INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

MESTRADO EM ENGENHARIA INFORMÁTICA

Harnessing Digital Twin Technology for Smart Grid Optimisation in Renewable Energy Communities

RUI MIGUEL RIO DE PINA
Junho de 2025

POLITÉCNICO
DO PORTO

ISEP INSTITUTO SUPERIOR
DE ENGENHARIA DO PORTO

Harnessing Digital Twin Technology for Smart Grid Optimisation in Renewable Energy Communities

Rui Miguel Rio de Pina

Dissertation to obtain the Master's Degree in
Informatics Engineering, Specialization Area in
Software Engineering

Supervisor: Doctor Luís Lino Ferreira, Polytechnic of Porto – School of Engineering
Co-supervisor: Tiago Fonseca, Polytechnic of Porto – School of Engineering

Porto, June 2025

Statement of Integrity

I hereby declare having that I have conducted this academic work with integrity.

I have not plagiarised or applied any form of undue use of information or falsification of results along the process leading to its elaboration.

Therefore, the work presented in this document is original and authored by me, having not previously been used for any other end. The exceptions are explicitly recognised in the section “Ethical considerations” of the first chapter. This section also states how AI tools were used and for what purpose.

I further declare that I have fully acknowledged the Code of Ethical Conduct of P.PORTO.

ISEP, Porto, June 29, 2025

Rui Miguel Rio de Pina

Resumo

A transição global para sistemas energéticos sustentáveis exige quadros metodológicos avançados capazes de gerir a complexa dinâmica estocástica inerente às Comunidades de Energia Renovável (CER). Os atuais paradigmas de simulação mostram-se insuficientes para capturar as interações multi-escalar entre recursos energéticos distribuídos, comportamento dos prossumidores e restrições da rede. Esta dissertação apresenta avanços significativos para a plataforma de simulação CityLearn, uma plataforma de código aberto para aprendizagem por reforço multiagente na gestão energética de edifícios, através de cinco inovações metodológicas fundamentais que, em conjunto, elevam a sua fidelidade em direção aos padrões de *Digital Twin*: (1) adaptação da granularidade temporal permitindo comutação dinâmica de resolução em intervalos infra horários, (2) integração de novos modelos físicos incluindo eletrodomésticos como máquinas de lavar, (3) otimização multiobjetivo que reconcilia metas de sustentabilidade e económicas concorrentes, (4) realismo estocástico através de modelação probabilística de incertezas na geração e procura, e (5) serialização padronizada de dados para interoperabilidade.

Uma revisão sistemática baseada na metodologia PRISMA identifica primeiro as limitações críticas das abordagens de modelação de CER existentes, particularmente no que diz respeito à sua representação de incertezas do mundo real e fenómenos emergentes à escala comunitária. O desenvolvimento subsequente segue uma metodologia de engenharia formal compreendendo: (1) síntese de requisitos a partir de estudos de caso operacionais de CER, (2) melhoria arquitetónica da estrutura de simulação, e (3) validação abrangente contra conjuntos de dados sintéticos e empíricos.

A validação experimental num caso de estudo com 17 edifícios demonstra as capacidades da estrutura através das novas funcionalidades desenvolvidas: redução de 73,5% nas emissões de carbono através de estratégias de controlo otimizadas para energia solar, redução de custos em 71% em edifícios com objetivos de otimização diferenciados, e manutenção dos parâmetros de conforto térmico dentro de limiares de ±0,5°C, tudo isto enquanto se alcança uma melhoria de 18,6% nas métricas de estabilidade da rede. Estes resultados fornecem evidência quantitativa para a otimização simultânea de objetivos de sustentabilidade, económicos e de fiabilidade em operações de CER.

As contribuições metodológicas proporcionam valor imediato para operadores de comunidades energéticas, enquanto estabelecem bases essenciais para o desenvolvimento futuro de *Digital Twins*. Para os decisores políticos e económicos, os resultados empíricos oferecem *benchmarks* verificáveis para o desenho de CER, particularmente no equilíbrio entre metas ambientais e restrições operacionais. Ao avançar as capacidades de simulação em direção aos padrões de *Digital Twin*, este trabalho fornece um elo crucial em falta entre os modelos teóricos de transição energética e os requisitos de implementação no mundo real.

Palavras-chave: *Digital Twin*, Comunidades de Energia Renovável, Redes Energéticas Inteligentes, Simulação de Energia, Granularidade Temporal, Otimização Energética

Abstract

The global transition toward sustainable energy systems necessitates advanced methodological frameworks capable of managing the complex, stochastic dynamics inherent in Renewable Energy Communities (RECs). Current simulation paradigms remain insufficient for capturing the multi-scale interactions between distributed energy resources, prosumer behaviour, and grid constraints. This dissertation presents significant advances to the CityLearn simulation platform, an open-source platform for Multi-Agent Reinforcement Learning (MARL) in building energy management, through five key methodological innovations that collectively advance its fidelity toward Digital Twin standards: (1) temporal granularity adaptation enabling dynamic resolution switching across sub-hourly intervals, (2) expanded asset integration incorporating household appliances including washing machines, (3) multi-objective optimisation reconciling competing sustainability and economic goals, (4) stochastic realism through probabilistic modelling of generation and demand uncertainties, and (5) standardised data export serialization for interoperability.

A systematic PRISMA-based review first establishes critical limitations in existing REC modelling approaches, particularly regarding their representation of real-world uncertainties and emergent community-scale phenomena. Subsequent development follows a formal engineering methodology comprising: (1) requirements synthesis from operational REC case studies, (2) architectural enhancement of the simulation framework, and (3) comprehensive validation against both synthetic and empirical datasets.

Experimental validation across a 17-building REC Case Study demonstrates framework's capabilities, through the newly developed functionalities: 73.5% reduction in carbon emissions through solar-optimised control strategies, 71% cost reduction in buildings with differentiated optimisation objectives, and maintenance of thermal comfort parameters within ±0.5°C thresholds, all while achieving 18.6% improvement in grid stability metrics. These results provide quantitative evidence for the simultaneous optimisation of sustainability, economic, and reliability objectives in REC operations.

The methodological contributions provide immediate value for energy community operators while establishing essential foundations for future Digital Twin development. For policymakers, the empirical results offer verifiable benchmarks for REC design, particularly in balancing environmental targets with operational constraints. By advancing simulation capabilities toward Digital Twin standards, this work provides a crucial missing link between theoretical energy transition models and real-world deployment requirements.

Keywords: Digital Twin, Renewable Energy Communities, Smart Grids, Energy Simulation, Temporal Granularity, Energy Optimisation

Acknowledgements

To my beloved father, who instilled in me a passion for learning and always encouraged me to pursue my dreams. Your unwavering support and guidance have been invaluable in my life. This dissertation is dedicated to you, with deepest gratitude and love.

I would also like to extend my heartfelt appreciation to my sister, nephews, and brother-in-law. Your constant encouragement and belief in my abilities have been a source of strength and inspiration.

Additionally, I am grateful for the guidance and support of my supervisors, Doctor Luís Lino Ferreira and Tiago Fonseca. Their profound insights and encouragement have played an integral role in shaping both my academic and personal growth.

I would like to express my sincere gratitude to SoftCPS for providing me with the opportunity to work on this project. The trust and support they have shown throughout this journey have been instrumental.

I am grateful for the invaluable experience and knowledge I have gained through our collaboration.

To my family, supervisors, and loved ones, your presence in my life has made all the difference. Though I may not always express it, your contributions and support have played a significant role in my achievements. Thank you for being there for me and believing in me. This report is dedicated to all of you, with deep appreciation and love.

Table of Contents

List of Figures

List of Tables

Acronyms

List of Acronyms
ACM Association for Computing Machinery
AI Artificial Intelligence
API Application Programming Interface
CD Continuous Deployment
CI Continuous Integration
CI/CD Continuous Integration / Continuous Deployment
CLI Command Line Interface
CPS Cyber-physical System
CN Connection
COP Coefficient of Power
CSV Comma Separated Values
DD Digital Twin Data
DER Distributed Energy Resource
DT Digital Twin
EMS Energy Management System
ES Energy Storage
ESS Energy Storage Systems
EU European Union
EV Electric Vehicle
FURPS+ Functionality, Usability, Reliability, Performance, Supportability and Others
GAN Generative Adversarial Network
GeLaP German Labelled Dataset for Power Consumption
HEMS Home Energy Management System
HVAC Heating, ventilation, and air conditioning
IEEE Institute of Electrical and Electronics Engineers
IoT Internet of Things
IPP Polytechnic Institute of Porto
ISEP School of Engineering of the Polytechnic of Porto
ISO International Organization for Standardization
IT Information Technology
JSON JavaScript Object Notation
KPI Key Performance Indicator
kWh Kilowatt Hour
LSTM Long Short-Term Memory
MARL Multi-Agent Reinforcement Learning
MAS Multi-Agent Systems
MLP Multilayer Perceptron
MO Main Objective
MPTT Maximum Power Point Tracking
MQTT Message Queuing Telemetry Transport
OPEVA Optimisation of Electric Vehicle Autonomy
P.Porto Polytechnic of Porto
PE Physical Entity
PLM Product Lifecycle Management

PRISMA Preferred Reporting Items for Systematic Reviews and Meta-Analyses
PV Photovoltaic
REC Renewable Energy Community
RES Renewable Energy Sources
RL Reinforcement Learning
RQ Research Question
SAC Soft Actor-Critic
SC Societal
SCADA Supervisory Control and Data Acquisition
Sct Scientific
SDG Sustainable Development Goals
SO Sub Objective
SOC State of Charge
SOFTCPS Software for Cyber Physical Systems Research Group
SS Service System
UI User Interface
UUID Universally Unique Identifier
W Watt
WBS Work Breakdown Structure
XML Extensible Markup Language
YAML YAML Ain't Markup Language
ZNE Zero Net Energy

List of Listings

1. Introduction

This chapter gives a comprehensive overview of the whole dissertation, which will set the proper groundwork to understand the context, challenges, and motives of this research. It moves smoothly from broad background information through increasingly focused discussion of the problem at hand.

1.1. Contextualisation

This work was carried out in the Software Technologies for Cyber-Physical Systems (SoftCPS) group at the School of Engineering, Polytechnic of Porto (ISEP), part of the HumanISE research centre within INESC TEC. SoftCPS focuses on software for cyber-physical systems, particularly in middleware, IoT, edge computing, and parallel/distributed processing. The research was conducted in collaboration with academic and industry partners under the European Union (EU)-funded OPEVA project [1], aimed at improving Electric Vehicle (EV) autonomy.

1.1.1. The Revolution in Energy Systems

Transitioning to renewable energy sources has become critical considering the growing urgency to tackle climate change and achieve sustainable development. This transformation, that is the transition of energy sources, is one that will reshape the sector with complex challenges related to managing decentralised and variable energy flows [2]. Renewable Energy Communities (RECs) have emerged as a locally confined solution that empowers prosumers to collectively manage energy production and consumption in collaboration, such as those from single households with solar panels. This would involve balancing local supply and demand of energy, optimisation of the usage of resources, and grid stability, while ensuring energy independence and sustainability [3].

Estimates suggest that RECs could represent 17% of installed wind capacity and 21% of solar energy capacity by 2050 in Europe [4]. By 2050, almost half of the households in the EU will produce renewable energy; thus, RECs also play a crucial role in achieving sustainable energy goals [5].

This shift in paradigm has been accompanied by technological changes, such as wide penetration of EVs, heat pumps, and distributed energy storage [6]. These innovations, while promising, introduce additional layers of complexity to the energy ecosystem. Effectively managing this complexity requires advanced tools and systems that can adapt to dynamic conditions, address variability, and coordinate among multiple stakeholders [3].

1.1.2. The Need for Simulation

The operation of energy systems, particularly in RECs, requires careful balancing of complex dynamics involving generation, storage, and consumption [7]. Deploying untested Energy Management System (EMS) algorithms directly in real-world settings poses significant risks, including jeopardizing grid stability and compromising user comfort [7]. For example, an improperly tested EMS could fail to charge an EV in time for its scheduled departure or mismanage demand-side resources, leading to inefficiencies and dissatisfaction among prosumers.

Simulation platforms address these challenges by providing a controlled and risk-free environment for developing and testing EMS strategies. By mimicking real-world conditions, simulators enable detailed scenario analysis, including rare or extreme events, ensuring that EMS algorithms are reliable and dependable before deployment [8]. This is particularly valuable for methodologies like Reinforcement Learning (RL), which rely on iterative trial-and-error processes to optimise decision-making [9]. In such cases, simulation is critical for training algorithms, as direct interaction with the grid for learning purposes could lead to unintended consequences.

