effectiveness and control of automated manufacturing systems.

[14] The present study delves into the use of Deep Reinforcement Learning (DRL) in the context of smart manufacturing, with a focus on how it can revolutionize different phases of the manufacturing process. By fusing Reinforcement Learning (RL) and Deep Neural Networks (DNN), DRL provides a flexible and dynamic approach to accurately making decisions in intricate manufacturing settings. The review follows the development of DRL from its launch in 2013 to the present, highlighting its rapid expansion in manufacturing applications and pointing out a lack of a thorough analysis specifically focused on smart manufacturing in the literature. It also offers insights into DRL's role throughout the engineering lifecycle, from design to maintenance, and highlights typical applications of DRL at each stage by analyzing 261 relevant publications. It also explores new technologies and approaches to improve deployment viability and learning efficiency, as well as challenges and future directions for DRL in smart manufacturing. Through tackling these obstacles and opportunities, the review seeks to guide future investigations and expedite the integration and progression of DRL in intelligent manufacturing, providing an invaluable asset for scholars and professionals operating within the domain.

[15] The paper emphasizes how challenging it is to apply reinforcement learning to practical situations, especially when trying to represent complex environments effectively from high-dimensional sensory inputs. A novel artificial agent known as a deep Q-network (DQN) has been created in response to this challenge thanks to recent developments in deep neural networks. Unlike earlier methods, DQN utilizes end-to-end reinforcement learning to learn effective policies directly from high-dimensional sensory inputs, which allows it to perform exceptionally well in domains with intricate, unstructured input spaces, like vintage Atari 2600 games. Using the same algorithm, network architecture, and hyperparameters across a diverse set of 49 games, leveraging DQN outperforms prior algorithms and yields results on par with expert human testers. This groundbreaking work represents a significant milestone in the field of artificial intelligence research, demonstrating not only the ability of DQN to bridge the gap between high-dimensional sensory inputs and actions, but also the potential of deep reinforcement learning agents to master difficult tasks without the need for handcrafted features or fully observed state spaces, thereby opening up new avenues for the solution of challenging real-world problems.

[16] In this paper, a novel family of policy gradient methods for reinforcement learning is presented: Proximal Policy Optimization (PPO). PPO allows for multiple epochs of minibatch updates to enhance sample complexity and performance. It does this by alternating between optimizing a surrogate objective function through stochastic gradient ascent and sampling data from the environment. In contrast to other current strategies like vanilla policy gradient methods and deep Q-learning, PPO uses only first-order optimization to achieve the dependable performance and data efficiency of trust region methods. The key to PPO's success is a novel objective that uses clipped probability ratios to provide stable and reliable optimization and a pessimistic estimate of policy performance. By combining the stability and reliability of trust region methods with the ease of use of first-order optimization, the paper shows that PPO performs better than previous algorithms on both continuous control tasks and

Atari game playing. PPO is designed to address challenges related to scalability, data efficiency, and robustness.

[17] This paper presents the first deep learning model that can learn control policies directly from high-dimensional sensory input, marking a revolutionary development in reinforcement learning (RL). This study focuses on applying a convolutional neural network (CNN) trained with a variant of Q-learning to seven Atari 2600 games from the Arcade Learning Environment (ALE), with notable success. The model uses raw pixels as input and a value function estimates future rewards. The primary innovation is CNN's capacity to surmount difficulties in learning control policies from unprocessed video data in intricate reinforcement learning environments, without necessitating game-specific data or manually-crafted visual elements.The results demonstrated how well the CNN model can learn challenging control policies for Atari 2600 computer games, outperforming earlier RL algorithms on six of the seven games it was tested on and even outperforming a skilled human player on three of them. Additionally, the study presents an improved version of online Q-learning that uses experience replay memory and stochastic minibatch updates to improve deep network training for reinforcement learning. Together, these methods produce cutting-edge outcomes in a variety of games without requiring changes to the model's architecture or hyperparameters. In general, the paper represents a noteworthy advance in reinforcement learning research, showcasing the capacity of deep learning models to acquire control policies straight from unprocessed sensory data, creating novel opportunities for resolving intricate RL assignments across multiple fields.

[18] The study offers an overview of the literature on modeling techniques for HVAC (heating, ventilation, and air conditioning) systems, which is essential for comprehending and optimizing energy usage in these systems. Three broad categories of modeling are outlined in the review: grey box models, physics-based models, and data-driven models. Data-driven approaches, such as frequency domain models, data mining algorithms like Artificial Neural Networks (ANN) and Support Vector Machines (SVM), and statistical models like AutoRegressive with exogenous input (ARX) and AutoRegressive Integrated Moving Average

(ARIMA), use measurement data of input and output variables to approximate system behavior through linear and nonlinear functions. When compared to data-driven models, physics-based models have better generalization capabilities since they are based on an understanding of the underlying physical laws and processes. Grey box models use measured data for parameter estimation and physical laws to define the model structure, combining elements of data-driven and physics-based methods. Model effectiveness is assessed using performance comparison metrics and qualitative factors like robustness, ease of tuning, strengths, and weaknesses.

[19] The research paper frames the problem as a continuous state, continuous action reinforcement learning (RL) problem and offers a novel decision-making framework specifically designed for operational indices in the process industry. The paper enriches decision policy learning by introducing a model-free RL algorithm and using an actor-critic (AC) framework with a multiactor networks ensemble (MAE) approach. In contrast to traditional techniques, this strategy uses experience replay and stochastic policy to avoid local optima and address data scarcity problems in reinforcement learning. The efficacy of the suggested algorithm is demonstrated in the paper via extensive simulation studies on actual data from a mineral processing plant, showing improvements in production yield and learning speed while avoiding local optima. Three main contributions are the creation of an efficient, quickly converging policy for decision-making at low cost, the use of experience replay to enhance data utilization, and the application of the MAE-AC framework to solve local optima problems. It is recommended that future research focus on developing advanced controllers that can adapt to dynamic environments and strengthening the security of reinforcement learning algorithms.

[20] This work addresses the crucial problem of real-time scheduling in networks of gas supply for multiple products, concentrating on steel companies where the availability of gas is essential to their manufacturing processes. The study aims to maximize resource utilization and operational efficiency in these networks using a reinforcement learning (RL) framework. A new reinforcement learning approach is put forth that includes prediction and safety

modules that use process knowledge to foresee state changes and stop risky behaviors. By penalizing the agent for unsafe actions, the safeguard module speeds up training and lowers trial-and-error costs. The effectiveness of the suggested RL method is validated by case studies on real gas supply networks, which demonstrate superior training speed and market adaptability through transfer learning. Key innovations include integrating process knowledge to ensure operational safety and offering streamlined online computation for real-time scheduling demands. Subsequent investigations could examine approaches customized for swiftly evolving environmental circumstances and tackle discrepancies between scheduling models and uncertainties in industrial processes.

[21] The paper tackles the problem of inconsistent performance evaluations resulting from flawed evaluation metrics, which makes a substantial contribution to the field of reinforcement learning (RL) research. In an effort to remove inadvertent biases frequently seen in research settings, the authors present a thorough evaluation methodology intended to generate accurate performance measurements for reinforcement learning algorithms. By establishing high-confidence bounds over the evaluation process, their method makes it easier to compare RL algorithms fairly. The authors also show how effective their methodology is by evaluating different RL algorithms on common benchmark tasks. The proposed framework has several important features, such as a fair comparison approach based on principles and implementations to help other researchers use this methodology. All things considered, the work greatly improves the consistency and repeatability of performance assessments in RL research.

[22] This work presents an extensive collection of metrics intended to quantify aspects of reliability, in particular variability and risk, during the training and post-learning phases. These flexible metrics are intended to support thorough comparisons between various scenarios, along with complimentary statistical tests. The paper highlights these metrics' characteristics and emphasizes how well-suited they are for a variety of situations. It also offers helpful guidelines for results reporting in order to create a uniform framework for assessing the reliability of RL algorithms. Additionally, these metrics and statistical tools are provided by

the study as an open-source library, which increases accessibility and promotes wider adoption in the RL community. The study shows the effectiveness of these metrics in revealing the strengths and weaknesses of different RL algorithms and environments through empirical application.

[23] Although reinforcement learning (RL) has shown to be remarkably effective at handling difficult decision-making tasks, its application in practical settings raises safety concerns, especially in fields like robotics and autonomous driving. The work investigates safe RL algorithms, which are presently in their infancy, as a solution to this. This paper offers a thorough analysis of safe reinforcement learning, including theory, methods, and applications. It lists five critical issues—referred to as "2H3W"—for safe RL deployment and evaluates how far these problems have come. Applications to autonomous driving and robotics are covered, as well as sample complexity and convergence of safe reinforcement learning techniques. In an effort to promote research in this area, the study also introduces a benchmark suite and an open-sourced repository with implementations of significant safe RL algorithms.

[24] Reinforcement Learning (RL) is a state-of-the-art machine learning method that mimics the human and animal trial-and-error learning processes to enable sequential decision-making in complex scenarios. RL agents learn optimal policies on their own by continuously interacting with stochastic environments. They have demonstrated amazing abilities, such as learning video games from pixel data alone. For researchers who are new to the field of reinforcement learning, this review offers an extensive introduction to the subject matter, including basic concepts, important techniques, and a wide range of applications. The review emphasizes the interdisciplinary significance of reinforcement learning (RL) across multiple domains by discussing state-of-the-art deep reinforcement learning (DRL) algorithms alongside conventional RL approaches. But there are still issues, mainly with comprehending and improving the interpretability of DRL algorithms, adapting RL techniques to practical problems, and meeting computational demands. Stability, convergence, scalability, and safety are important issues to address as reinforcement learning (RL) develops in order to advance the field and realize its maximum potential in solving real-world problems.