For RECs, the role of simulation goes beyond algorithm testing. A well-designed simulation environment serves as a benchmark for comparing different EMS approaches, fostering collaboration and standardization within the research community. It also enables the identification of the most effective solutions under consistent and replicable conditions, accelerating innovation and guiding the development of practical tools for energy management.

When designed to closely replicate real-world energy systems, simulators effectively function as DTs, providing a virtual counterpart of the REC’s physical assets [10]. This ensures that insights gained in the simulation environment can be transferred to real-world applications. By enabling the safe exploration of advanced energy strategies, simulators are essential for bridging the gap between theoretical research and practical implementation, helping to realise the full potential of RECs and other decentralised energy systems [11].

1.1.3. Identified Problems

Despite advances in EMSs and simulation tools, a standardised framework for accurately modelling the complexity of RECs remains lacking. Current tools often fail to deliver comprehensive DTs, high-fidelity virtual models that simulate interactions among diverse assets like solar panels, heat pumps, storage systems, and decentralised sources.

This shortfall presents several challenges. Tools such as Energym [12] and BOPTEST [13] which engines such as EnergyPlus [14] and Modelica [15] provide detailed simulations at the building level but lack scalability to model community-wide energy dynamics. Similarly, platforms like SAM [16] and GridSim [17] offer insights into solar and storage performance but cannot support dynamic interactions or holistic REC optimisation.

The complexity of REC systems, comprising interconnected components requiring precise coordination, demands scalable architectures and advanced algorithms [18]. Simulating these intricate interdependencies requires advanced algorithms and scalable architectures, which are often lacking in current tools [18]. Existing tools struggle to manage the heterogeneity of assets, from small appliances to large generators, each producing varied and difficult-to-integrate data [19], [20]. This lack of adaptability and consistency limits the effectiveness of these tools in designing and testing innovative energy management strategies.

Another critical limitation of existing tools is their inability to provide a unified testing and benchmarking environment. Research efforts are often siloed, with studies focusing on the optimisation of individual assets or subsystems rather than the collective behaviour of energy communities [21]. Without a standardised framework to simulate collective flexibility, collaboration and innovation remain constrained.

DTs offer the potential to overcome these limitations by creating virtual environments that mirror real-world REC systems in both complexity and detail. However, existing simulation tools rarely meet the requirements needed for effective DT implementation, such as scalability, adaptability, and interoperability with various energy management technologies [22]. Additionally, the computational demands and expertise required to build and maintain accurate DTs present significant barriers to adoption, further widening the gap between research needs and available tools.

Addressing these challenges requires the development of simulation frameworks that not only replicate the behaviour of individual components but also capture their interactions within the broader energy ecosystem. Such frameworks must be flexible enough to incorporate diverse technologies, scalable to handle community-wide simulations, and adaptable enough to model dynamic, real-world conditions. Bridging this gap is essential for advancing energy research, enabling more effective EMS development, and ultimately supporting the transition to sustainable energy systems.

One emerging platform with significant potential in REC modelling is CityLearn [23], a RL environment designed for studying energy management at the multi-building level. CityLearn allows the exploration of coordination strategies across a cluster of buildings, focusing on energy efficiency and load balancing. However, its current architecture falls short of fulfilling the broader requirements of a comprehensive DT. The platform lacks integration of essential energy assets, such as common household appliances like dishwashers, refrigerators, and washing machines. Additionally, it does not possess the capability to interface with real-world environments for data acquisition or to dynamically adapt to environmental conditions. Furthermore, CityLearn's granularity in simulating complex interdependencies and stochastic behaviours in energy supply and demand is limited, leaving gaps in its ability to replicate the intricate dynamics of real-world REC systems. The absence of hybrid simulation capabilities further restricts its applicability, as it cannot effectively combine real-time data with virtual simulations to reflect the evolving characteristics of energy systems in a dynamic and accurate manner.

1.1.4. Resource and Stakeholder Constraints

While DTs technology offers numerous advantages, its adoption is currently hindered by significant challenges that align with those faced by Artificial Intelligence (AI) and Internet of

Things (IoT) technologies. Key barriers include issues related to data standardization, data management, and data security, which complicate the integration of diverse systems and the safeguarding of sensitive information [24]. Furthermore, the transformation of legacy systems to incorporate DTs often demands updating outdated IT infrastructure, addressing connectivity issues, and implementing privacy and security measures [24].

The deployment of DTs is also constrained by high costs associated with initial implementation and ongoing operations [24]. Substantial investment is required in technology platforms, including sensors, software, and data infrastructure, along with maintaining data quality control and security solutions [24]. These expenses, coupled with the complexity of DT architectures, contribute to the significant fixed costs that discourage widespread adoption. Other notable challenges include the lack of standardised modelling approaches and integration issues with existing proprietary systems, which limit interoperability and scalability [25].

These resource and technological constraints are recognised by both the research community and industry stakeholders as critical factors likely to hamper the growth of the DT market. Addressing these challenges requires collaborative efforts to standardise frameworks, lower costs through innovation, and provide support for transitioning legacy systems. Without targeted strategies, the high fixed costs and complexity of DT solutions may continue to slow their deployment and limit their transformative potential.

1.1.5. Proposed Solution: Advancing Digital Twin Simulation for Renewable Energy Communities

To address the challenges of simulating RECs, several platforms, previously addressed on Section 1.1.3 were evaluated, with CityLearn ultimately selected as the foundation for this dissertation's proposed solution.

CityLearn offers key advantages that directly align with advancing DT simulations for RECs. Its focus on multi-building energy management provides a strong foundation for addressing the complexities of REC modelling. Moreover, CityLearn's modularity makes it highly adaptable, positioning it as an ideal platform for the proposed enhancements.

Building on this understanding, this dissertation proposes to expand CityLearn's existing framework to make it more aligned with the characteristics of a Digital Twin for RECs. By enhancing its scalability, integrating a broader range of energy assets, and incorporating stochastic elements to model uncertainties in energy behaviours, bridging the identified gaps in REC simulation, the improved CityLearn platform will bring it closer to a fully-realised Digital Twin, while still leaving room for future improvements as the field evolves.

Key advancements will include:

1. **Improved Temporal Granularity:** Refining the temporal resolution to balance computational efficiency and detail, enabling more nuanced analysis of REC operations.

2. **Adaptation of Existing Models for Temporal Granularity:** As a result of changing temporal granularity, existing asset models will need to be adapted to support different levels of resolution, ensuring consistent performance and accuracy.

3. **Modular Asset Integration:** Extending CityLearn to include more home appliances e.g. washing machines, fridges, etc. to improve its modelling capabilities and bring it closer to a DT.

4. **Incorporation of Stochasticity:** Introducing randomness in energy generation and consumption to simulate real-world variability, improving the effectiveness of energy management strategies tested within the platform.

5. **Data Export Functionality**: Enabling the export of simulation data and agent performance metrics in standardised formats (e.g., CSV, JSON), to facilitate post-simulation analysis, reproducibility, and integration with external tools such as data analytics pipelines or Machine Learning frameworks.

6. **Refined EV Modelling:** Shift the EV representation from a charger-centric perspective to a household-centric one, where homes intermittently receive EVs. This better reflects the residential focus of CityLearn and more accurately captures real-world energy dynamics involving EV usage.

7. **Comprehensive System-Level Improvements**: Addressing platform-wide issues and enhancing the overall coherence of the system by resolving cross-cutting problems, improving integration between components, and ensuring the reliability of the enhanced simulation framework.

8. **Testing Framework Development**: Establishing systematic testing methodologies to validate the correctness and reliability of both new and existing components, facilitating continuous improvement and supporting reproducibility of results.

These enhancements will position CityLearn as a pivotal tool for studying the domain of RECs, accelerating innovation in EMS development and the transition to resilient, sustainable energy systems.

1.2. Contributions and Benefits

The subsequent section explores the broader contributions of this dissertation, discussing its potential to drive both scientific advancements and societal benefits. It reflects on how addressing the challenges faced by current simulation tools can accelerate progress in sustainable energy systems and the implications of not overcoming these barriers, while considering diverse stakeholder perspectives.

1.2.1. Societal Contributions

This research contributes to the sustainability and resilience of energy communities through the development of a novel DT simulation framework specifically designed for integrating renewable energy systems. The framework offers a critical tool for addressing key challenges in the management and optimisation of energy resources, enabling the effective planning, testing, and deployment of advanced energy solutions in RECs. Without such solutions, energy communities risk facing persistent inefficiencies, higher costs, and reduced feasibility in managing renewable energy resources. This could lead to slower adoption of sustainable practices, hinder progress toward climate change mitigation goals, and perpetuate reliance on traditional energy systems, with significant environmental and societal consequences.

Through the development of a DT simulation framework, this work has the potential to benefit a wide range of stakeholders:

- **Investigators and Researchers** can utilise the DT framework as a valuable tool to test solutions and validate hypotheses before implementation. By providing a risk-free environment for experimentation, the framework accelerates the development of new approaches for optimising RECs.

- **REC Managers** within renewable communities may experience improved energy reliability and potentially lower electricity costs through community-level optimisations that balance load and enhance grid efficiency.

- **Energy consumers** in renewable communities can gain from improved energy reliability and potentially lower electricity costs as community-level optimisations balance the load and enhance grid efficiency.

- **Energy producers** supplying renewable energy to communities can benefit from enhanced forecasting and resource allocation capabilities. By simulating energy flows and demand, producers can better anticipate fluctuations, optimise energy distribution, and reduce wastage, thereby increasing the efficiency and profitability of their renewable energy operations.

- **Prosumers**—individuals or groups who both produce and consume renewable energy may benefit from higher returns on their energy investments by using surplus generation effectively within the community or selling it back at advantageous rates.

- **Policymakers** may use insights from such simulations to understand the benefits and scalability of energy communities, aiding in policy formulation that supports sustainable energy practices.

- **Society at Large** benefits from reduced greenhouse gas emissions, a crucial factor in combating climate change. Enhanced energy resilience at the community level contributes to the overall stability of national and regional energy systems.

By addressing the challenges currently faced by energy communities and its stakeholders, this research provides the following key societal contributions:

- **SC1** – Facilitates the design and testing of reliable EMSs for RECs, reducing risks and ensuring better performance before deployment.

- **SC2** – Accelerates the development of innovative energy solutions by providing a framework for safe, controlled testing.

- **SC3** – Enhances the scalability and integration of energy management strategies by simulating the dynamic interactions of diverse energy assets.

- **SC4** – Increases the operational efficiency of local energy systems by providing REC managers with insights into cost-effective solutions and investment strategies.

- **SC5** – Supports improved energy reliability for consumers, with optimised supply and demand balancing at the community level.

- **SC6** – Empowers prosumers to actively participate in energy management, maximizing the benefits of self-consumption and community energy sharing.

1.2.2. Scientific Contributions

The scientific advancements achieved through this dissertation address the gaps in knowledge and technology, identified on Section 1.1.3, offering comprehensive solutions for both academic and practical challenges in energy management. These contributions are as follows:

- **SctC1** – Advance the understanding of REC modelling by synthesizing insights from existing simulation methods and identifying gaps that hinder progress in the field.

- **SctC2** – Refine the methods used to determine temporal granularity in energy simulations, improving both the accuracy of decision-making processes and the computational efficiency of the models.

- **SctC3** – Advance the development of asset models by adapting them to incorporate more detailed and accurate representations, adjusting them according to the temporal granularity, and improving their adaptability, accuracy, and relevance for the complexities of dynamic energy systems.

- **SctC4** – Evolve advanced simulation environments, to provide effective platforms for analysing and optimising the interactions of flexible energy assets in RECs.

- **SctC5** – Enhance the scope and applicability of energy simulations by introducing realistic representations of diverse energy assets and devices, increasing their scientific relevance to real-world energy challenges.

- **SctC6** – Strengthen the reliability of energy simulation frameworks by integrating stochastic variability to emulate real-world uncertainties and improve the reliability of simulation outcomes.

- **SctC7** - Contribute to a better understanding of the interplay between individual building-level objectives and broader REC goals, enabling more effective strategies for energy management and optimisation.

- **SctC8** – Contribute to the open-source energy simulation community by developing and submitting technical improvements to a large-scale public project, supporting transparency, collaboration, and future advancements in energy management.

Beyond its direct contributions to the OPEVA project, this work has also formed the basis of a research paper that is intended for publication in a scientific journal.