[25] This paper lists nine key obstacles that must be overcome before RL can be effectively used in real-world situations. Every challenge is carefully outlined along with current methods from the literature and recommended assessment criteria. If these issues are resolved in their entirety, RL may become useful for a variety of real-world situations. The paper also provides an example domain that has been modified to include these difficulties, acting as a useful testbed for RL research. The goal of the paper is to close the gap between theoretical developments and practical applications of reinforcement learning by highlighting these obstacles and providing recommendations for researchers and practitioners.The paper highlights the significance of model-based reinforcement learning (RL), ensembles, and expert-human cooperation in addressing these obstacles and clearing the path for the implementation of RL systems in various real-world products and systems.

2.2 KEY GAPS IN THE LITERATURE

The above literature can be divided into four sub-sections, each section focusing on a different aspect of the project.
The first section focuses on Zigbee, PLC, embedded systems and inter-communications between them in industrial set-up. The second section focuses on the modelling techniques for HVAC. The third section, on the other hand, focuses on deep learning, reinforcement learning, deep reinforcement learning and the policies and opportunities present in them. And the last section focuses on the evaluation strategies for reinforcement learning.

Below are the main gaps identified while exploring the above literatures:

- **Lack of proper evaluation metrics for RL**
  Inconsistent performance evaluations are caused by the absence of comprehensive and standard evaluation metrics for reinforcement learning algorithms. In order to guarantee fair comparisons and remove inadvertent biases, suggested solutions seek to define high-confidence bounds over the evaluation procedure. To further improve and standardize these evaluation techniques, more study is necessary.

- **Lack of a standard modelling technique that can represent complex nature of HVAC systems**
  Comprehensive and standardized modelling techniques that are capable of accurately representing the complex dynamics of HVAC systems are lacking in the HVAC industry. The suggested approaches centre on fusing graph theory and state-space modelling to create modular HVAC component models that facilitate effective dynamic system modelling. To validate these models in actual HVAC systems and investigate cutting-edge control algorithms for better system performance, more research is necessary.

- **Unexploited potential of DRL and DL in smart manufacturing**
  Although DRL has the potential to completely transform smart manufacturing, there

aren't many thorough reviews that are especially focused on DRL applications in this field. By offering a methodical analysis of DRL's function throughout the engineering lifecycle of smart manufacturing, suggested solutions seek to close this gap. Nevertheless, additional investigation is required to tackle obstacles and investigate cutting-edge technologies in order to enhance the practicability and effectiveness of DRL in manufacturing systems. Additionally, An exhaustive examination of techniques and efficacy is lacking when it comes to the use of deep learning algorithms in smart manufacturing. Solutions that are suggested are meant to offer a survey of this kind, emphasizing how deep learning can be used to optimize different phases of the manufacturing process. More investigation is necessary to address issues with deployment viability and learning efficiency in manufacturing systems, as well as to investigate more sophisticated deep learning approaches.

CHAPTER 3: SYSTEM DEVELOPMENT

3.1 REQUIREMENTS AND ANALYSIS

Hardware Requirements:

DHT22 Sensor	For measuring temperature and humidity
Arduino Uno	Microcontrollers for data processing and interfacing with sensors and communication modules
XBee S2C Modules	For wireless data transmission between nodes
Jumper Wires	To establish electrical connections between components
Programmable Logic Controller (PLC)	For controlling HVAC system actuators based on sensor data
Xbee Adapters, Breadboard & USB Cables	To facilitate the hardware connections

Table 3. 1 Hardware requirements

Software Requirements:

Google Colab	It compensates for the GPU and high computing requirements.
Python (3.10)	Programming language
OpenAI Gym	For simulating the HVAC model to train the agent
Stable Baselines	To use RL agent training policies
Programmable Logic Controller (PLC)	For controlling HVAC system actuators based on sensor data
Zigbee Communication Protocol	For data transmission between Xbee Modules
Arduino IDE	For compiling and uploading Arduino Sketch in Arduino Uno
XCTU Software	For configuring xbee modules for successful wireless data transmission.

Table 3. 2 Software requirements

3.2 PROJECT DESIGN AND ARCHITECTURE

The project focuses on the development of a self-adaptive Cyber-Physical System (CPS). For better understanding and focused work optimising Heating, Ventilation, and Air Conditioning (HVAC) systems is considered. The architecture of the CPS framework is designed to seamlessly integrate hardware components with advanced control strategies, leveraging Reinforcement Learning (RL) algorithms to achieve optimal HVAC system performance.

At its core, CPS framework consists of two main components: hardware and software.
The hardware component encompasses the physical components involved in data gathering and processing, while the software component comprises the algorithms and simulation environment responsible for system optimization.

Important parts of the hardware component are sensors, microcontrollers, and communication modules. The primary sensor used to collect temperature and humidity data from the surroundings is the DHT22 sensor. The microcontroller Arduino Uno serves as the data processing unit and receives this data. Jumper wires enable communication between the microcontroller and sensor, resulting in a smooth transfer of data.The processed data is then wirelessly transferred by the Arduino Uno microcontroller to a slave XBee module. Data is sent by the slave XBee module to the master XBee module, which then relays it to an additional Arduino Uno microcontroller that is linked to a PLC (programmable logic controller). For control decision-making, real-time data transmission from the sensor to the PLC is made possible by this hierarchical communication architecture.

To maximise HVAC system performance, sophisticated control techniques are used on the software side. RL libraries like TensorFlow or PyTorch, along with the Python programming language, are used to implement the RL simulation environment
Sensor data is received by the hardware parts, processed, and then wirelessly transmitted. The RL agent in the simulation environment uses testdata to make control decisions. The CPS framework can adjust and optimise HVAC system operation in real-time, balancing energy efficiency.