- [Journal] Pina, R., Fonseca, T., ______ - “Smart Grid Optimisation in Renewable Energy Communities via Digital Twin Technology”, to be published to Energy Scientific Journal 2025.

1.3. Objectives and Research Questions

To ensure the research aligns with the scope of this work, this section formally outlines a set of project objectives and Research Questions (RQ).

1.3.1. Objectives

Following the proposed solution in Section 1.1.5, the **Main Objective (MO)** of this dissertation is defined as:

MO - Develop, implement, and validate a simulation framework based on DT technology that models RECs, focusing on scalability and integration of flexible energy assets.

To achieve this Main Objective, several **Sub-Objectives (SO)** are outlined as follows:

- **SO1** – Conduct a comprehensive review of the current state-of-the-art of RECs, DTs and Energy Simulation Frameworks

- **SO2** – Export simulation data and Key Performance Indicators (KPIs) to a centralised format, enabling stakeholders to visualise and analyse the simulator's output effectively through a dashboard being developed by a parallel project initiative [26].

- **SO3** – Investigate the impact of different temporal granularities (e.g., minute-by-minute, 15-minute intervals) on simulation accuracy and computational efficiency, particularly for RL and decision-making processes.

- **SO4** – Adapt asset models for changes in temporal granularity (e.g., hourly to 15-minute intervals) using real-world energy data to keep simulation accuracy and realism.

- **SO5** – Design and implement a DT simulation platform capable of modelling the dynamic interactions of diverse energy assets, including household appliances, and energy storage systems to emulate realistic REC scenarios.

- **SO6** – Incorporate individual building-level objectives (e.g., cost reduction, self-consumption maximization) into the simulation framework and evaluate their impacts on overall REC performance and energy management strategies.

- **SO7** – Replace static values in the simulation with stochastic variability to better emulate unpredictable real-world conditions, increasing the reliability of the simulation outcomes.

- **SO8** – Validate the correctness and reliability of the simulation framework through systematic testing and verification.

- **SO9** – With the goal to publish it open source, collaborate with the project maintainers, from the University of Texas, to align development efforts, integrate contributions, and address global issues and bugs in the framework, working for a more resilient and interoperable solution

- **SO10** - Update the EV simulation model to reflect a more realistic setup, with stationary charging infrastructure and mobile EVs, replacing the earlier approach where chargers adapted to EVs. This shift improves the representation of mobility and asset interaction in REC scenarios.

1.3.2. Research Questions

The primary question of this research was to understand how DT technology can be enhanced to create more accurate and effective simulations for RECs. To address this, it was first necessary to examine what DTs are and how they function within energy systems. Following this, how the integration of smart devices and EVs could impact energy management within these communities was explored.

From this investigation, several specific research questions emerged:

- **Q1:** How can a DT-based simulation framework be designed and implemented to effectively model RECs with scalable integration of flexible energy assets?
- **Q2:** How does the choice of temporal granularity (e.g., minute-by-minute, 15-minute intervals) impact the accuracy and computational performance of simulations?
- **Q3:** How can diverse assets such as energy storage systems, EVs, and household appliances (e.g., refrigerators, dishwashers) be effectively modelled within a scalable DT platform to emulate real-world energy behaviours?
- **Q4:** How can the simulation framework incorporate individual building goals, such as cost reduction or self-consumption maximization, and what impact do these personalised objectives have on overall energy management in RECs?

1.3.3. Methodology: Design and Creation Approach

This dissertation uses the Design and Creation methodology [27] to develop, implement, and validate a novel DT simulation framework for RECs. By bridging theoretical exploration with practical application, this approach ensures the framework addresses critical aspects such as scalability, adaptability, and the integration of flexible energy assets.

to develop, implement, and validate a novel DT simulation framework for RECs, emphasizing scalability, adaptability, and integration of flexible energy assets.

Starting with problem definition, identifying gaps in existing DTs via a comprehensive literature review, the process designs a scalable framework for simulating energy production, consumption, and flexibility. Real-world datasets drive realism, while iterative scenario analysis and benchmarking refine performance. Testing and optimisation validate the framework's sustainability and operational reliability.

Design and Creation excels here by blending theoretical exploration with practical tool-building and stakeholder feedback, accommodating the stochastic, interconnected nature of RECs. Continuous testing, cross-validation, sensitivity analyses, and stakeholder input minimise bias and ensure adaptability.

By integrating engineering, data science, and machine learning, this interdisciplinary approach delivers a validated, real-world solution for energy management. The resulting framework offers a structured, iterative path to meet current and future REC challenges.

To ensure that this work was conducted with the highest ethical standards, several key guidelines and frameworks were followed throughout the project, which are presented in the following subsection.

1.4. Ethical Considerations

This study was carried out as a component of the OPEVA [1] initiative, a project affiliated with the European Union that aims to promote cooperative innovation and advance sustainable energy technologies. To create DT solutions for communities using renewable energy, the project mainly relied on data supplied by partner businesses. All data usage adhered to the Data Governance Act (Regulation (EU) 2022/868) [28], which guarantees safe and open industrial data exchange throughout the European Union.

The IPP Code of Conduct [29] served as the ethical foundation for all facets of this dissertation. The significance of academic and professional integrity is emphasised in this Code especially as stated in Article 6 which lays out precise ethical guidelines. As required by clause 2.8, this project made sure that all sources were properly attributed and referenced. Additionally as mandated by clause 2.9 [29] no previously published or displayed work was presented as original without explicit acknowledgment. As stressed by clause 2.11 no fabricated or deceptive data was used, and cooperative efforts were duly approved and acknowledged in line with clause 2.10. The results shown in this document are genuine and well-documented and these principles were consistently applied [29].

Initially, Article 8 of the IPP Code of Conduct was addressed through a formal declaration of commitment, underscoring the integrity of the academic work. This declaration reinforced the project's originality and its unwavering adherence to the ethical standards established by P. PORTO [29].

The dissertation also complied with the ethical principles of Article 10, which required careful and balanced investigations, accurate and comprehensive citation of all referenced materials, and the transparent presentation of results to ensure their verifiability and reproducibility [29].

Given that this dissertation is a component of a master’s program in software engineering the IEEE Code of Ethics [30] was yet another essential ethical framework that influenced this work. To ensure that all work maintains the highest standards of quality the IEEE framework places a strong emphasis on values like honesty integrity and transparency. It also highlights the importance of impartiality, avoidance of conflict of interest, and assurance that personal biases do not undermine the fairness of decision-making processes.

Throughout this project, AI-based tools, primarily large language models such as OpenAI’s ChatGPT, were used as a productivity aid to support specific technical tasks during development. In particular, AI tools assisted in debugging and refining code snippets, generating examples for synthetic data creation, and developing visualisation scripts for data plots (such as those presented in Section 6). For instance, AI was used to review and optimise Python scripts that generated synthetic load profiles based on publicly available, open-source datasets, such as those provided by the CityLearn framework. As CityLearn is open source, no proprietary or confidential data was used or exposed during this process. Moreover, no sensitive data of any kind was shared with AI systems at any stage of the project. The use of AI remained strictly limited to coding assistance and technical clarification, always applied with sound judgement and, where appropriate, peer review. All design decisions, data analysis, and academic contributions were carried out independently by the author, in collaboration with colleagues and academic advisors, ensuring full adherence to the IPP Code of Ethical Conduct and preserving the integrity of this dissertation.

In alignment with these principles, realistic and well-defined project objectives were established through collaborative discussions with the advisors of this project. Ethical, economic, cultural, legal, and environmental considerations were carefully addressed, and open-source and freely accessible resources were utilised whenever possible. By integrating these frameworks, the project exemplifies a commitment to ethical rigour and professional excellence while contributing meaningfully to the field of renewable energy technologies.

With the ethical foundations firmly presented, the following chapter presents a detailed overview of the dissertation's structure, offering the reader a clear roadmap through the successive stages of inquiry, design, implementation, and evaluation that define this research contribution.

1.5. Document's Structure

This dissertation is organised to systematically guide the reader through the research journey, beginning with foundational concepts and culminating in the implementation and validation of the proposed solution. The structure reflects the iterative nature of the *Design and Creation* methodology while ensuring a logical progression from problem identification to solution development.

The document opens with **Chapter 1 (Introduction)**, which establishes the broader context of RECs and the critical challenges they face in energy management. The limitations of existing simulation tools are identified, laying the foundation for the proposed enhancements to the CityLearn platform. The chapter also outlines the research objectives, expected contributions, and ethical considerations that shape the study.

Subsequently, **Chapter 2 (State of the Art)** goes into the existing pool of knowledge surrounding RECs, smart grids, and digital twin technologies. By synthesizing relevant literature, this chapter not only contextualises the research within current academic discourse but also pinpoints the specific gaps that this dissertation aims to address. The literature review is mapped to the research questions, ensuring a targeted examination of prior work.

With the theoretical framework established, **Chapter 3 (Project Planning)** transitions into the practical aspects of the research. This chapter details the systematic approach taken to transform research goals into actionable tasks, including work breakdown, scheduling, and risk management. It emphasises the alignment with the *Design and Creation* methodology, highlighting how iterative development and continuous evaluation are part of the project's execution.

The focus then shifts to **Chapter 4 (System Design)**, where the conceptual framework begins to take form. This chapter bridges the gap between planning and implementation by introducing a set of FURPS+ requirements [31] that guide the architectural enhancements to CityLearn. Key design innovations are presented, such as the modular configuration schema for diverse energy assets, support for sub-hourly temporal resolution, and the integration of stochastic modelling to better reflect real-world conditions.

Chapter 5 (Implementation) provides a detailed account of how the proposed design was implemented. It describes the technical modifications made to the CityLearn platform, including the introduction of declarative noise modelling for variability, hierarchical initialisation for multi-scale energy dynamics, and comprehensive testing protocols to ensure system resilience. This chapter underscores the practical challenges encountered and the solutions devised to overcome them.

The efficacy of the enhanced simulator is then evaluated in **Chapter 6 (Evaluation)**, where the platform's performance is assessed against the baseline CityLearn framework. Using a combination of real and synthetic datasets, this chapter examines critical KPIs such as energy efficiency, cost savings, carbon footprint reduction, and operational flexibility. The analysis covers both static and stochastic scenarios, providing a view of the framework's capabilities and limitations.

Finally, **Chapter 7 (Conclusion)** synthesises the findings, reflecting on the dissertation's contributions to the field of renewable energy management. It discusses the broader implications of the research, acknowledges its constraints, and suggests avenues for future work to further advance the development and adoption of digital twin technologies in energy communities.

1.6. Conclusions

Having now introduced the structure of this dissertation, it can be said that this introductory chapter has laid the essential groundwork for understanding the critical challenges and motivations driving this research. The transition to RECs represents a fundamental shift in how energy systems are managed, bringing both opportunities and complexities. As decentralised energy production becomes more prevalent, the need for advanced simulation tools capable of modelling these dynamic systems has never been more urgent.

In a nutshell, the limitations of existing simulation platforms, such as insufficient scalability, lack of asset heterogeneity, and the absence of standardised benchmarking, highlight the necessity for an adaptable DT framework. By enhancing CityLearn with finer temporal granularity, expanded asset models, stochastic elements, and improved EV integration, this research bridges the gap between theoretical and real-world energy management.

Consequently, the proposed solution not only addresses technical gaps but also aligns with broader societal and scientific goals. By enabling more accurate, risk-free testing of EMS, this work contributes to the stability, efficiency, and sustainability of RECs, ultimately supporting the global transition toward renewable energy.

The following chapters will systematically explore these themes, beginning with a comprehensive review of the State of the Art in the next chapter (Chapter 2), which will contextualise this research within existing academic discourse and further justify the proposed enhancements to CityLearn. From there, the dissertation will progress through project planning, system design, implementation, and validation, culminating in a discussion of the broader implications for energy research and policy.

By advancing Digital Twin technology for RECs, this research seeks to provide a scalable, adaptable, and realistic simulation environment, one that empowers researchers, policymakers, and energy stakeholders to accelerate the adoption of sustainable energy solutions.

2. State of the Art

This chapter introduces essential terminologies such as renewable energies, smart grids, RECs, DTs, and simulators, then moves on to research questions from Section 1.3.2 by reviewing relevant literature. It examines modelling RECs' challenges concerning these technologies for optimising their performance in energy systems. The chapter focuses basically on how digital tools enable renewables integration and management, flagging those that may direct their use in increasing efficiency and sustainability.

2.1. Renewable Energies and Prosumers

Renewable energy sources play a crucial role in global issues like climate change and energy insecurity [32]. Technologies like solar panels, wind turbines, or biomass ensure decentralised energy generation in transitioning to a cleaner, low-carbon energy system [33]. The concept of *prosumers* [34], [35], individuals or entities that both produce and consume energy, has gained significant prominence. Prosumers improve electricity use efficiency and sustainability by utilizing microgeneration technologies like roof-mounted solar panels and small-scale wind turbines to fulfil self-consumption and contribute to the grid. This shift allows prosumers to access energy markets where they benefit from the use of cheaper consumption, a degree of autonomy, and sales of excess energy [35]. These developments have great meanings regarding their effects on sustainability and engagement with energy poverty [34].

Despite the potential benefits offered by prosumerism, it is not without challenges, such as regulatory restraints, high initial capital costs, and minimal public awareness in some regions [35]. However, supportive policies and financial incentives, such as feed-in tariffs, tax incentives and subsidies [36], have significantly accelerated its growth in countries like the Netherlands which from 2015-2020 doubled its number of prosumers from 500,000 to over 1 million [37] and Poland which went from 51,000 prosumers to 847,000 in 2021 [37]. Integrating prosumers into energy systems aligns with broader objectives of achieving sustainable development and enhancing decentralised energy access [38].

2.2. Smart Grids

The increasing role of prosumers shows that there is an urgent call for smarter and efficient energy systems. Within this context, the concept of smart grids comes at high relevance. In simple terms, smart grids are the evolution of electricity networks that integrate state-of-the-art digital technologies, offering an enhanced energy efficiency, reliability, and sustainability of energy supply [39]. They will increase real-time monitoring and two-way communication between utilities and consumers, enabling the better integration of renewable sources and Distributed Energy Resources (DERs) [40]. Smart infrastructure allows intelligent demand-side management, grid stability, and uses of new energy paradigms like prosumerism [41].

Key innovations within smart grids include smart meters [42], advanced distribution management systems [43], and blockchain technology for peer-to-peer energy trading [44]. These empower users to understand energy utilization in real time and reduce costs effectively by participating in energy marketplaces. By and large, smart grids also support energy storage systems and microgrids [45], provide resilience from outages [46], [47], and allow energy independence [48].

2.3. Renewable Energy Communities

Building on the concepts of prosumers and smart grids, energy communities represent cooperation in the generation, management, and consumption of energy. These communities are typically created by groups of people or bodies willing to cooperate in the production and sharing of renewable energy within their local networks [49]. Through aggregation of resources, energy communities gain an economy of scale, hence enhancing access to renewable energy technologies at lower costs by its members [3]. In this respect, their model will have an added advantage when addressing matters of energy poverty-a very key aspect where surveys show that 6.9% of EU citizens cannot afford to keep their homes adequately warm [50]. Reducing the cost of electricity will yield such positive indicators as a result of this model, where energy costs could be lowered by 10%, hence directly benefiting energy-insecure households [50].

Energy communities are among the manifestations of circular and sharing economy principles and contribute to social cohesion as well as environmental care [51]. In several countries, these schemes fall within supportive regulatory frameworks enabling collective self-consumption and/or decentralised energy trading [52]. More recently, energy communities entered a fast growth path in Europe, further spurred by enabling policies of the *Clean Energy for All Europeans* package [52]. These communities can also help achieve carbon neutrality while increasing local energy security and promoting economic development [3].

Interlinking these three pillars, renewable energy sources, smart grids, and RECs, modern energy systems now undergo a shift toward sustainable, decentralised, and participatory futures. Such frameworks contribute to supporting global goals like those under the Sustainable Development Goals (SDGs) [53] which relate to Clean Energy, Climate Action, and Sustainable Urbanisation.

2.4. Simulators

Simulations are indispensable in supporting the three pillars of renewable energy systems, renewable energy sources, smart grids, and energy communities, by providing critical tools for designing, analysing, and optimising energy networks [54]. For decades, simulators have played a central role across various industries, creating virtual environments to model complex physical systems, optimise processes, and predict performance outcomes [55]. Their primary purpose is to reduce uncertainties and mitigate risks associated with physical trials by enabling in-depth analysis of system behaviour in a controlled, virtual setting [56]. This ability to model, simulate, and test systems under diverse conditions is vital for sectors such as aerodynamics, urban planning, manufacturing, and especially energy systems [56]. In the energy domain, simulators provide a solid foundation for developing and refining electrical grids and renewable energy systems. These tools are crucial for addressing the challenges of integrating renewable energy sources, ensuring stability, and achieving efficiency, which are all essential components of RECs.

In energy applications, simulators have evolved into highly specialised tools for modelling the dynamics of renewable energy generation, consumption, and storage. These tools enable engineers and operators to evaluate system performance under different operational scenarios, including variations in weather patterns, energy demand, and grid conditions [54]. Given the intermittent nature of renewable energy sources like solar and wind, simulators are essential for forecasting energy production variations and accurately predicting supply and demand dynamics. This capability is particularly valuable for optimising the integration of renewables into grids, ensuring stability and efficiency. For example, simulations can assess the impact of integrating new energy storage systems or analyse how different energy mixes affect grid reliability under fluctuating demand [57].

Simulators also play a crucial role in identifying vulnerabilities, such as potential grid failures or inefficiencies in energy distribution. By creating virtual replicas of physical infrastructures, these tools enable energy planners to conduct "what-if" analyses, exploring the effects of various failure modes or disturbances on grid stability. Additionally, simulations help determine optimal configurations for placing energy storage units, renewable generation sources, or smart grid technologies [58].

Although both traditional simulations and DTs (discussed in Section 2.6) aim to replicate physical systems in virtual environments, they differ significantly in their capabilities and applications. Traditional simulators model system behaviour based on historical data, facilitating predictions, analyses, and scenario testing. However, their static nature limits them to predefined inputs, restricting their adaptability in real-time scenarios.

DTs, on the other hand, extend simulation capabilities by incorporating real-time data and feedback loops. These tools dynamically synchronise virtual models with sensor data from the physical system, enabling real-time monitoring, predictive analytics, and actionable insights. This two-way interaction allows DTs not only to replicate systems but also to actively influence their operation [59].

As these technologies continue to evolve, the distinction between simulators and DTs becomes increasingly significant. These two systems complement each other: simulators provide the basic model, while DTs use real-time data to facilitate quicker decision-making. It is an important synergy that in renewable energetics will enhance grid stability, optimise resource usage, and will allow for much easier integration of wind or solar sources.

One notable example of a modern simulator designed to address the specific challenges of renewable energy systems and energy communities is CityLearn, which is examined in detail in the following section (Section 2.5). While traditional simulators offer static frameworks for testing and evaluating scenarios, CityLearn introduces a modular, adaptive environment designed to facilitate intelligent control strategies through RL. As such, CityLearn embodies many of the advantages discussed in this section, including scenario testing, performance evaluation, and predictive optimisation, while also paving the way toward the dynamic, real-time interactions more commonly associated with DTs (discussed in Section 2.6). This progression illustrates how simulation platforms are evolving to meet the increasing complexity and interactivity demanded by modern energy systems.

2.5. CityLearn: A Simulator for Energy Communities

Building on the general role of simulators in energy systems and the derivation of system requirements outlined in Section 1.3.1, this dissertation centres itself on the CityLearn platform. Developed at the University of Texas at Austin, CityLearn is an open-source simulation environment specifically designed to benchmark MARL algorithms in the context of urban building energy management. It provides a flexible and scalable framework that supports high-fidelity modelling and reproducible experimentation. These qualities were essential in its to CityLearn's widespread adoption within the academic and research community. The platform has been employed in numerous peer-reviewed studies [60], doctoral theses [61], and benchmarking efforts focused on topics such as energy flexibility, demand response, and intelligent control. CityLearn has proven to be a credible and versatile tool, with demonstrated impact across areas including smart grid coordination, occupant-centric energy management, cluster-level control of building systems, and MARL evaluation.

CityLearn is built on the OpenAI Gym interface, a popular toolkit in the RL community, and its core is based on the concept of the environment [62]. Represented by the Env class, this environment encapsulates the dynamics of tasks, such as managing the energy of a single building or a group of buildings and dictates how an agent can interact with and influence the system [62].

The agent's actions are informed by observations, which provide insights into the current state of the environment [63]. These observations typically include variables such as energy consumption, building temperature, and weather conditions. Based on this information, the agent calculates an action value that determines the decision it will make, such as activating an EV charger or turning off the heat pump once the desired temperature is reached. As the agent interacts with the environment, its actions induce changes that are reflected in the subsequent states. Over time, the agent learns which actions lead to more favourable outcomes, such as improved energy efficiency, and refines its decision-making strategy through feedback from past experiences. In Figure 1 a schematic of this interaction can be viewed.

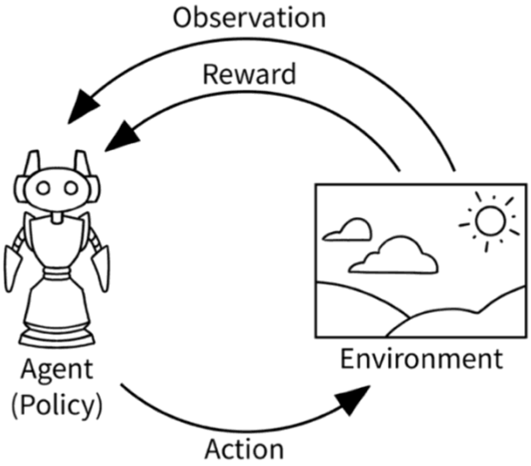


Figure 1 - Inner working of the Gymnasium interface (From [62])

The agent-environment interaction can be systematised with Figure 2, it follows a closed loop and is driven by a few core functions. Env.reset() initialises the environment by setting up building models, loading historical data, and resetting internal buffers, ensuring a consistent starting state. Env.step(action) governs the main interaction loop: observations from all buildings are aggregated and passed to agents, which select actions based on the current state. These actions are applied to the buildings, and agent-specific rewards are calculated to quantify the effectiveness of actions toward objectives like minimizing energy costs, emissions, or discomfort. These rewards guide learning through trial and error. Finally, Env.render(), aligned with this dissertation's goals, can be used to visualise the environment for monitoring and debugging.

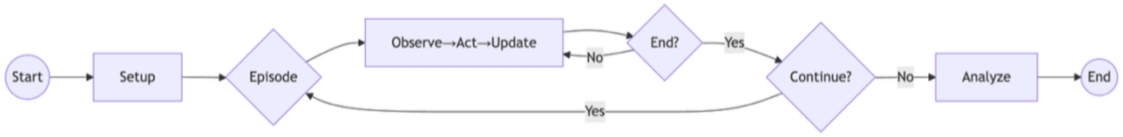


Figure 2 - Flowchart of the simulation loop and agent interaction pipeline.

The simulation also involves defining the observation space and action space. The observation space describes what the agent can perceive about the environment, such as energy usage or external conditions like the weather [62]. On the other hand, the action space specifies the set of valid actions the agent can take, such as modifying temperature settings or controlling energy storage levels [62]. Both spaces define the boundaries within which the agent can operate, and understanding these limits is crucial for ensuring proper interaction between the agent and the environment [62].

2.5.1. Core Architectural Components

As seen in Figure 3, the architecture of CityLearn divides the energy community simulation into four key subsystems: the Environment Engine, Agent Orchestrator, Reward Calculator, and Data Pipeline. Each of these modules plays their role in managing time-series data, control actions, and thermodynamic dynamics in the simulation. In the following sections, a detailed description of each component is presented, highlighting their specific functions within the system.

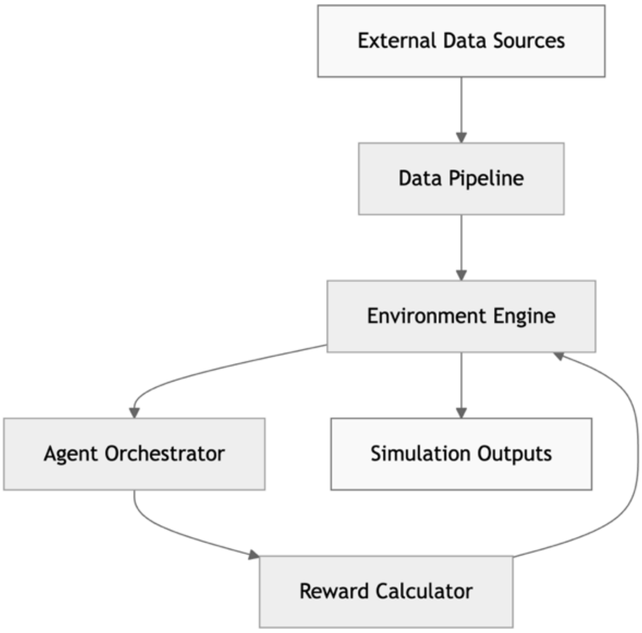


Figure 3 - CityLearn's Architecture Overview

Environment Engine

One of the most crucial subsystems is the Environment Engine, which serves as the temporal and physical core of the simulation. It governs the progression of simulation time, interfaces with external data sources such as weather conditions, electricity tariffs, and carbon intensity, and applies physics-based models to subsystems like HVAC, energy storage systems, and building thermal envelopes. A key limitation of the original version is its rigid use of hourly timesteps, which restricts the precision of control strategies and the fidelity of the models.