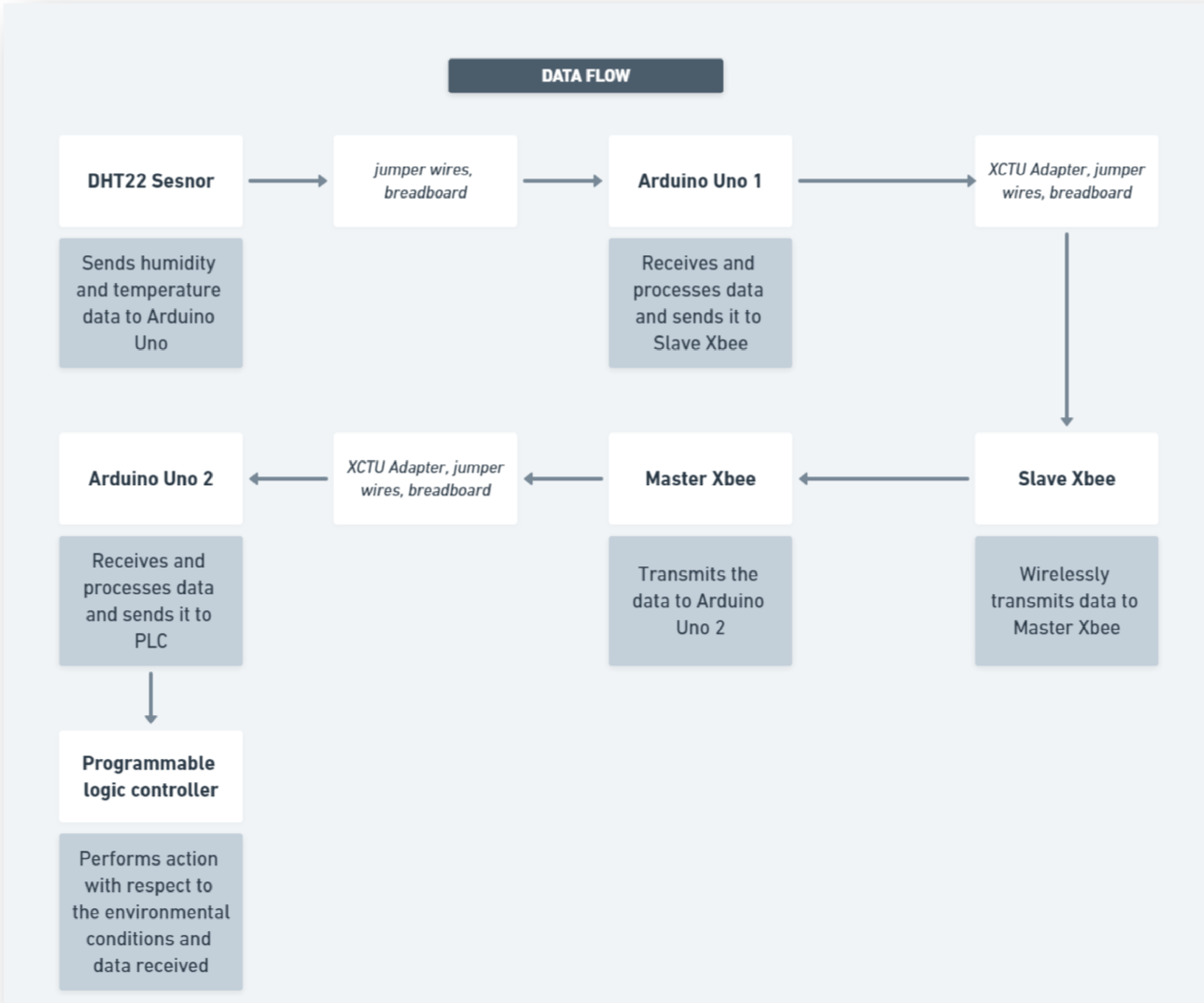

Figure 3. 1 Data flow diagram between hardware components

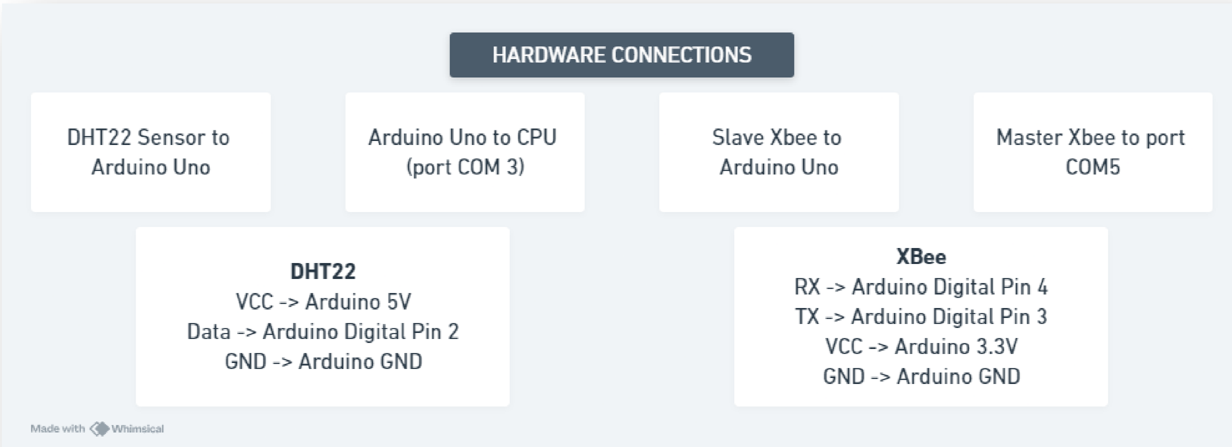

Figure 3. 2 Hardware Connections

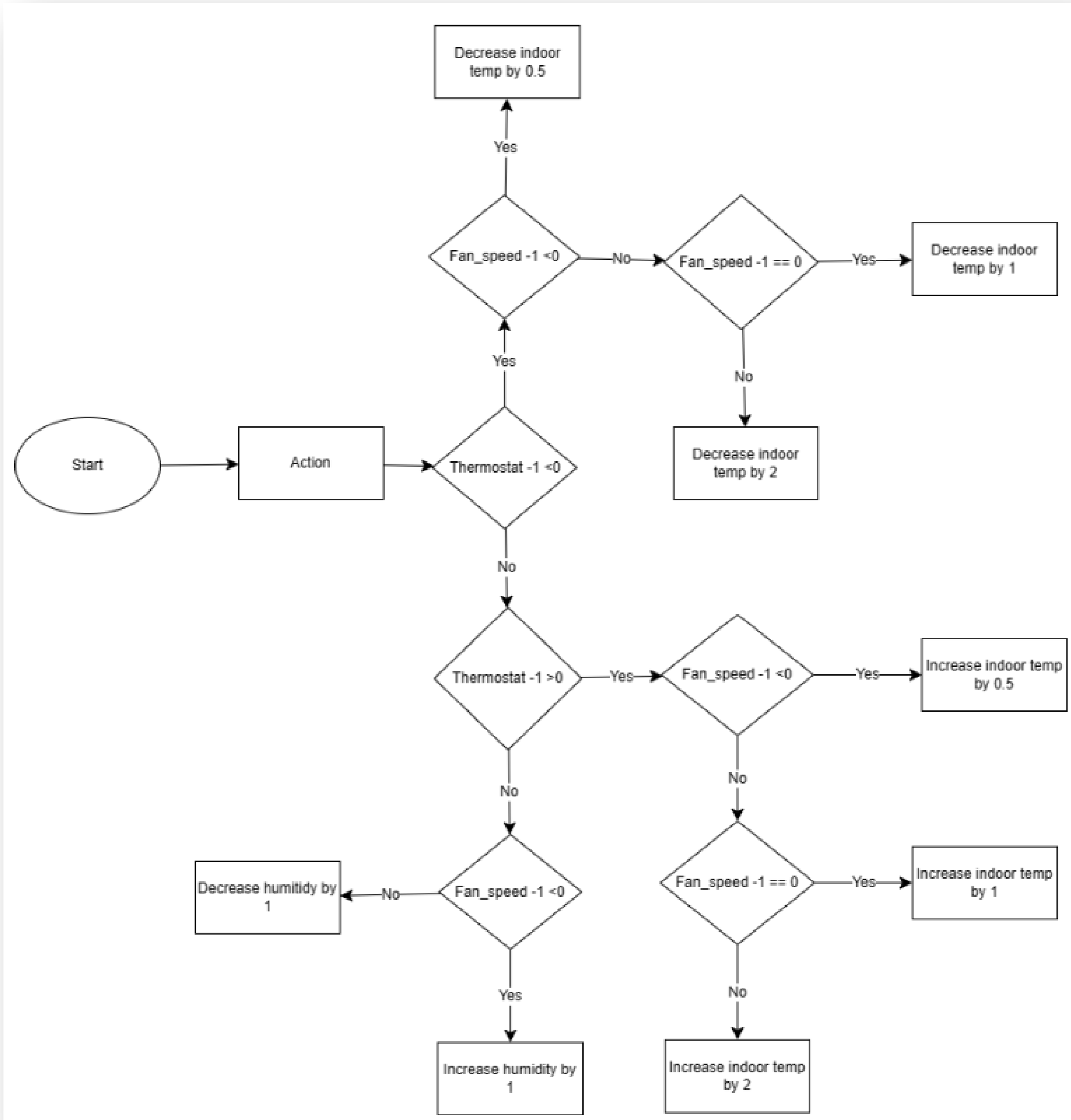

Figure 3. 3 Action Space of the RL model

3.3 DATA PREPARATION

Status	Humidity (%)	Temperature (C)	(F)
OK	26.0	24.0	75.2
OK	26.0	24.0	75.2
OK	26.0	24.0	75.2
OK	26.0	24.0	75.2
OK	26.0	24.0	75.2

Figure 3. 4 Real time data generated by DHT22 sensor

OK	26.0	24.0	75.2
OK	28.0	24.0	75.2
OK	30.0	24.0	75.2
OK	30.0	24.0	75.2
OK	31.0	24.0	75.2

Figure 3. 5 Real time data generated by DHT22 sensor

OK	34.0	24.0	75.2
OK	37.0	24.0	75.2
OK	39.0	24.0	75.2
OK	43.0	24.0	75.2
OK	44.0	25.0	77.0
OK	47.0	25.0	77.0
OK	48.0	25.0	77.0
OK	49.0	25.0	77.0
OK	51.0	25.0	77.0
OK	53.0	25.0	77.0

Figure 3. 6 Real time data generated by DHT22 sensor

OK	57.0	25.0	77.0
OK	54.0	25.0	77.0
OK	52.0	24.0	75.2
OK	52.0	24.0	75.2
OK	51.0	24.0	75.2
OK	50.0	24.0	75.2

Figure 3. 7 Real time data generated by DHT22 sensor

```
#include <DHT.h>
#define DHTPIN 2
#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600);
  dht.begin();
}

void loop() {
  Serial.println("Hari Sud");
  delay(2000);

  float humidity = dht.readHumidity();
  float temperature = dht.readTemperature();

  if (isnan(humidity) || isnan(temperature)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  if (isnan(humidity) || isnan(temperature)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
  xbeeSerial.print("H:");
  xbeeSerial.println(humidity);
  xbeeSerial.print("T:");
  xbeeSerial.println(temperature);
}
```

Figure 3. 8 Arduino Sketch for getting data from DHT22 Sensor and further processing it.

3.4 IMPLEMENTATION

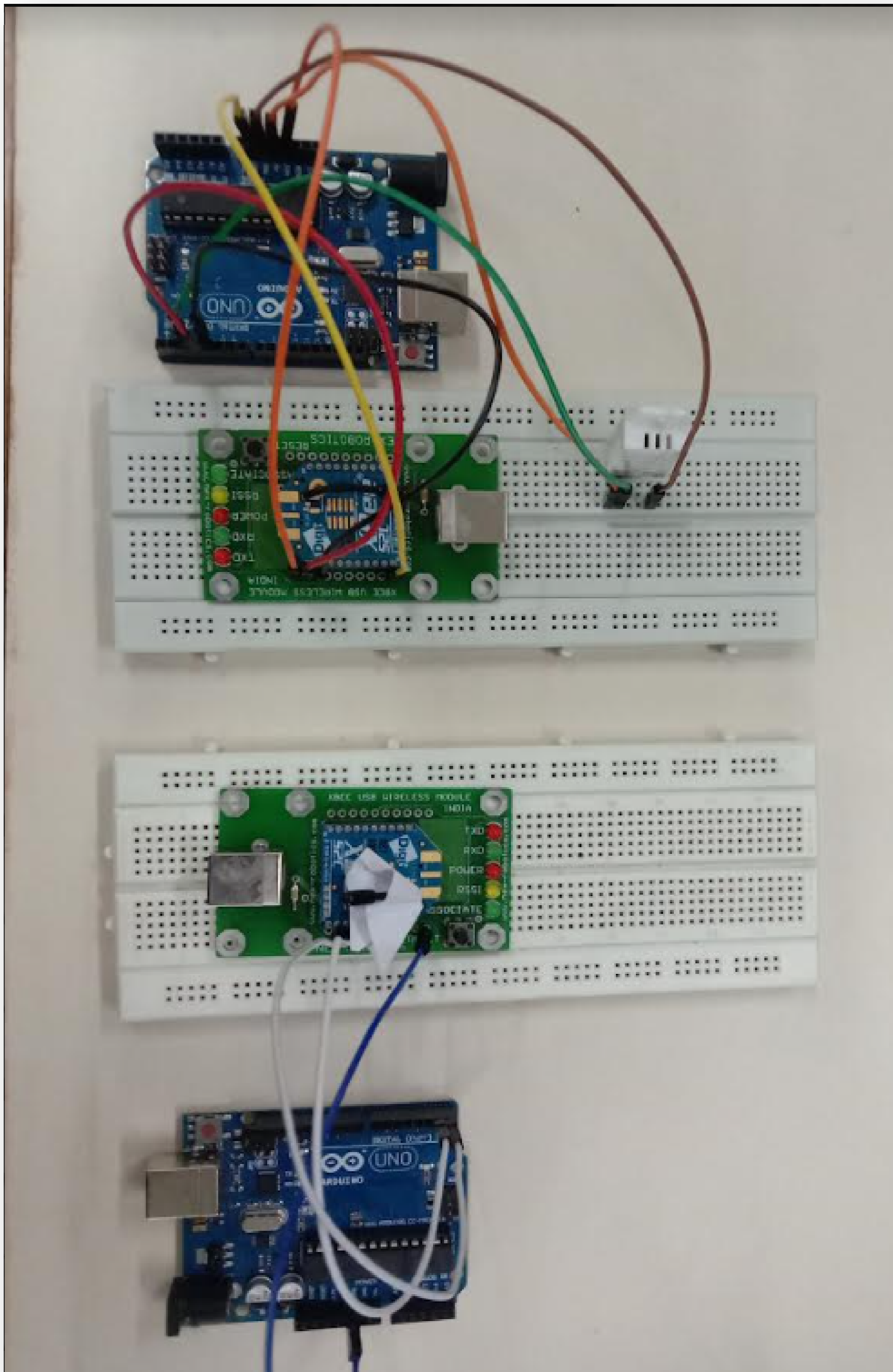

Figure 3. 9 Hardware connections between Arduino Uno and Xbee

Figure 3. 10 Tangible connections including Arduino UNo, DHT11 Sensor, Relay, DC power Supply, PLC

```
MasterSlave §
#include <DHT.h>
#include <SoftwareSerial.h>
#define DHTPIN 2
#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);
SoftwareSerial xbeeSerial(3, 4); // RX, TX

void setup() {
  Serial.begin(9600);
  xbeeSerial.begin(9600);
  dht.begin();
}

void loop() {
  Serial.println("Hari Sud");
  delay(2000); // Wait 2 seconds between measurements

  float humidity = dht.readHumidity();
  float temperature = dht.readTemperature();
```