In this component, dynamic timestep resolution will be introduced, supporting sub-hourly intervals (such as 15-minute timesteps). New physics-based models, including refrigerators and washing machines, further enhance the environment. This development is discussed in more detail in Section 4.3.2 and 4.3.4.

Agent Orchestrator

On the other hand, the Agent Orchestrator manages the decision-making process for each asset in the simulation. Each agent receives a vectorised observation, which includes information about its current state, such as historical and forecasted energy demand, the state-of-charge (SOC) of its storage systems, and projected solar energy generation, and then determines appropriate control actions like charge/discharge rates and HVAC setpoints.

In the original implementation, the reward feedback was shared among all agents, following a cooperative, monolithic model. This dissertation introduces an extension which will allow agents to operate under heterogeneous objective functions, meaning that each building can prioritise different goals. For instance, some buildings may focus on reducing peak energy demand, while others may optimise for carbon intensity or thermal comfort. This enhancement introduces greater experimental flexibility and is especially useful for simulating community-scale optimisation problems. This is further discussed in Section 4.3.3.

Reward Calculator

Subsequently, the Reward Calculator assesses the performance of each agent and computes both individual and collective reward signals. Similarly to the previous, the module only supported only one objective for the whole environment, such as reducing community-wide peak demand. This functionality has been expanded to allow for custom, and even conflicting, agent-specific goals to be defined on a per building basis. This is also discussed in Section 4.3.3.

Data Pipeline

CityLearn's Data Pipeline is responsible for ingesting structured time-series data, performing schema validation, and ensuring the integrity of the simulation. It aligns datasets, checks for physical plausibility (e.g., ensuring no PV generation during the night), and verifies that the data conforms to the required schema. While retaining this validation mechanism, the Data Pipeline has been extended to support stochastic load profiles, generated using statistical models such as the Gaussian or Markov process. This is further discussed in Section 4.3.5.

2.6. Digital Twins

Subsequently and as mentioned earlier, the distinction between DTs and simulators is often blurred. To understand the implications of DTs, it is essential to explore the concept of twin modelling, which has long been integral to human innovation. While some traditional methods, such as wind tunnels for aerodynamics [64], remain relevant, the emergence of information technology has revolutionised modelling with the introduction of the DT concept.

DTs, first proposed by Professor Michael Grieves at the University of Michigan in 2003, marked a significant leap forward in modelling techniques. Initially termed a "mirrored space model," the term "Digital Twin" gained prominence in 2011 through Grieves' book Almost Perfect: Driving Innovation and Lean Products through PLM [65]. The DT framework comprises three key components: the physical product in the real world, its virtual counterpart in the digital domain, and the data connection linking the two. This data link enables real-time synchronisation between the physical and digital worlds, facilitating advanced simulations, analyses, and predictive insights without disrupting real-world operations [65]. Elements of this DT-style feedback loop are beginning to appear in advanced simulators like CityLearn (as seen previously in Section 2.5.1), which support dynamic environmental modelling and agent-specific objectives, blurring the line between static simulation and real-time digital interaction.

DTs surpass traditional physical models by eliminating constraints such as location and material costs. They can simulate intricate details and complex functions while excelling in processing and exchanging information. Furthermore, their ability to visually represent data and processes makes them accessible to both experts and broader audiences, enhancing learning and supporting informed decision-making [66].

In modern applications, DTs perform diverse functions, from product design to real-time monitoring and post-production updates. For example, during product development, a DT can simulate scenarios to optimise processes and reduce time-to-market. Post-production, the DT continues to monitor performance, enabling predictive maintenance and operational efficiency.

2.6.1. Dimensions and Functions of Digital Twins

The traditional three-dimensional model of DT systems has recently expanded into a five-dimensional framework [67]. This enhanced model builds on the original components, Physical Entity (PE), Virtual Entity (VE), and Connection (CN), by introducing Digital Twin Data (DD) and the Service System (SS). These additions emphasise the critical role of data and dynamic operations within modern DT systems, making them more versatile and adaptable [67].

The five dimensions are deeply interconnected, as illustrated in Figure 4. At the core is the Physical Entity, representing the real-world object or system. Sensors embedded in the physical entity collect real-time data, which forms the foundation for its digital counterpart, the Virtual Entity. This digital representation leverages real-time and historical data to conduct simulations, support decision-making, and optimise the physical system without interference [22], [68].

Digital Twin Data serves as the system's core, enabling frictionless interaction between physical and virtual entities. This data, enriched through analysis, is critical for generating insights and facilitating effective system operations. Complementing these elements is the Service System, which provides tools, applications, and workflows for monitoring, analysis, and management, ensuring smooth interaction between dimensions [69].

Connections between these dimensions allow bi-directional communication, ensuring that changes in the physical entity are reflected in its virtual counterpart and vice versa. For example, IoT sensors enable data flow from the physical entity, while insights from simulations are applied to optimise the physical system. The Service System further enhances this interaction by supporting diagnostics, real-time monitoring, and performance optimisation [69].

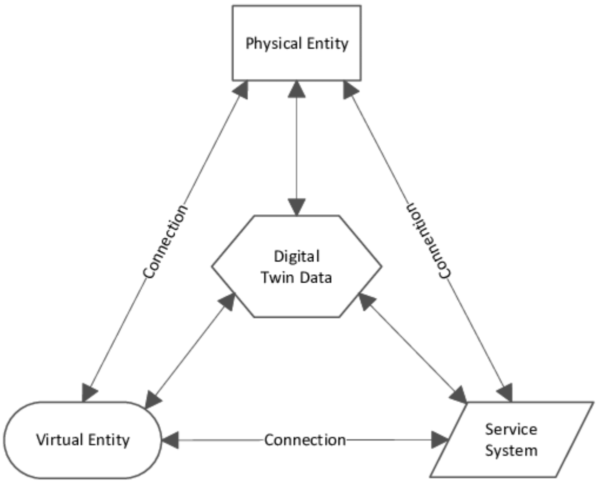


Figure 4 - Five-dimension model of Digital Twins [70]

The integration of these dimensions provides a solid framework for practical applications, including RECs. By synchronising physical entities, virtual models, data, and services, DTs can transform energy management and optimisation processes. This sets the stage for exploring the state-of-the-art in DT applications for RECs. To ensure a comprehensive investigation, this research employs the PRISMA framework, a methodology for identifying, selecting, and reviewing relevant literature to address key questions in DT-based simulation frameworks.

2.7. Literature review

This section explores into the literature relevant to the research questions outlined in Section 1.3.2, offering a thorough exploration of the existing knowledge base. The review is organised around each research question, addressing them individually to ensure clarity and focus.

To address them it was used the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) framework, a structured methodology widely used in systematic reviews. Originally introduced in 1997, PRISMA has undergone multiple updates, with its most recent revision in 2020 [71], [72]. Designed to promote transparency and rigour, PRISMA provides a detailed process for identifying, selecting, and analysing relevant studies. It starts by defining the research topic and formulating questions, which are used to extract key terms for constructing search queries. These queries are applied to databases with filters for language, document type, and publication date. The resulting collection of documents is screened in successive stages, first by title and abstract, and then through a detailed review, to ensure relevance and quality.

In this literature review, the PRISMA framework is applied selectively. For some research questions, a systematic and comprehensive PRISMA-based review is conducted, ensuring thoroughness and methodological precision. For other questions, a more relaxed approach is taken, involving a smaller, curated set of studies that provide key insights without exhaustive documentation. This combination ensures a flexible yet effective exploration of the literature.

By addressing each research question individually, the review highlights key themes, gaps, and opportunities critical to the development of the DT-based simulation framework for RECs.

The following section presents the results of the systematic review conducted for Research Question 1: “How can a Digital Twin-based simulation framework be designed and implemented to effectively model RECs with scalable integration of flexible energy assets?” This question, as described in Section 1.3.2, is central to the dissertation's objectives.

2.7.1. Research Question 1

For this research question, a systematic review following the PRISMA framework was done. Having been briefly defined previously on Section 2.7, to ensure a transparent and rigorous methodology. Details regarding the PRISMA flowchart, search strategy, inclusion/exclusion criteria, and the review process are provided in the Appendix A.

Across the reviewed studies from this systematic review, several themes emerge that are deeply relevant to the dissertation, guiding both what should be replicated and the pitfalls that should be avoided.

A consistent takeaway is the emphasis on modular, layered architectures. The papers highlight the importance of a dedicated data layer to enable real-time system monitoring and data interoperability [73], [74].

The papers explore various architectural frameworks, including two-layer, three-layer, and more advanced four-layer and hyper-layer designs [73], [75]. The four-layer and hyper-layer architectures are particularly notable for their ability to address complex applications in DT systems, specifically within cyber-physical systems (CPS).

The four-layer architecture consists of the physical layer, which includes both the physical assets and associated data, the information layer that connects the physical and DTs, the model layer which acts as a virtual mirror of the physical entity, and the application layer that facilitates optimisation and management tasks [73], [75].

In contrast, hyper-layer architectures extend beyond four layers, offering enhanced flexibility and scalability. For example, a customizable architecture was proposed in 2020 [76], that integrates advanced technologies like big data, AI, and cloud computing, as a form of being a DT for Industrial Energy Systems. By incorporating multiple layers and diverse technological tools, hyper-layer DT architectures enable real-time data interaction, decision-making, and system optimisation [73]. This makes them particularly suited for complex systems, such as RECs or industrial energy systems, ensuring that DTs can evolve and adapt to incorporate new technologies as they emerge.

Both approaches prioritise scalability and flexibility, making them ideal for integrating new energy assets without requiring a complete system overhaul. In the context of the proposed solution, adopting such modular architectures means the system will be capable of incorporating new energy assets as they emerge. This ensures that the design remains adaptable, a crucial feature for RECs, enabling them to evolve and integrate diverse energy sources with ease.

The need for effective data interoperability is a recurring challenge across all the studies. Integrating diverse data sources, from IoT-enabled sensors to AI-driven analytics, is essential for dynamic energy management frameworks [73], [76]. The DT must address this by including a well-structured data processing layer capable of standardizing and harmonising inputs from various systems. Using middleware or standardised protocols will be essential to ensure communication between components, avoiding common pitfalls related to data fragmentation [73], [76]. This aligns closely with SO4, as well as SO3.

Predictive analytics, including techniques such as deep learning, RL, and optimisation algorithms like white shark optimisation, have already been extensively explored and implemented in existing DT frameworks to forecast energy loads and optimise grid operations [74], [77]. Given the considerable progress made in this area, including on CityLearn's framework [23], there is no need to focus heavily on this aspect in this dissertation. The work on this field has largely been accomplished, and the primary focus of this dissertation will be on the other aspects outlined in Section 1.3.

Cybersecurity plays a pivotal role in DT systems, as emphasised by some papers, who emphasise the integration of cyber-physical security measures using techniques such as deep RL and collaborative algorithms to detect and mitigate cyber threats [74], [75]. While these insights are crucial for ensuring the resilience of DT systems, they fall outside the immediate scope of this dissertation, which focuses on optimising energy management within RECs and the modelling of energy assets. However, the importance of cybersecurity in future developments of DT systems remains acknowledged.

Energy trading and dynamic management emerge as additional dimensions where DT frameworks add significant value. Dynamic pricing and collaborative decision-making optimise energy consumption and support trading within renewable energy systems [75] [77]. These functionalities align closely with SO6, as they contribute to achieving building-level objectives like cost reduction and self-consumption maximization, and SO3, by requiring the integration of real-world energy consumption, production, and flexibility data to simulate realistic trading scenarios.

In essence, the studies reviewed underscore critical challenges in scalability, system complexity, and data interoperability within DT frameworks. To navigate these, the approach to be used centres on a modular design that harmonises complexity with usability, enabling both flexibility and scalability without sacrificing performance.

The literature underscores the value of predictive capabilities and dynamic decision-making in optimising energy consumption and facilitating energy trading in RECs. While cybersecurity remains a vital consideration, this dissertation will focus on creating a flexible, predictive simulator tailored to energy management priorities.

Key takeaways include the necessity of modularity, the integration of predictive analytics, and the use of real-world data to ensure realistic simulations. By adopting standardised protocols and middleware, the DT will ensure data interoperability, avoiding the fragmentation observed in prior studies. Above all, the development will prioritise a balance between model sophistication and user interpretability.

An essential factor in this development is the choice of temporal granularity for simulations. The level of detail, minute-by-minute or 15-minute time steps, in energy system models impacts accuracy, operational insights, and computational efficiency.

The next subsection deals with the problem of how much the choice of temporal granularity influences the accuracy of simulation and computational performances, in view of implications for the design of DT frameworks in renewable energy systems.

2.7.2. Research Question 2

Research question 2 asks: "How does the choice of temporal granularity (e.g., minute-by-minute, 15-minute intervals)" impact the accuracy and computational performance of simulations?". Temporal granularity, as previously mentioned, is a crucial factor in the modelling and simulation of energy systems, especially within renewable energy contexts, where its influence spans across prediction accuracy, operational insights, and computational efficiency. As renewable energy systems, such as PV arrays, battery storage, and community-scale grids, often exhibit dynamic behaviour over short timescales, fine-grained data resolution becomes critical for both practical applications and theoretical analysis.

Energy production and consumption exhibit significant variability at minute-level timescales, with household appliances like dishwashers and washing machines displaying transient behaviours that cannot be accurately captured at coarser temporal resolutions. Dishwashers, for instance, experience large energy spikes during water heating phases, followed by periods of lower consumption during subsequent cycles. These fluctuations are best observed through high-resolution simulations, such as minute-by-minute data, which enable precise load forecasting and optimised operations. On the contrary, at a coarser time interval, for instance, data on an hourly basis smooth out such fluctuations and may misrepresent the behaviour of the system, therefore making decisions even less accurate.

For example, as illustrated in Figure 5 a), the power consumption of a washing machine is not constant. Over a period of one hour, between 13:00 and 14:00, the washing machine appears to consume about 12 kWh, but this average value is deceptively wrong. Looking at this signal with finer time resolution reveals that during certain moments, such as the spin cycle, power consumption can peak at 900 W, while at other times it fluctuates between 100 and 200 W. These variations are only clearly visible when observed at finer temporal resolutions. High-resolution load profiles are crucial for accurately capturing these dynamic fluctuations caused by the turning on and off of appliances, making them significantly more representative of real-world energy consumption compared to low-sampling-rate models [78]. Given this context, and the fact that real-world appliance behaviour shows such complex, time-varying characteristics, to base our simulator on real consumption data was chosen, rather than relying on fixed mathematical models for the washing machine. Further details on the implementation of real consumption data are provided in Section 6.1.3.

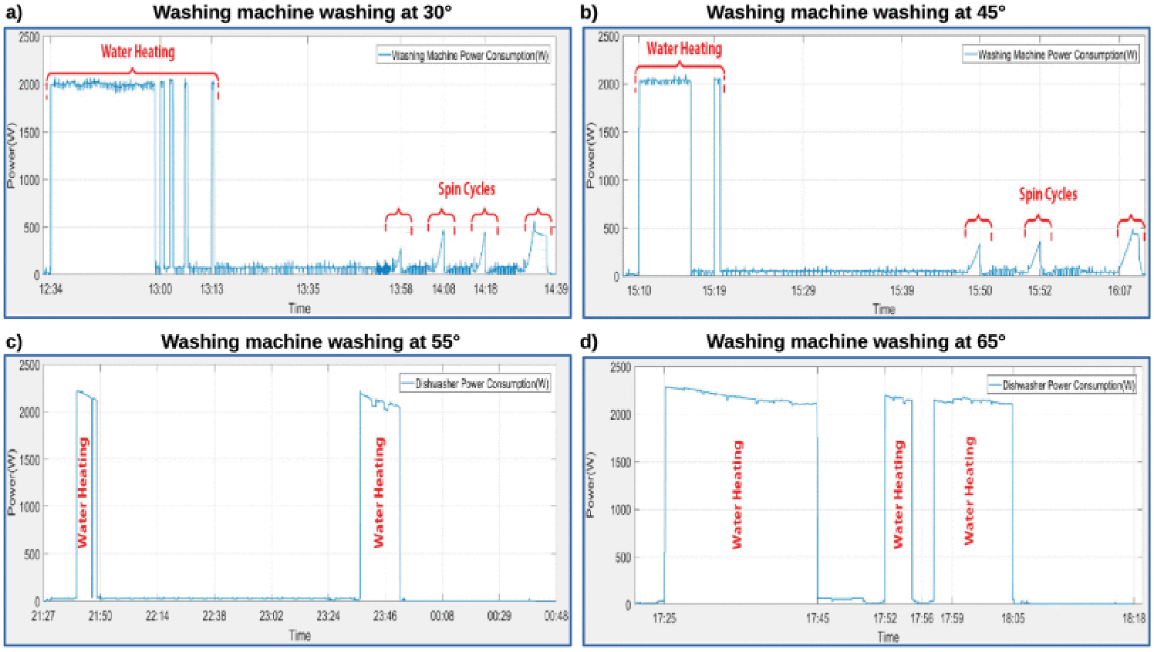


Figure 5 - Power consumption in Watts of different household appliances across time [78]

Detailed energy modelling captures the complexity of household electricity usage more effectively by considering the fluctuating operation patterns of various devices. Such models are critical for providing an accurate and comprehensive understanding of how residential electricity is consumed, especially as they incorporate insights often missed by less granular approaches. Energy modelling methods in residential settings can generally be categorised by their level of detail into three tiers: low resolution, medium resolution, and high-resolution models [78].

Low resolution models, often based on data with intervals exceeding fifteen minutes, were primarily developed and are designed for broad analyses. These approaches are well-suited for examining large-scale trends, such as regional or RECs consumption patterns, or assessing general household electricity behaviours under varying pricing strategies [78].

Medium resolution models, rely on data sampled at intervals between one minute and fifteen minutes. These models aim to bridge the gap by focusing on individual household energy patterns with moderate granularity, enabling a more nuanced understanding compared to their coarse-grained counterparts [78].

High resolution models, meanwhile, operate with data captured at intervals shorter than one minute, often down to a few seconds. Such models, which became more prevalent after 2010, delve deeply into the energy usage of specific appliances and systems within a household. Despite the potential of even finer measurements, on the millisecond scale, such resolutions remain largely unexplored, with only rare attempts to utilise sampling rates under one second, due to performance issues to process such a magnitude of data [78].

This dynamic variability is also evident in renewable energy systems. PV panels, for example, are highly sensitive to environmental factors like cloud cover and shading, which cause rapid fluctuations in energy output.

Figure 6, which seems to be using a fine grained illustrates this behaviour, particularly on a cloudy day (13-Jul-20), where frequent changes in sunlight result in sharp variations in power generation. The fine-grained data presented here highlights the significant impact of weather conditions on PV performance.

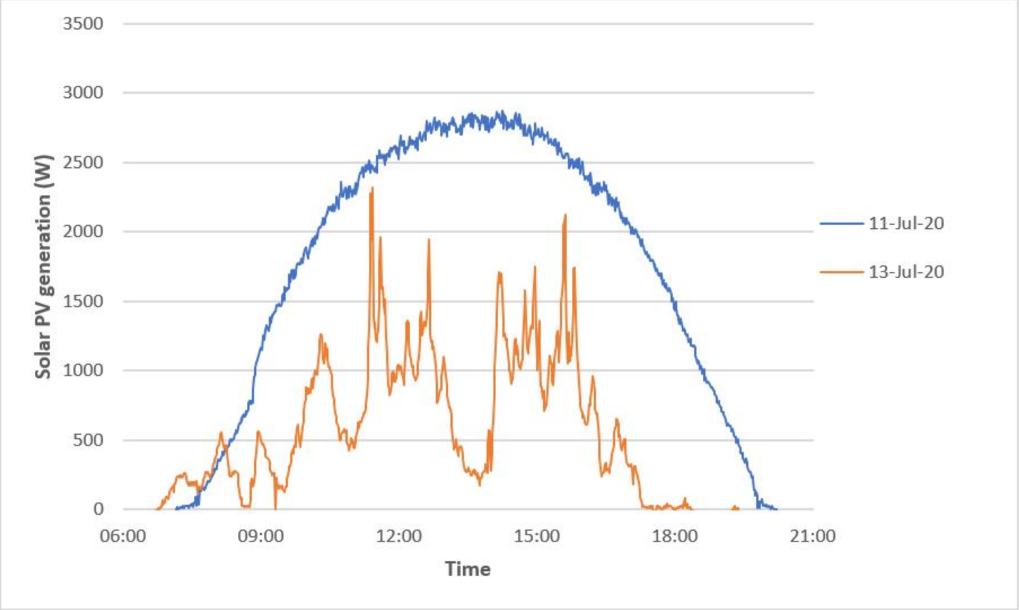


Figure 6 - Generation of power in Watts of a PV panel, during two selected days [79]

In contrast, Figure 7 shows data collected at a coarser resolution, with measurements taken every 30 minutes. This aggregation significantly smoothens the curve, obscuring critical insights into system behaviour. Despite being derived from a different scenario, it underscores the limitations of low-resolution data in capturing the intricacies of energy generation and consumption.

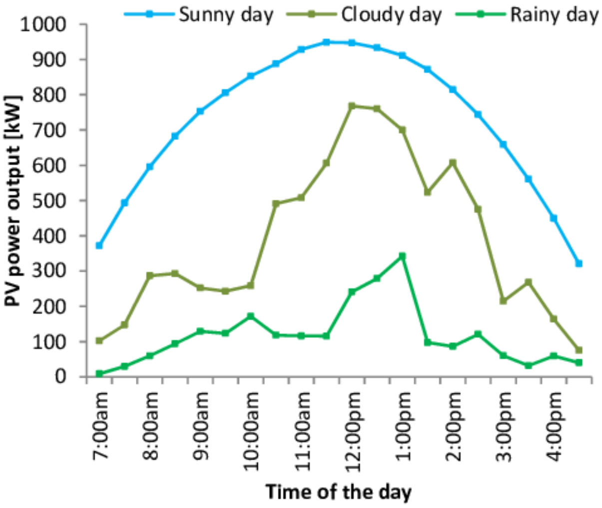


Figure 7 - Generation of power in Watts of a PV panel under different weather conditions [80]

High-resolution simulations (e.g., minute-by-minute intervals) are essential to accurately capture these fluctuations and optimise Maximum Power Point Tracking (MPPT) strategies, that is strategies to maximise energy extraction as conditions vary [81]. Similarly, battery systems, which depend on precise charge and discharge cycles for effective energy storage and delivery, rely on high-resolution data to accurately estimate efficiency, lifespan, and grid contributions. Battery systems also depend on detailed temporal data to model charge-discharge cycles accurately, influencing efficiency, lifespan, and grid services. Coarser data can misestimate performance, impairing demand-response and energy arbitrage strategies [82].

Demand-side management and dynamic pricing benefit from minute-level granularity, enabling real-time consumption adjustments. Coarser data risks oversimplifying fluctuations, weakening these strategies' effectiveness [82]. However, fine-grained simulations over long periods demand significant computational resources, requiring trade-offs between granularity and practicality [83].

The concept of aggregation in energy system modelling aims to balance computational efficiency with the need for maintaining accuracy. This approach involves dynamically adjusting the temporal granularity of the data, as highlighted in recent studies. During periods of rapid changes, such as peak demand or ramp-up phases, high-resolution modelling ensures accuracy by capturing critical dynamics. Conversely, during stable periods, coarser granularity reduces computational overhead, making the process more efficient without significantly compromising precision [84], [85], [86].

One prominent method of aggregation is clustering, which involves grouping similar time periods or selecting representative days. This technique reduces the volume of data processed while preserving the essential dynamics of the energy system. Such methods have been widely recognised for their ability to strike a balance between model fidelity and computational complexity. For instance, time-series averaging may suffice for long-term strategic planning, while optimisation-based clustering is more appropriate for short-term, detailed analyses. These distinctions are crucial for adapting aggregation strategies to specific objectives, whether for operational simulations or strategic energy system planning [85], [86].