Figure 3. 11 Arduino Sketch for reading data from DHT22 Sensor and sending it to Slave Xbee (a)

```
  if (isnan(humidity) || isnan(temperature)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
Serial.println(humidity);
  // Send data to XBee module
  xbeeSerial.print("H:");
  xbeeSerial.print(humidity);
  xbeeSerial.print("\n");
  xbeeSerial.println(temperature);
}
```

Figure 3. 12 Arduino Sketch for reading data from DHT22 Sensor and sending it to Slave Xbee (b)

```
#include <SoftwareSerial.h>
SoftwareSerial xbeeSerial(2, 3);
void setup() {
  Serial.begin(9600);
  xbeeSerial.begin(9600);
}

void loop() {
  delay(2000);
  if (xbeeSerial.available()) {
    char data = xbeeSerial.read();
    Serial.print(data);
  }
  else
  {
    Serial.print("::::");
  }
}
```

Figure 3. 13 Arduino Sketch to read data from Master Xbee

Considering the observation space:

- indoor temp: -30 -60 (14 - 30 degree celsius)
- outdoor temp: -30 - 60
- humidity: 0 - 100 (30-50%)
- energy consumption: 0-5 kwh(considering well insulated infrastructure)
- thermostat: 14-32 degree celsius
- fan speed: 200 and 2500 cubic feet of air per minute

Considering the action space:

- thermostat setting
- fan speed

Using standard valuesfor reward function

Figure 3. 14 The inticacies of the model

```
!pip install stable-baselines3[extra]
```

Show hidden output

```
# importing liberaries
import gym
from gym import Env
from gym.spaces import MultiDiscrete, Box

import numpy as np
import random
import os

from stable_baselines3 import PPO
from stable_baselines3.common.vec_env import DummyVecEnv
from stable_baselines3.common.evaluation import evaluate_policy
```

Figure 3. 15 Importing libraries

```
class HVACEnv(Env):
    def __init__(self):
        # 3 type of actions inc, stay same, dec: 2 parameters thermostat setting, fan speed
        # 6 parameters in observation space
        # indoor_temp, outdoor_temp, humidity, energy_consumption, thermostat setting, fan_speed
        self.action_space = MultiDiscrete([3,3])
        self.observation_space =
        Box(low=np.array([-30, -30, 0, 0, 14, 200]), high=np.array([60, 60, 100, 5000, 32, 2500]))

        self.indoor_temp =  25 + random.randint(-3,3)
        self.outdoor_temp = 14 + random.randint(-12,12)
        self.humidity = 40 + random.randint(-10,10)
        self.energy_consumption = 0
        self.thermostat_setting = 25 + random.randint(-3,3)
        self.fan_speed = 400 + random.randint(-200,200)
        self.time_stamp = 10
```

Figure 3. 16 Defining the training environemnt

```
    def step(self, action): #assuming 1 step in 1 timestamp
        thermostat_setting, fan_speed = action

      # indoor_temp and humidity change with thermostat_setting and fan_speed
        if thermostat_setting-1<0:
          if fan_speed-1<0:
            self.indoor_temp -=0.5
          elif fan_speed-1 == 0:
            self.indoor_temp -=1
          else:
            self.indoor_temp -=2
        elif thermostat_setting-1>0:
          if fan_speed-1<0:
            self.indoor_temp +=0.5
          elif fan_speed-1 == 0:
            self.indoor_temp +=1
          else:
            self.indoor_temp ==2
        else:
          if fan_speed-1<0:
            self.humidity += 1
          elif fan_speed-1>0:
            self.humidity -= 1

        # one step happens in 1 minute and energy consumed in 1 minute = 50 wh
        self.time_stamp -=1
        self.energy_consumption += 500
```

Figure 3. 17 Defining the Step function of training class

```
        # reward
        reward = 0
        if self.indoor_temp>=22 and self.indoor_temp<=27:
          reward +=1
        else:
          reward -=1
        if self.humidity>=35 and self.humidity<=40:
          reward +=1
        else:
          reward -=1
        if self.energy_consumption<=3000:
          reward +=1
        else:
          reward -=1

        # termination
        done = False
        if self.time_stamp<=0:
          done = True
        elif self.energy_consumption >= 4000:
          done = True
        elif self.indoor_temp < 22 or self.indoor_temp > 27:
          done = True
        elif self.humidity < 30 or self.humidity > 60:
          done = True
        return np.array([self.indoor_temp, self.outdoor_temp, self.humidity,
                         self.energy_consumption, self.thermostat_setting, self.fan_speed]),
         reward, done, {}
```

Figure 3. 18 defining the reward function for every step

```
    def reset(self):
        self.indoor_temp =  25 + random.randint(-3,3)
        self.outdoor_temp = 14 + random.randint(-12,12)
        self.humidity = 40 + random.randint(-10,10)
        self.energy_consumption = 0
        self.thermostat_setting = 25 + random.randint(-3,3)
        self.fan_speed = 400 + random.randint(-200,200)
        self.time_stamp = 10
        return np.array([self.indoor_temp, self.outdoor_temp, self.humidity,
                         self.energy_consumption, self.thermostat_setting, self.fan_speed])
```

Figure 3. 19 Defining the reset function to reset the environment

```
[ ] env = HVACEnv()
```

Show hidden output

```
env.reset()
```

Show hidden output

Test Environment

```
[ ] episodes = 5
    for episode in range(1,episodes+1):
      obs = env.reset()
      done = False
      score = 0

      while not done:
        action = env.action_space.sample()
        obs, reward, done, info = env.step(action)
        score += reward
      print('Episode:{} Score:{}'.format(episode, score))
```

```
Episode:1 Score:-1
Episode:2 Score:20
Episode:3 Score:-1
Episode:4 Score:1
Episode:5 Score:14
```

Figure 3. 20 Testing the training environment

Train Model

```
log_path = os.path.join('Training','Logs')
model = PPO('MlpPolicy',env,verbose = 1, tensorboard_log = log_path)
```

Show hidden output

Figure 3. 21 Training the model on PPO policy and defining the log_path

```
model.learn(total_timesteps = 40000)
```

```
Logging to Training/Logs/PPO_3
---------------------------------
| rollout/           |          |
|    ep_len_mean     | 6.83     |
|    ep_rew_mean     | 6.87     |
| time/              |          |
|    fps             | 529      |
|    iterations      | 1        |
|    time_elapsed    | 3        |
|    total_timesteps | 2048     |
---------------------------------
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 7.01        |
|    ep_rew_mean          | 8.01        |
| time/                   |             |
|    fps                  | 499         |
|    iterations           | 2           |
|    time_elapsed         | 8           |
|    total_timesteps      | 4096        |
| train/                  |             |
|    approx_kl            | 0.004524854 |
|    clip_fraction        | 0.0289      |
|    clip_range           | 0.2         |
|    entropy_loss         | -1.04       |
|    explained_variance   | 0.512       |
|    learning_rate        | 0.0003      |
|    loss                 | 8.98        |
|    n_updates            | 410         |
|    policy_gradient_loss | -0.00503    |
|    value_loss           | 10.5        |
```

```
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 6.9         |
|    ep_rew_mean          | 7.72        |
| time/                   |             |
|    fps                  | 421         |
|    iterations           | 20          |
|    time_elapsed         | 97          |
|    total_timesteps      | 40960       |
| train/                  |             |
|    approx_kl            | 0.015257767 |
|    clip_fraction        | 0.0681      |
|    clip_range           | 0.2         |
|    entropy_loss         | -0.712      |
|    explained_variance   | 0.632       |
|    learning_rate        | 0.0003      |
|    loss                 | 5.24        |
|    n_updates            | 590         |
|    policy_gradient_loss | -0.00365    |
|    value_loss           | 7.67        |
-----------------------------------------
<stable_baselines3.ppo.ppo.PPO at 0x7c3fea315d20>
```

Figure 3. 22 learning the parameters

```
model_path = os.path.join('Training','Saved_Models','Model_PPO')
```

```
model.save(model_path)
```

```
del model
```

```
model = PPO.load(model_path,env)
```

```
Wrapping the env with a `Monitor` wrapper
Wrapping the env in a DummyVecEnv.
```

```
evaluate_policy(model,env,n_eval_episodes = 100, render = False)
```

```
(8.6, 5.124451190127583)
```