Trade-offs in aggregation methods depend largely on the modelling goals and the system's specific needs. Sophisticated approaches, such as dynamic time-series clustering, enable models to focus computational resources on periods of critical system activity, while simpler methods remain effective for less granular needs. By implementing these strategies, models become computationally feasible without losing their ability to deliver insights during key system events, as demonstrated in recent reviews of aggregation techniques [84], [85], [86].

In summary, temporal granularity fundamentally shapes simulation accuracy and utility. Minute-level resolution provides detailed insights at higher computational cost, while intermediate resolutions balance detail and efficiency. Aggregation strategies further optimise this trade-off. Advances in machine learning and spatio-temporal modelling promise to enhance these capabilities, ensuring simulations remain accurate and computationally feasible.

Building on this understanding of temporal granularity, the next section addresses representing diverse energy assets within scalable DT platforms. These assets, including storage and household appliances, are crucial in capturing the dynamic behaviours of real-world energy systems.

2.7.3. Research Question 3

This section explores how diverse assets, including energy storage systems, EVs, and household appliances (e.g., refrigerators, dishwashers), can be effectively modelled within a scalable DT platform to emulate real-world energy behaviours. Similarly to Section 2.7.1, a systematic review done following the PRISMA framework. As before, details of the PRISMA flowchart, search strategy, eligibility criteria for exclusion/inclusion, and the review process itself are accounted for in Appendix B.

The increasing integration of energy storage systems, EVs, and household appliances into smart grids has raised many demands concerning sophisticated modelling frameworks able to emulate real energy behaviours. Recent studies have developed methodologies where the importance of statistical and probabilistic approaches is underlined in modelling energy consumption, focusing on household appliances and their integration into grid-interactive systems.

The generation of synthetic appliance-level data has proven particularly valuable in addressing challenges such as the scarcity of real-world data and the complexity of accurately representing appliance behaviours. Machine learning models, including Multilayer Perceptrons (MLPs) [87], Long Short-Term Memory (LSTM) networks [88], and Generative Adversarial Networks (GANs) [89], have been employed to synthesise realistic power consumption patterns. These methods enable the creation of datasets that reflect the diversity and variability of appliances in a typical household, essential for training predictive models and optimising energy systems.

To model appliances within a DT framework, one study suggested the adoption of a bottom-up approach [90]. This involves simulating individual appliance-level power consumption and aggregating it to represent household or community-level energy use. The whole process starts with classifying appliances based on their consumption pattern; for example, single-use appliances like a kettle or toaster will show repetitive patterns, while a washing machine or a dishwasher will have variation in their programmed cycles of operation [90]. Periodical appliances, such as refrigerators, have predictable cyclic behaviours, and others, like televisions or computers, are usage-dependent [90].

Thereafter, to model a device, there needs to be sufficient energy data collected in the real world, such as, for example, smart plugs [90]. However, the data collected from smart plugs can be, misleading due to differences in resolution and slow or faulty logging [90]. This paper suggests the usage of a few databases such as GeLaP [91], REFIT [92] or ENERTALK [93], which provide ground truth data and are publicly available. However, the real data collected is limited, so duplications may occur if this data is used as a basis to simulate.

Once the appliance models are established, integrating them into a DT requires a framework for orchestration. This orchestration can use predictive models to determine when appliances are activated and their corresponding energy usage. For instance, an activity prediction model, leveraging LSTM networks, can forecast appliance operation based on historical usage data [93]. This approach allows the DT to simulate not only the consumption of individual appliances but also their interactions and aggregate effects on the energy system.

Within RECs, these appliance models play a critical role in optimising energy flows. The DT can simulate scenarios such as peak demand management, renewable energy integration, and energy sharing among households. By modelling appliances at this level of granularity, it becomes possible to test and implement strategies that align consumption patterns with renewable generation, reduce wastage, and support demand-side flexibility.

Additionally, one of the papers [94] used statistical modelling of household appliances' power consumption which employed statistical distributions, such as Normal, Exponential, and Weibull distributions, validated through Kolmogorov-Smirnov and Pearson’s chi-squared tests [94]. By generating random values and comparing them with real-world averages, the study achieved low error margins (e.g., <4% for refrigerators and lighting), underscoring the precision of their models in predicting energy usage [94]. However, the framework assumes homogeneity across households with similar demographics and climatic conditions, limiting scalability across diverse settings. Despite that, this paper still gives key takeaways. Their approach emphasised the utility of statistical distributions to overcome data limitations and simulate extended operational scenarios, demonstrating the relevance of statistical modelling in DT platforms.

Another paper [95] extended this conversation by focusing on the demand-response potential of residential appliances within grid-interactive buildings. Using a bottom-up modelling approach, they leveraged disaggregated high-frequency data from over 500 homes to capture appliance-level energy usage and flexibility potentials [95]. Their analysis highlighted the importance of integrating occupant-driven usage patterns and the temporal variability of energy consumption [95]. For instance, clothes dryers were identified as the most impactful in demand-response scenarios, offering substantial potential for load modulation during peak hours [95]. Additionally, their study introduced the concept of aggregating individual household data to inform grid-level strategies, demonstrating how small-scale modelling can influence macro-level energy optimisation [95].

The previous two studies [94], [95] underscore the necessity of tailored models that balance detailed appliance-level insights with scalability. The first paper emphasis on statistical distribution aligns with the second one’s granular appliance modelling, offering complementary approaches. The former's statistical generalization ensures flexibility, while the latter's focus on high-resolution data provides actionable insights for demand-response planning. However, these methodologies face challenges in data integration and the computational demands of scaling to complex DT environments.

Modelling assets like storage systems, EVs, and appliances in DT platforms requires managing granularity and scalability, while addressing data quality and computational constraints. Combining statistical and machine learning approaches enables realistic simulations essential for optimising renewable integration and grid efficiency. Appliance-specific patterns and occupant behaviours are key to building accurate, actionable models for smart energy systems.

The next section builds on this foundation, shifting to a more granular and personalised dimension: integrating individual building objectives, such as cost reduction or self-consumption, into simulation frameworks. These objectives introduce complexity but enhance the adaptability and effectiveness of energy management strategies in RECs.

2.7.4. Research Question 4

This section examines how simulation frameworks can incorporate individual building goals, such as cost reduction or self-consumption maximization, and their impact on overall energy management within RECs. REC simulation is evolving from community-wide algorithms to building-specific systems, driven by modular design and decentralised decision-making. Centralised approaches, while effective for high-level optimisation, often overlook the diversity of building energy profiles and unique objectives.

Modularity is key to building adaptable, scalable frameworks that accommodate diverse algorithms suited to each building's characteristics. For instance, buildings with dynamic PV generation benefit from RL models that adapt to variability, while more stable profiles may rely on rule-based or hybrid approaches for efficiency. Modular RL architectures also support reassignment and fine-tuning as needs evolve. Decentralised designs, such as multi-agent systems (MAS), add flexibility and resilience by treating each building as an autonomous agent that optimises locally while cooperating to meet REC-wide goals [96]. However, MAS frameworks are not without limitations, especially in scenarios requiring advanced coordination.

MARL builds upon the foundation of MAS, offering advanced capabilities for conflict resolution and equitable resource allocation among agents [97]. The collaborative decision-making processes embedded within MARL are vital to preventing suboptimal outcomes, such as overloading shared energy assets or neglecting the contributions of specific buildings. These frameworks not only optimise operational performance but also enhance fairness, ensuring a balanced distribution of energy resources across decentralised systems.

To realise the full potential of MARL, various architectural approaches have been developed, each addressing specific challenges in multi-agent systems. Independent Learning, Centralised Critic, and Value Decomposition are three prominent frameworks [98], each offering unique strengths and trade-offs. These architectures differ in their handling of information flow, policy updates, and coordination among agents, as illustrated in Figure 8.

In the Independent Learning model (Figure 8a), each agent learns autonomously using only local observations. Its Q-function and actor-critic framework operate independently, simplifying learning and reducing computational load [98]. This approach works effectively in scenarios where the actions of individual agents do not conflict. However, its limitations become evident in complex systems requiring high levels of coordination. Two significant challenges emerge in this context. The first is the non-stationary environment, where the policies of interacting agents constantly evolve, complicating the convergence of learning [98]. The second is the difficulty in assigning responsibility for specific actions when multiple agents act simultaneously, which hampers task coordination and leads to inefficiencies [98].

To address these limitations, the Centralised Critic architecture introduces a shared evaluation mechanism that considers the collective actions of all agents [98], as shown in Figure 8b. By centralizing the evaluation process while maintaining decentralised decision-making, this approach mitigates non-stationarity and fosters better alignment of individual policies with overarching system goals [98]. However, the reliance on shared information introduces higher computational and communication demands, making this approach less practical for highly distributed systems [98].

A more balanced solution emerges in the form of Value Decomposition [98], illustrated in Figure 8c. This hybrid framework decomposes the global value function into individual contributions, enabling agents to collaborate effectively while retaining a degree of local autonomy [98]. Value decomposition not only facilitates coordination but also ensures that agents' individual decision-making processes are aligned with the system's shared objectives [98]. This method addresses the inherent coordination challenges in decentralised systems, fostering a shared understanding among agents to achieve optimised collective outcomes.

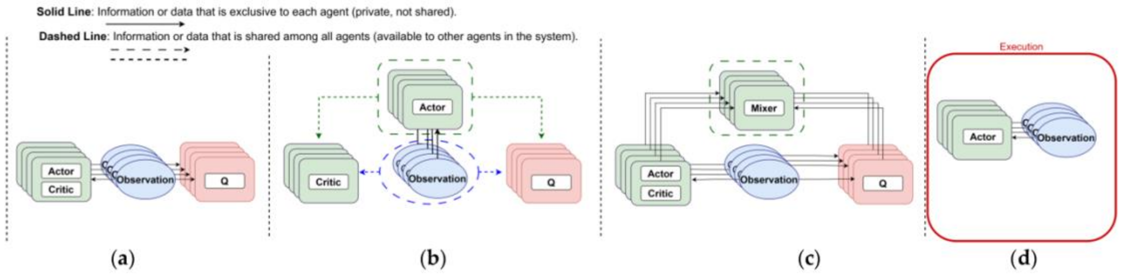


Figure 8 - Architectural models used in MARL frameworks [98]

The scalability of this approach is further enhanced by hierarchical optimisation frameworks that link building-level autonomy with community-wide oversight [99]. While buildings pursue their individual objectives, a central or distributed controller aggregates and balances these efforts to ensure collective performance metrics, such as renewable energy utilization and peak load reduction, are met [99]. This dual-layered architecture preserves the integrity of both individual and collective energy management strategies, ensuring they remain aligned even as conditions or priorities shift.

Building on this discussion of hierarchical frameworks, the next section explores the integration of real-world data into DT simulations. Real-time and localised data, such as weather and energy consumption patterns, are critical to bridging the gap between theoretical models and real-world applications. These inputs significantly enhance the realism and predictive accuracy of simulations, enabling more effective management of RECs.

2.8. Chapter conclusions

The state of the art, building upon the research questions presented on Section 1.3.2, demonstrates the transformative potential of renewable energies, smart grids, and DT technologies in advancing RECs. By integrating modular and scalable frameworks, enhancing data interoperability, and adopting high-resolution simulations, RECs can achieve optimised energy management and sustainable practices. Despite significant progress, challenges remain in standardizing frameworks, ensuring scalability, and balancing granularity with computational efficiency. Addressing these gaps is crucial to realizing the full potential of RECs as decentralised, participatory energy systems that align with global sustainability goals.

The following section goes into the planning and the activities laid out for this. It reinforces the research methodology, project timeline, and key deliverables, focusing on the practical application of the insights gained from the state-of-the-art review.

3. Planning

As relevant literature states, careful planning means identifying how to achieve the objectives of the research within a stated scope, timeframe, using available resources. This chapter defines an overall Work Breakdown Structure, or WBS [100] of the project.

The WBS aligns closely with the Design and Creation methodology, providing a structured approach to developing, implementing, and validating a DT simulation framework for RECs. This ensures that the scope is clearly defined, with tasks appropriately distributed across each phase to address deliverables and dependencies. The iterative nature of the methodology is reflected in feedback loops and validation mechanisms embedded throughout the lifecycle.

The planning process identifies the essential skills needed for project completion, incorporates tasks to develop and apply these skills, and accounts for task dependencies and milestone tracking. Additionally, risks are identified, analysed, and mitigated to ensure smooth project execution. Cost estimation and monitoring/control procedures are also integrated where applicable to reflect the project's operational needs.