Figure 3. 23 Evaluating the policy

Evaluation

```
class HVACEvaluationEnv(Env):
    def __init__(self):
        # Define action and observation spaces
        self.action_space = MultiDiscrete([3, 3])
        self.observation_space = Box(low=np.array([-30, -30, 0, 0, 14, 200]),
                                     high=np.array([60, 60, 100, 5000, 32, 2500]))

    def step(self, action):
      thermostat_setting, fan_speed = action

      # Update indoor temperature and humidity based on actions
      if thermostat_setting-1<0:
        if fan_speed-1<0:
          self.indoor_temp -=0.5
        elif fan_speed-1 == 0:
          self.indoor_temp -=1
        else:
          self.indoor_temp -=2
      elif thermostat_setting-1>0:
        if fan_speed-1<0:
          self.indoor_temp +=0.5
        elif fan_speed-1 == 0:
          self.indoor_temp +=1
        else:
          self.indoor_temp ==2
      else:
        if fan_speed-1<0:
          self.humidity += 1
        elif fan_speed-1>0:
          self.humidity -= 1
```

Figure 3. 24 Defining the test environment

```
# Update energy consumption
  self.time_stamp -=1
  self.energy_consumption += 500


  # Calculate reward based on current state (simplified for evaluation)

  reward = 0
  if self.indoor_temp >= 22 and self.indoor_temp <= 27:
      reward += 1
  else:
      reward -= 1

  if self.humidity >= 35 and self.humidity <= 40:
      reward += 1
  else:
      reward -= 1

  if self.energy_consumption <= 3000:
      reward += 1
  else:
      reward -= 1

  # Check termination conditions
  done = False
  if self.time_stamp <= 0:
      done = True
  elif self.energy_consumption >= 4000:
      done = True
  elif self.indoor_temp < 22 or self.indoor_temp > 27:
      done = True
  elif self.humidity < 30 or self.humidity > 60:
      done = True
```

Figure 3. 25 Defining the similar reward function for testing environment

```python
  return np.array([self.indoor_temp, self.outdoor_temp, self.humidity,
                   self.energy_consumption, self.thermostat_setting, self.fan_speed]),

def reset(self):
    # Randomize initial conditions for each episode
    self.indoor_temp = 25 + random.randint(-3, 3)
    self.outdoor_temp = 14 + random.randint(-12, 12)
    self.humidity = 40 + random.randint(-10, 10)
    self.energy_consumption = 0
    self.thermostat_setting = 25 + random.randint(-3, 3)
    self.fan_speed = 400 + random.randint(-200, 200)
    self.time_stamp = 10

    return np.array([self.indoor_temp, self.outdoor_temp, self.humidity,
                     self.energy_consumption, self.thermostat_setting, self.fan_speed]
```

Figure 3. 26 Defining the reset function for test environment

```python
# Initialize evaluation environment
evaluation_env = HVACEvaluationEnv()

# Replace 'your_model_path' with the path to your trained model
trained_model = PPO.load(model_path)

# Run evaluation episodes
num_episodes = 500
energy_consumption_data = []
indoor_temperature_data = []
humidity_data = []

for _ in range(num_episodes):
    obs = evaluation_env.reset()
    episode_energy_consumption = []
    episode_indoor_temperature = []
    episode_humidity = []
    done = False
```

Figure 3. 27 Running evaluation episodes

```python
while not done:
    action, _ = trained_model.predict(obs, deterministic=True)
    obs, _, done, _ = evaluation_env.step(action)
    episode_energy_consumption.append(obs[3])  # Index 3 corresponds to
    episode_indoor_temperature.append(obs[0])  # Index 0 corresponds to
    episode_humidity.append(obs[2])  # Index 2 corresponds to humidity

# Check if the episode has sufficient data points
if len(episode_energy_consumption) >= 8:
    energy_consumption_data.append(episode_energy_consumption)
    indoor_temperature_data.append(episode_indoor_temperature)
    humidity_data.append(episode_humidity)
```

Figure 3. 28 evaluation episode continued

```python
# Aggregate data
avg_energy_consumption = np.mean(energy_consumption_data, axis=0)
avg_indoor_temperature = np.mean(indoor_temperature_data, axis=0)
avg_humidity = np.mean(humidity_data, axis=0)

# Visualize results
plt.figure(figsize=(10, 10))
plt.subplot(3, 1, 1)
# plt.plot(avg_energy_consumption marker: Literal['o'] ption')
plt.plot(avg_energy_consumption, marker='o', linestyle='-', label='Energy C
plt.xlabel('Time Steps')
plt.ylabel('Energy Consumption')
plt.title('Average Energy Consumption')
plt.axhspan(2500, 3500, color='gray', alpha=0.3, label='Target Range')
plt.legend()
```

Figure 3. 29 plotting the energy consumption v/s timestamp

```
# Visualize results
plt.figure(figsize=(10, 10))
plt.subplot(3, 1, 1)
# plt.plot(avg_energy_consumption marker: Literal['o'] ption')
plt.plot(avg_energy_consumption, marker='o', linestyle='-', label='Energy
plt.xlabel('Time Steps')
plt.ylabel('Energy Consumption')
plt.title('Average Energy Consumption')
plt.axhspan(2500, 3500, color='gray', alpha=0.3, label='Target Range')
plt.legend()

plt.subplot(3, 1, 2)
# plt.plot(avg_indoor_temperature, label='Indoor Temperature')
plt.plot(avg_indoor_temperature, marker='o', linestyle='-', label='Indoor
plt.xlabel('Time Steps')
plt.ylabel('Temperature')
plt.title('Average Indoor Temperature')
plt.axhspan(22, 27, color='gray', alpha=0.3, label='Target Range')
plt.legend()

plt.subplot(3, 1, 3)
plt.plot(avg_humidity, marker='o', linestyle='-', label='Humidity')
# plt.plot(avg_humidity, label='Humidity')
plt.xlabel('Time Steps')
plt.ylabel('Humidity')
plt.title('Average Humidity')
plt.axhspan(35, 45, color='gray', alpha=0.3, label='Target Range')
plt.legend()
```

Figure 3. 30 Plotting Humidity v/s time stamp and Plotting indoor temperature v/s time stamp

```python
# Calculate statistics
energy_mean = np.mean(avg_energy_consumption)
energy_std = np.std(avg_energy_consumption)
energy_percentiles = np.percentile(avg_energy_consumption, [25, 50, 75])

temperature_mean = np.mean(avg_indoor_temperature)
temperature_std = np.std(avg_indoor_temperature)
temperature_percentiles = np.percentile(avg_indoor_temperature, [25, 50, 75])
```

Figure 3. 31 Calculating Statistics

```python
humidity_mean = np.mean(avg_humidity)
humidity_std = np.std(avg_humidity)
humidity_percentiles = np.percentile(avg_humidity, [25, 50, 75])

# Print statistics
print("Energy Consumption:")
print(f"Mean: {energy_mean}, Standard Deviation: {energy_std}")
print(f"Percentiles (25th, 50th, 75th): {energy_percentiles}\n")

print("Indoor Temperature:")
print(f"Mean: {temperature_mean}, Standard Deviation: {temperature_std}")
print(f"Percentiles (25th, 50th, 75th): {temperature_percentiles}\n")

print("Humidity:")
print(f"Mean: {humidity_mean}, Standard Deviation: {humidity_std}")
print(f"Percentiles (25th, 50th, 75th): {humidity_percentiles}\n")
```

Figure 3. 32 printing statistics

```python
# Extract episode rewards from ep_info_buffer
episode_rewards = [ep_info['r'] for ep_info in model.ep_info_buffer]

# Plot learning curve
plt.plot(episode_rewards)
plt.xlabel('Episode')
plt.ylabel('Episode Reward')
plt.title('PPO Learning Curve')
plt.show()
```

Show hidden output

Figure 3. 33 Plotting PPO learning curve

Exploitation vs Exploration

```python
[131] # Extract entropy values from ep_info_buffer
entropy_values = [ep_info.get('entropy', 0.0) for ep_info in model.ep_info_buffer]

# Plot entropy values
plt.plot(entropy_values)
plt.xlabel('Episode')
plt.ylabel('Entropy')
plt.title('Policy Entropy')
plt.show()
```

Figure 3. 34 Exploitation v/s exploration curve

3.5 KEY CHALLENGES

Seamless communication and data exchange between hardware elements such as the DHT22 sensor and microcontrollers was one of the crucial aspects of this project. The inherent complexity of implementing Reinforcement Learning (RL) algorithms in the simulation environment was a big challenge.

A thorough grasp of machine learning principles was necessary for developing suitable reward functions, training RL agents, and comprehending and putting into practice RL concepts. To overcome this obstacle, the use of already-existing RL frameworks and libraries was done with knowledge through online resources and academic literature.

Optimising energy efficiency in HVAC systems using RL-based control strategies introduced further challenges due to the system dynamics and trade-offs between energy consumption and user comfort. Balancing these competing objectives while ensuring optimal system performance demanded careful consideration and iterative refinement of control policies. A multi-objective optimization approach was adopted, defining suitable reward functions and state-action spaces to achieve a balance between energy savings and user satisfaction.

Testing and validating the performance of the hardware prototype and RL simulation environment presented additional challenges. Comprehensive testing procedures, encompassing unit tests, integration tests, and performance evaluations were essential to ensure the system's reliability and responsiveness.

CHAPTER 4: TESTING

4.1 TESTING STRATEGY

1. Unit Testing:
Hardware Unit Testing: Each hardware component (DHT22 Sensor, Arduino Uno, XBee modules) is tested individually to ensure proper functionality and connectivity.
Software Unit Testing: Unit tests are conducted for each software module, including Arduino sketches for data processing and communication, Python scripts for interfacing with the RL simulation environment, and RL algorithms.

2. Integration Testing:
Hardware Integration: Hardware integration between Arduino boards, sensors, and Xbee modules have been done properly for Seamless communication and data exchange.

Wireless Communication Testing: Tested the reliability and range of wireless communication between XBee modules, ensuring stable transmission of sensor data.

Serial Communication Testing: Serial communication between Arduino Uno and the PLC, verifying the accuracy of data transmission and reception.

3. Functional Testing:
Sensor Data Acquisition: Thoroughly checked accuracy and reliability of temperature and humidity data acquired by the DHT22 sensor.

Wireless Data Transmission: Ensured the successful transmission of sensor data between Arduino boards via XBee modules.

RL Simulation Environment: Tested the RL simulation environment's ability to accurately model HVAC system dynamics and generate optimal control actions.

4. Performance Testing:

RL Algorithm Performance: Evaluate the performance of RL algorithms in optimising HVAC system control in the simulated environment.

Tools Used:

1. Arduino IDE: For developing and uploading Arduino sketches for data processing and control logic.
2. Python: For scripting and interfacing with the RL simulation environment, as well as implementing control algorithms.
3. OpenAI Gym: If using a RL simulation environment, OpenAI Gym can be utilised for developing and testing RL algorithms.
4. XBee Configuration Tools (XCTU): Tools provided by the XBee manufacturer for configuring and testing XBee modules' communication settings.

4.2 TEST CASES AND OUTCOMES

Different test cases were used to make sure that the data and communication flow is seamless.Each hardware component was tested individually to ensure proper functionality and connectivity.It has been verified that the sensor accurately measures temperature and humidity. Functionality tests were performed to ensure that the microcontroller can process data correctly and communicate with other components. Checked the communication capabilities of both slave and master XBee modules to ensure they could reliably transmit and receive data. The DHT22 sensor provided accurate readings, the Arduino Uno processed data correctly, and the XBee modules reliably transmitted data wirelessly.

Arduino Sketches were tested to ensure proper data processing and communication logic. Python Scripts were tested for their ability to interface with the RL simulation environment and handle data accurately. The tests were conducted to ensure that the RL algorithms could operate within the simulation environment without errors.
The software modules demonstrated correct functionality in data processing, interfacing, and algorithm execution.

Evaluated the performance of RL algorithms in optimising HVAC system control within the simulated environment. The RL algorithms significantly improved HVAC system efficiency and performance. The system adapted to changing conditions effectively, optimising energy usage and maintaining comfort levels.

CHAPTER 5: RESULTS AND EVALUATION

5.1 RESULTS

The project focused on two distinct components: a hardware prototype and a Reinforcement Learning (RL) simulation environment.The hardware prototype was made up of integrated sensors, microcontrollers, and communication modules that were used to gather and wirelessly transmit real-time environmental data. In particular, the prototype made use of XBee modules for wireless communication, an Arduino Uno microcontroller for data processing, and a DHT22 sensor for temperature and humidity data collection. The prototype effectively illustrated the viability of gathering environmental data and sending it to the RL simulation environment for additional analysis through hardware integration and testing.

The RL simulation environment was developed to simulate the decision-making process of the CPS framework. Utilising RL libraries like TensorFlow and PyTorch and the Python programming language, the simulation environment made it possible to implement and test RL algorithms for optimising HVAC system control. The RL simulation environment was an essential tool for testing various control strategies and algorithms in a controlled environment.

The findings demonstrated how flexible and responsive the CPS framework is when it comes to dynamically modifying control parameters in response to shifting environmental circumstances. The framework demonstrated the capacity to optimise HVAC system operation in real-time by utilising RL algorithms, which effectively balanced energy consumption and user comfort requirements.Analysis examined the scalability and robustness of the developed CPS framework in addition to performance evaluation. It was noted that the framework showed encouraging potential for scalability, as it could support more sensors and actuators for more thorough system optimization and monitoring.

The interpretation of the findings emphasises how important each part is to achieving the project's goals. Although the hardware prototype demonstrated how data acquisition and transmission mechanisms could be implemented practically, the RL simulation environment offered a way to test and improve control strategies without requiring their physical deployment. The project created the foundation for the eventual integration of software and hardware components into a single, self-adaptive CPS framework by developing these components concurrently.

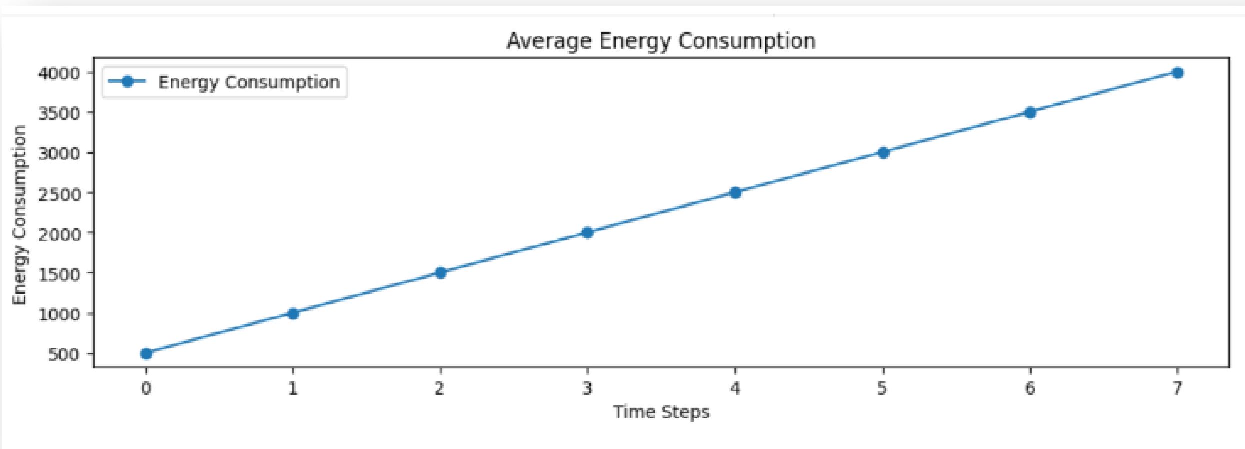

Figure 5. 1 Average Energy Consumption

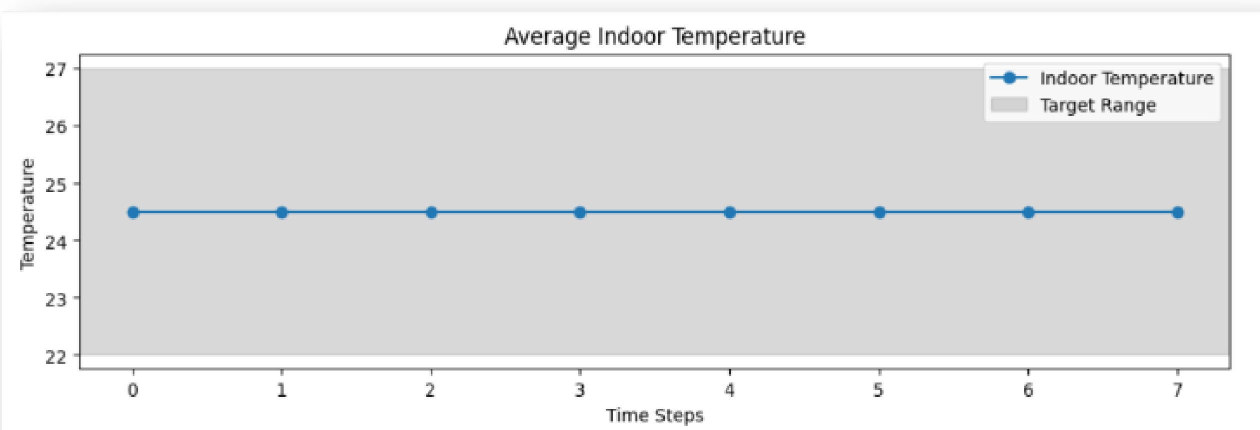

Figure 5. 2 Average Indoor Temperature

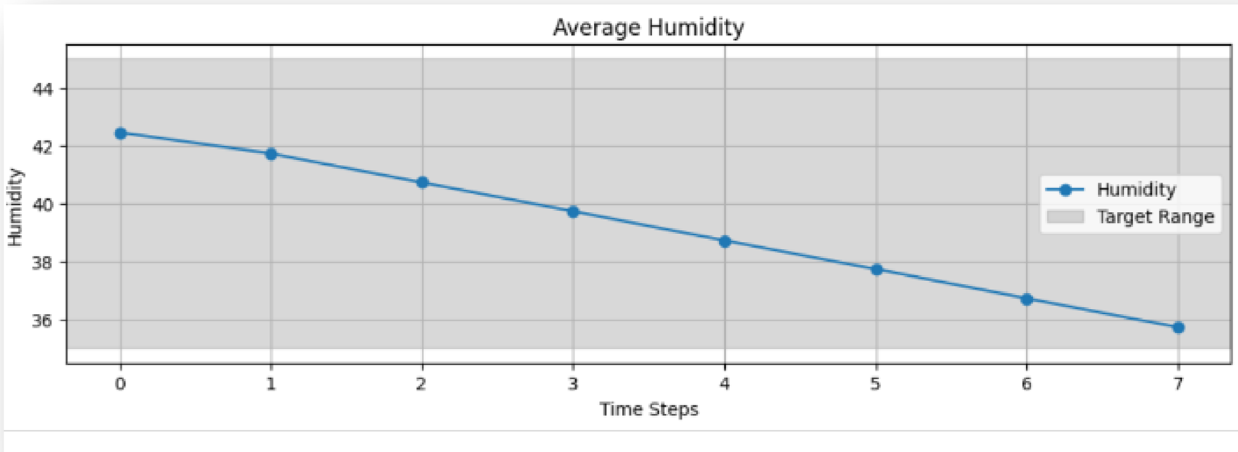

Figure 5. 3 Average Humidity

The average energy consumption is increasing in a continuous manner, as supposed to be in a real-world system. The average indoor temperature remains almost constant, maintaining good comfort level, indicating good optimization of the parameter. The graph of humidity versus time steps showing a declining straight line indicates a decrease in humidity over time. This implies that the environment is becoming drier or that moisture is being removed from the air gradually, suggesting the operation of natural drying of the air due to heating.

The continuous trends in the graphs above show that the agent has learnt to handle all the three parameters in an optimal manner and has learnt a fair trade-off between them during the training phase and is good at applying the same when exposed to unforeseen situations.

```
Energy Consumption:
Mean: 2250.0, Standard Deviation: 1145.64392373896
Percentiles (25th, 50th, 75th): [1375. 2250. 3125.]

Indoor Temperature:
Mean: 24.486238532110093, Standard Deviation: 0.0
Percentiles (25th, 50th, 75th): [24.48623853 24.48623853 24.48623853]

Humidity:
Mean: 39.19304281345566, Standard Deviation: 2.23941262360362
Percentiles (25th, 50th, 75th): [37.47782875 39.22782875 40.97782875]
```

Figure 5. 4 Statistical results

The statistical analysis gives more inference on the performance of the RL agent, describing the mean and variance of all the three parameters namely, Energy consumption, Indoor Temperature and Humidity. The standard deviation value of Indoor Temperature is 0.0 highlighting the model's capability to exactly determine the environment conditions and adapt to it.

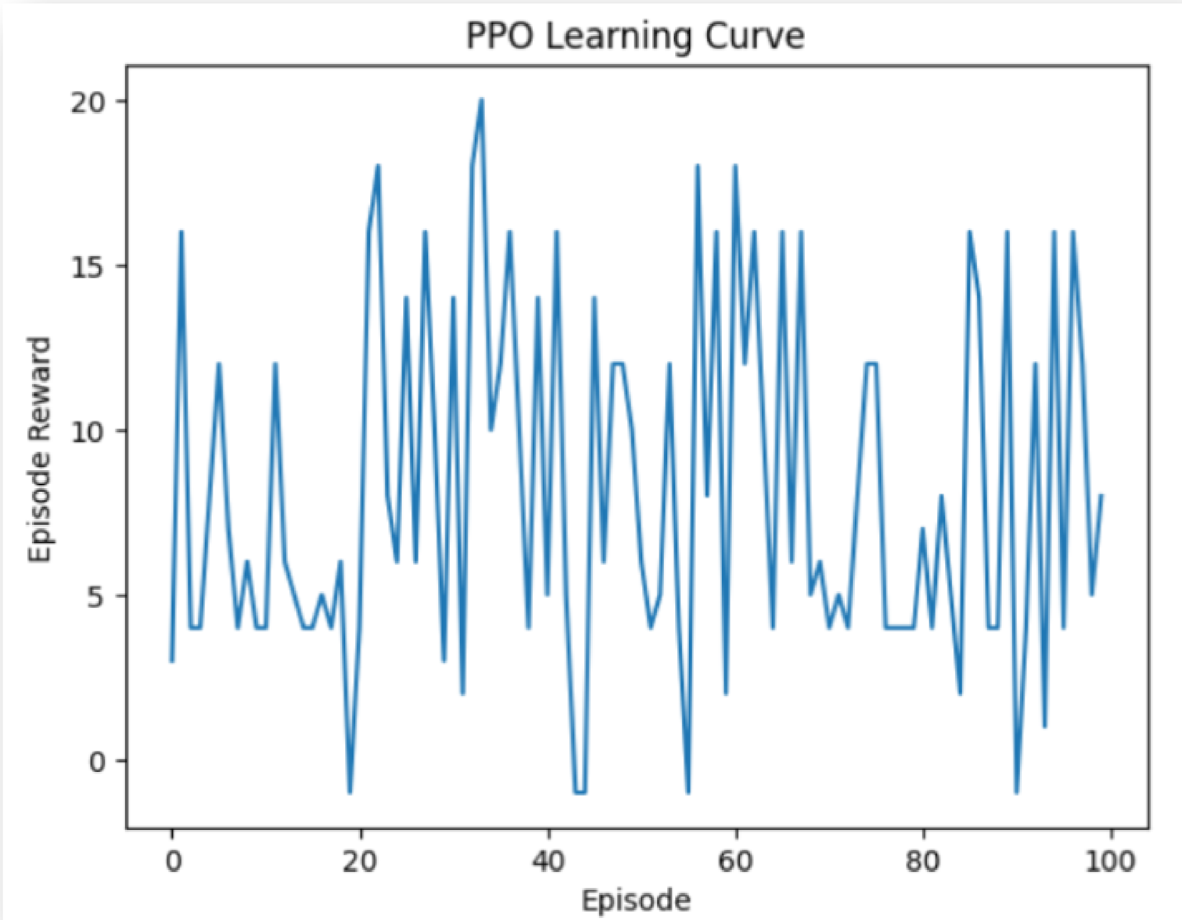

Figure 5. 5 PPO learning curve

The PPO learning Curve graphically shows the learning process of the RL agent in the HVACEnv environment. The agent adjusts its actions to maximise the rewards per episode to define, update and redefine the policies it leans.

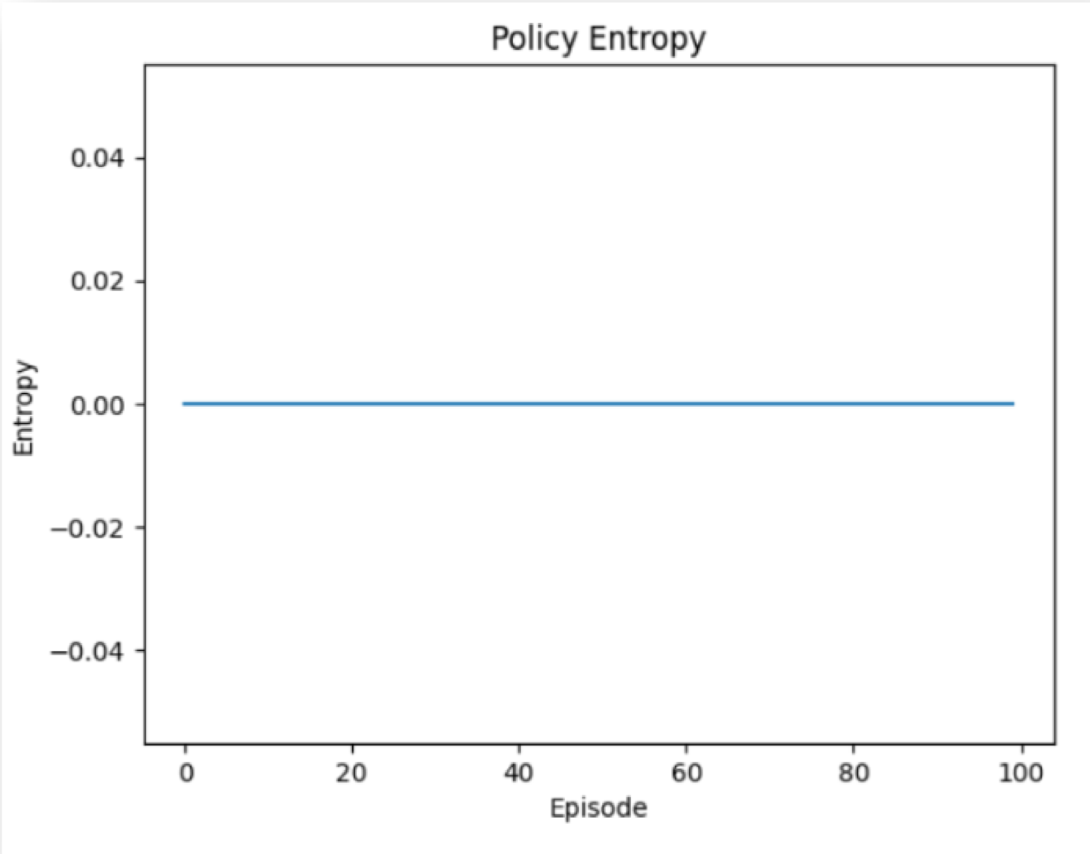

Figure 5. 6 Policy entropy curve

5.2 COMPARISON WITH EXISTING SOLUTIONS

Existing Solutions typically consist of standard sensors, thermostats, and basic controllers with limited processing capabilities.The hardware prototype for the RL-based system includes advanced sensors, microcontrollers capable of handling complex data processing, communication modules for seamless data exchange, and a simulation environment for RL decision-making. They may rely on wired communication or basic wireless protocols with potential latency issues.Whereas this project utilises advanced wireless communication modules that ensure low latency and high reliability in data transmission between sensors, controllers, and the RL environment.

In contrast to prevalent deterministic approaches in Cyber-Physical Systems (CPS) modeling, the proposed project advocates for a shift towards stochastic modeling techniques and reinforcement learning (RL) algorithms, notably utilizing the Proximal Policy Optimization (PPO) algorithm. While current methodologies, such as frequency domain models, Artificial Neural Networks (ANN), Support Vector Machines (SVM), AutoRegressive with exogenous input (ARX), and AutoRegressive Integrated Moving Average (ARIMA), rely on data-driven approaches to approximate system behavior, they often overlook the inherent uncertainties and dynamic characteristics of industrial environments. This limitation leads to suboptimal performance and energy inefficiencies, hindering adaptability to changing circumstances. By integrating stochastic aspects into CPS models and leveraging RL algorithms within a simulation environment, the project aims to enhance decision-making and control strategies. The RL simulation environment facilitates iterative learning and adaptation, enabling the refinement of control policies for HVAC management in response to evolving conditions and user requirements. Through the implementation of the PPO algorithm, the project seeks to develop stochastic policies that optimise HVAC system performance, thereby addressing the challenges faced by autonomous industries and advancing the efficiency and adaptability of CPS in industrial settings.

By dynamically modifying control parameters in response to shifting environmental factors and user preferences, the framework exhibits exceptional adaptability. The framework continuously learns and adapts based on real-time feedback, ensuring optimal performance under varying operating conditions, in contrast to rule-based approaches that depend on fixed control policies.

CHAPTER 6: CONCLUSION AND FUTURE SCOPE

6.1 CONCLUSION

It is shown via test results that the incorporation of Reinforcement Learning (RL) techniques with hardware components allows for autonomous learning and adaptation of control strategies, leading to notable gains in user comfort and energy efficiency over conventional rule-based approaches. The investigation also demonstrated the framework's resilience, scalability, and adaptability, underscoring its potential for practical uses in industrial contexts.

Despite its promising performance, the self-adaptive CPS framework exhibits certain limitations that warrant consideration for betterment. One drawback is the difficulty and resources needed to integrate hardware and implement RL algorithms, which could make it difficult for them to be widely adopted and used. Additionally, more investigation and improvement may be necessary to address particular domain requirements and constraints before the framework can be applied to a variety of industrial environments and HVAC systems, as new models are required for new environments everytime.

The project makes several noteworthy contributions to the field of CPS technology and HVAC system optimization. Firstly, It has been shown that it is both feasible and effective to combine RL techniques with hardware elements to build CPS frameworks that are self-adaptive, opening the door to more adaptive and efficient industrial processes. Important insights are offered for future CPS technology research and development, guiding the design of more reliable and scalable solutions, by exposing the shortcomings and performance of the framework. Finally, the project promotes innovation and interdisciplinary collaboration in the field by improving discussion on energy efficiency, sustainability, and autonomy in industrial systems.

6.2 FUTURE SCOPE

There are further things that can be added in this project. The integration of the hardware prototype with the RL simulation environment to create a unified CPS framework will be the main focus in future. This integration will enable real-time interaction between the physical environment and the decision-making algorithms which will lead to autonomous control and optimization of HVAC systems.

Further research and experimentation will be conducted to refine and optimise the RL algorithms used in the simulation environment. The RL model will be properly trained and optimised for a large number of parameters. This will increase the learning capabilities and decision-making accuracy.

Training and testing will be done with respect to the real world industrial settings making it more reliable and efficient.

Additional sensors and actuators will be integrated to capture more comprehensive environmental data and enable better control of HVAC systems, thus, making it more optimised. This includes exploring the use of IoT devices and new learning technologies to enhance data collection and processing capabilities.

REFERENCES

[1] Somani, Nisha Ashok, and Yask Patel. "Zigbee: A low power wireless technology for industrial applications." International Journal of Control Theory and Computer Modelling (IJCTCM) 2.3, 27-33, 2012.

[2] Wen-Tsai Sung, Yao-Chi Hsu, "Designing an industrial real-time measurement and monitoring system based on embedded system and ZigBee" in Expert Systems with Applications, Volume 38, Issue 4, 2011

[3] Q. Wang and J. Jiang, "Comparative Examination on Architecture and Protocol of Industrial Wireless Sensor Network Standards," in IEEE Communications Surveys & Tutorials, vol. 18, no. 3, pp. 2197-2219, third quarter 2016.

[4] Tyrovolas, M., & Hajnal, T. "Inter-communication between Programmable Logic Controllers using IoT technologies: A Modbus RTU/MQTT Approach." ArXiv. /abs/2102.05988, 2021.

[5] Bakshi, Somesh, et al. "Monitoring and control of PLC based automation system parameters using IoT." Int. Res. J. Eng. Technol 6.03, 650-652, 2019.

[6] V. C. Gungor and G. P. Hancke, "Industrial wireless sensor networks: Challenges design principles and technical approaches", IEEE Trans. Ind. Electron., vol. 56, no. 10, pp. 4258-4265, Oct. 2009.

[7] Wagner, Raymond S. "Standards-based wireless sensor networking protocols for spaceflight applications." 2010 IEEE Aerospace Conference. IEEE, 2010.

[8] Gómez, María José, Cristina Castejón, and Juan Carlos García-Prada. "Automatic condition monitoring system for crack detection in rotating machinery." Reliability Engineering & System Safety 152 (2016): 239-247.

[9] Afroz, Zakia, et al. "Modeling techniques used in building HVAC control systems: A review." Renewable and sustainable energy reviews 83, 64-84, 2018.

[10] Yao, Ye, et al. "Modular modeling of air-conditioning system with state-space method and graph theory." International Journal of Refrigeration 99, 9-23, 2019

[11] Amir, M., Poncela, J., Uqaili, M.A. et al. "Optimal Design of Remote Terminal Unit (RTU) for Wireless SCADA System for Energy Management." Wireless Pers Commun 69, 999–1012, 2013.

[12] Wang, Jinjiang, et al. "Deep learning for smart manufacturing: Methods and applications." Journal of manufacturing systems 48, 144-156, 2018.

[13] Q. Xin, G. Wu, W. Fang, J. Cao and Y. Ping, "Opportunities for Reinforcement Learning in Industrial Automation," 2021 7th International Conference on Big Data and Information Analytics (BigDIA), Chongqing, China, 2021.

[14] Li, Chengxi, et al. "Deep reinforcement learning in smart manufacturing: A review and prospects." CIRP Journal of Manufacturing Science and Technology 40, 75-101, 2023.

[15] Mnih, V., Kavukcuoglu, K., Silver, D. et al. "Human-level control through deep reinforcement learning", Nature 518, 529–533, 2015.

[16] Schulman, John, et al. "Proximal policy optimization algorithms." arXiv preprint arXiv:1707.06347, 2017.

[17] Mnih, Volodymyr, et al. "Playing atari with deep reinforcement learning." arXiv preprint arXiv:1312.5602 ,2013.

[18] .Afram, Abdul, and Farrokh Janabi-Sharifi. "Review of modeling methods for HVAC systems." Applied thermal engineering 67.1-2, 507-519, 2014.

[19] C. Liu, J. Ding and J. Sun, "Reinforcement Learning Based Decision Making of Operational Indices in Process Industry Under Changing Environment," in IEEE Transactions on Industrial Informatics, vol. 17, no. 4, pp. 2727-2736, April 2021.

[20] Zhou, Pengwei, et al. "Safe reinforcement learning method integrating process knowledge for real-time scheduling of gas supply network." Information Sciences 633, 280-304, 2023.

[21] Jordan, Scott, et al. "Evaluating the performance of reinforcement learning algorithms." International Conference on Machine Learning. PMLR, 2020.

[22] Chan, Stephanie CY, et al. "Measuring the reliability of reinforcement learning algorithms." arXiv preprint arXiv:1912.05663, 2019.

[23] Gu, Shangding, et al. "A review of safe reinforcement learning: Methods, theory and applications." *arXiv preprint arXiv:2205.10330*, 2022.

[24] Shakya, Ashish Kumar, Gopinatha Pillai, and Sohom Chakrabarty. "Reinforcement learning algorithms: A brief survey." Expert Systems with Applications, 2023.

[25] Dulac-Arnold, Gabriel, Daniel Mankowitz, and Todd Hester. "Challenges of real-world reinforcement learning." arXiv preprint arXiv:1904.12901, 2019.

paper

ORIGINALITY REPORT

7%	5%	8%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Abdul Afram, Farrokh Janabi-Sharifi. "Review of modeling methods for HVAC systems", Applied Thermal Engineering, 2014 Publication	1%
2	Chao Liu, Jinliang Ding, Jiyuan Sun. "Reinforcement Learning based Decision-making of Operational Indices in Process Industry under Changing Environment", IEEE Transactions on Industrial Informatics, 2020 Publication	<1%
3	Gómez, M.J., C. Castejón, and J.C. García-Prada. "Automatic condition monitoring system for crack detection in rotating machinery", Reliability Engineering & System Safety, 2016. Publication	<1%
4	iifiir.org Internet Source	<1%
5	researchrepository.murdoch.edu.au Internet Source	<1%
	ds.inflibnet.ac.in	

6	Internet Source	<1 %
7	www.researchgate.net Internet Source	<1 %
8	www.science.gov Internet Source	<1 %
9	xml.jips-k.org Internet Source	<1 %
10	Vehbi Gungor. "Industrial Wireless Sensor Networks: Challenges, Design Principles, and Technical Approaches", IEEE Transactions on Industrial Electronics, 2009 Publication	<1 %
11	Submitted to Western Illinois University Student Paper	<1 %
12	www.semanticscholar.org Internet Source	<1 %
13	Submitted to University College London Student Paper	<1 %
14	papasearch.net Internet Source	<1 %
15	www.coursehero.com Internet Source	<1 %
16	www.mromagazine.com Internet Source	<1 %

17	Jinjiang Wang, Yulin Ma, Laibin Zhang, Robert X. Gao, Dazhong Wu. "Deep learning for smart manufacturing: Methods and applications", Journal of Manufacturing Systems, 2018 Publication	<1 %
18	link.springer.com Internet Source	<1 %
19	Submitted to Charles Sturt University Student Paper	<1 %
20	Submitted to Coventry University Student Paper	<1 %
21	digitalcommons.wcupa.edu Internet Source	<1 %
22	e-archivo.uc3m.es Internet Source	<1 %
23	Hongjia Li, Ruizhe Cai, Ning Liu, Xue Lin, Yanzhi Wang. "Deep reinforcement learning: Algorithm, applications, and ultra-low-power implementation", Nano Communication Networks, 2018 Publication	<1 %
24	Pengwei Zhou, Zuhua Xu, Xiaoping Zhu, Jun Zhao, Chunyue Song, Zhijiang Shao. "Safe reinforcement learning method integrating process knowledge for real-time scheduling	<1 %

of gas supply network", Information Sciences, 2023
Publication

25	Submitted to University of Liverpool Student Paper	<1 %
26	Yousef M. K. Ali, Omar A. Zargelin, Fadel Lashhab, Abdulbasit Alaribi. "Water Level Control System using Programmable Logic Controller (PLC): Rujban Water Supply System", 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), 2021 Publication	<1 %

Exclude quotes	On	Exclude matches	< 14 words
Exclude bibliography	On		

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: ……………………….

Type of Document (Tick): **PhD Thesis** **M.Tech Dissertation/ Report** **B.Tech Project Report** **Paper**

Name: ___________________________ __**Department:** _________________ **Enrolment No** _________

Contact No. ______________________________**E-mail.** ______________________________________

Name of the Supervisor: ____

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): ________________________

____

____

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

– Total No. of Pages =
– Total No. of Preliminary pages =
– Total No. of pages accommodate bibliography/references =

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at ……………….(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor) **Signature of HOD**

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	**Excluded**	**Similarity Index (%)**	**Generated Plagiarism Report Details** (Title, Abstract & Chapters)	
	• All Preliminary Pages • Bibliography/Images/Quotes • 14 Words String		Word Counts	
Report Generated on			Character Counts	
		Submission ID	Total Pages Scanned	
			File Size	

Checked by
Name & Signature **Librarian**

………

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com