3.1. Alignment with Scope and Research Methodology

The WBS, as seen in Figure 9, effectively integrates the project scope with the Design and Creation methodology, focusing on iterative development, interdisciplinary insights, and continuous validation. Each phase contributes to the research objectives and ensures that deliverables are achieved systematically. The first phase, Context, establishes the foundation by reviewing the research gaps which were identified in Section 1.1.3 and further analysed and iterated in Section 2. This phase connects the state of the art to practical needs.

The second phase, Acquire and Improve Skills, focuses on developing negotiation and presenting skills, identified as weak through self-assessment and peer feedback in Appendix C. These skills will be applied in engaging with stakeholders, presenting findings, and managing tasks. As noted in Section 3.2, this phase spans most of the project, ensuring continuous improvement. This ongoing effort supports effective collaboration, leadership, and alignment with project objectives.

The third phase, Analysis and Design, takes the foundational understanding from the first phase and turns it into practical steps for development. This stage involves carefully examining the needs of stakeholders, reviewing existing real-world datasets, and, if needed, creating synthetic data to fill in any gaps. Following this, the design phase focuses on refining existing features and developing new ones that will directly guide the creation of the DT framework.

The Implementation phase translates the design into reality. It involves the development of the framework's core functionalities, followed by iterative testing cycles to ensure reliability and performance.

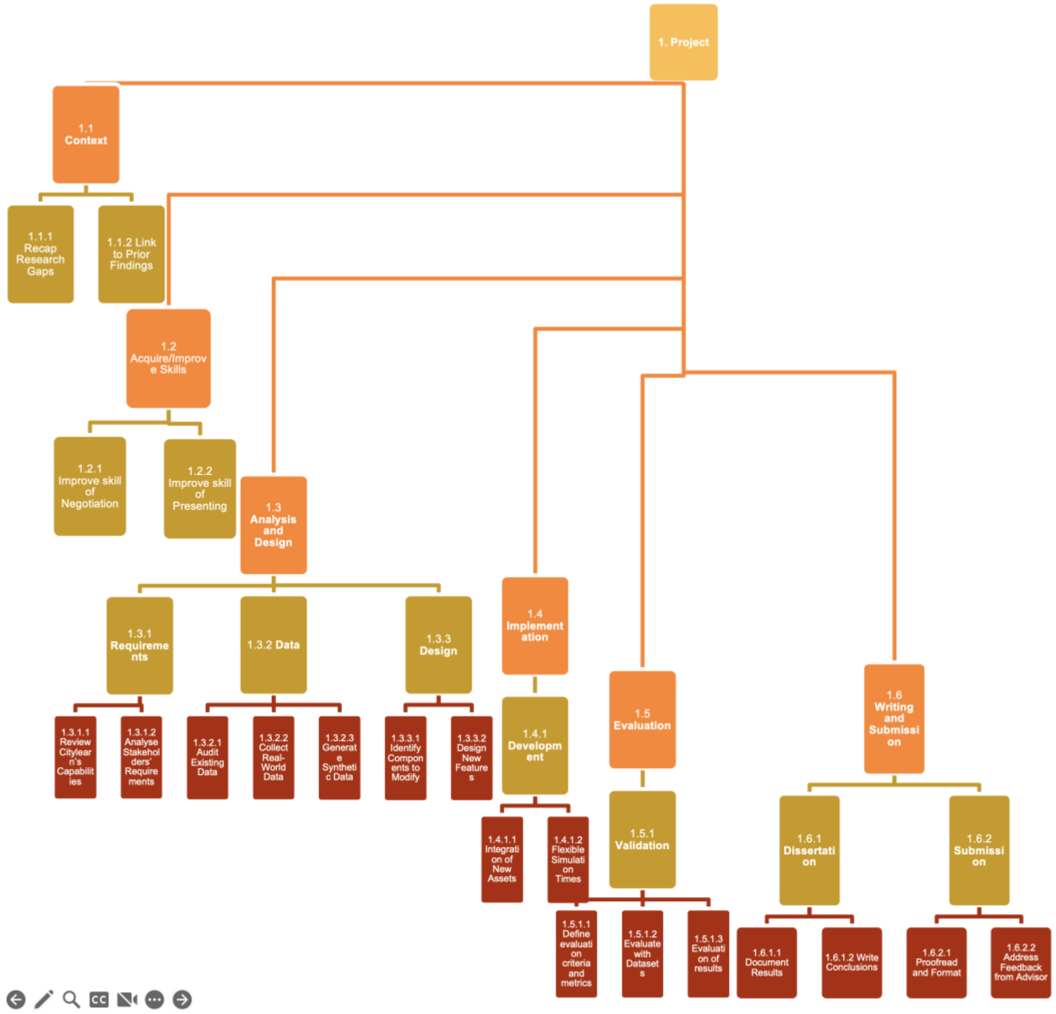


Figure 9 - Work Breakdown Structure of the project

The fifth phase, Evaluation, benchmarks the framework against real-world datasets and stakeholder expectations, validating that the results meet the project's objectives and address the identified challenges. The final phase, Writing and Submission, involves documenting the results, drawing well-supported conclusions, and refining the work through meticulous proofreading and incorporation of advisor feedback. Through this structure, the WBS ensures that tasks are clearly defined, iterative feedback is incorporated, and the scope remains adaptable to new discoveries, all while maintaining alignment with the chosen methodology.

3.2. Project Schedule

The project schedule has been carefully thought to establish realistic timelines while retaining flexibility to accommodate any necessary adjustments should the plan deviate. Created using Microsoft Project, it outlines clear start and end dates for each task alongside detailed duration estimates, as seen in Appendix C and D. Milestones mark critical points in the project lifecycle, such as the completion of phases or delivery of major outputs. These milestones act as checkpoints for evaluating progress and monitoring the project's status. Instead of uncertainty about the current state of the project, the schedule offers a clear overview, enabling efficient tracking of progress and identification of areas needing attention.

Dependencies between tasks have been analysed and mapped, ensuring that tasks follow a logical sequence. This structure allows delays to be anticipated and mitigated effectively. While the schedule is designed to maintain flexibility, it ensures steady progress toward the project's goals. Additionally, the improvement of skills, first introduced in Section 3.2, is a continuous process that will take place throughout the entire project. This emphasises that skills can always be further enhanced, contributing to the project's success.

3.3. Cost Estimation

While the project requires minimal direct funding, resource management remains important. A key potential expense is attending academic conferences to present results, which supports visibility, collaboration, and knowledge exchange. These events typically incur registration, travel, and accommodation costs and should be included in the budget.

Beyond direct costs, the project demands significant indirect resources. The time and effort of advisors and colleagues represent an opportunity cost, as their support could have been allocated elsewhere. Acknowledging these commitments highlights the need for careful planning and justification of the project's scope and value.

3.4. Monitoring and Control Procedures

As previously mentioned in Section 3.2, the project will be actively monitored using Microsoft Project, ensuring consistent tracking of progress and alignment with the planned timeline. A baseline has been established to provide a reference point, allowing for easy comparison between the planned and actual progress. Key milestones, deliverables, and deadlines will be regularly updated throughout the project lifecycle to maintain focus and ensure timely completion of critical tasks.

The progress that is being made, challenges, and any necessary adjustment of this plan will be reviewed in weekly meetings. Such meetings allow for the tracking of completed tasks, identification of any delays or problems, and discussions on how best to find solutions. They will also serve as a forum through which feedback from the advisors and colleagues, not to mention the occasional input from the stakeholders, can be received to ensure the project is on course and that any benefits from their experience are availed. Updates will be shared regularly, and where relevant, changes in the schedule or deliverables will be affected through discussions. This proactive approach to monitoring and communication will help ensure the project's success and timely completion.

Furthermore, periodic audits will be conducted to evaluate adherence to the ethical plan delineated, ensuring processes remain efficient and aligned with objectives. Any discrepancies or risks identified during these audits will prompt immediate corrective action. Regular updates will be shared with the team, and adjustments to schedules, resources, or deliverables will be made as necessary.

3.5. Risk Management (including assumptions and limitations)

Risk management has been conducted proactively to enhance the project's resilience against unexpected challenges. Risks have been identified and documented in a detailed risk register, considering both internal and external factors that may affect the schedule and deliverables. Each risk has been analysed for likelihood and impact, enabling prioritization and the development of effective mitigation strategies to reduce their probability and minimise consequences.

In addition to conventional risks, assumptions and limitations regarding the Digital Twin's development are acknowledged, such as the accuracy of input data (e.g., energy consumption profiles, grid characteristics) and the reliability of current technologies, sensors, and smart meters for real-time simulations. The ongoing shift toward smart energy communities and renewable energy underscores the need for advanced simulation tools.

However, limitations must be addressed, including significant computational requirements that may restrict simulation scope and data granularity due to hardware/software constraints. The effectiveness of the Digital Twin depends on high-quality, real-time data, which may be undermined by poor data integrity. High development and implementation costs could hinder large-scale prototyping, and scalability across diverse grid types, particularly in metropolitan or rural areas, may demand extensive adaptation.

To mitigate these issues, proactive adjustments will be made throughout the project to keep critical tasks on track. The risk register (Appendix E) outlines key risks, including computational constraints, data inadequacies, scalability challenges, and ethical concerns such as data privacy. Additional risks, such as stakeholder misalignment, knowledge gaps, and budget limitations, are also addressed. This register is a living document, regularly updated and reviewed in team meetings to keep the project adaptable and responsive to emerging risks.

3.6. Conclusions taken from the chapter

The planning framework outlined in this section provides a structured and adaptable roadmap to achieve the project's objectives. By integrating a comprehensive WBS, realistic scheduling, cost estimation, and proactive risk management, the project is designed to navigate complexities effectively while remaining aligned with the Design and Creation methodology. The emphasis on continuous monitoring, skill development, and iterative validation ensures flexibility and resilience, enabling the project to respond to challenges while maintaining focus on deliverables. Together, these elements form a cohesive strategy that will have influence on the successful execution and timely completion of the project.

4. Analysis and Design of the Proposed Enhanced Simulation Platform

The architecture of the proposed simulation platform is engineered to bridge the gap between theoretical REC models and operational DTs, considering the requirements introduced in Section 1.3.1. By extending CityLearn, a well-established environment for multi-agent energy system simulations, this chapter introduces a comprehensive methodology that integrates several advanced capabilities:

Central to these improvements is the **Improved Temporal Granularity**, which refines the temporal resolution to allow for a more detailed yet computationally efficient simulation of REC operations. As a direct consequence of this refinement, the **Adaptation of Existing Models for Temporal Granularity** becomes necessary, requiring modifications to asset models so they can support multiple time resolutions without compromising accuracy or performance.

The platform also introduces **Modular Asset Integration**, extending CityLearn to simulate a broader array of household appliances such as refrigerators and washing machines. This not only increases modelling fidelity but also brings the platform closer to functioning as a digital twin. Complementing this is the **Incorporation of Stochasticity**, which introduces random variations in energy consumption and generation. This addition improves the simulation's realism and enhances its capacity to evaluate strategies under real-world variability.

To support interoperability and post-simulation analysis, the platform includes **Data Export Functionality**, allowing simulation outputs and agent performance metrics to be exported in standardised formats like CSV and JSON. Another key development is the **Refined EV Modelling**, which transitions from a charger-centric view to a household-centric approach, more accurately capturing the intermittent presence of electric vehicles in residential settings and better reflecting actual usage patterns.

The architectural design elucidated in this section also ensures **forward compatibility**, enabling scalability and experimentation with novel REC strategies. This is achieved through modular, configurable, and standards-aligned design principles.

This chapter outlines the platform’s architecture through the following structure:

1. **Section 4.1** – Introduces the FURPS+ framework, which categorises and details the functional and non-functional requirements guiding the platform's design;
2. **Section 4.2** – Positions the enhanced simulation platform within the broader OPEVA project, using the C4+1 architectural model to illustrate its role and integration;
3. **Section 4.3** – Describes the core architectural extensions to the CityLearn environment, including multi-resolution support, stochastic modelling, and enhanced modularity;

The architectural design is informed by principles from computational sustainability and aims to provide a forward-compatible environment for simulating and evaluating energy community strategies at scale.

4.1. FURPS+

The FURPS+ model categorises non-functional requirements into Functionality, Usability, Reliability, Performance, Supportability, and additional constraints (e.g., design, legal). This section details these in the context of the thesis goal to extend the CityLearn-based simulator.

4.1.1 Functionality (F)

Functionality requirements define the core capabilities of the platform, focusing on its ability to model, simulate, and analyse decentralised energy systems realistically and flexibly, as they were previously introduced in Sections 1.1.5 and 1.3.1.

- **F1:** The platform shall support multiple temporal resolutions (e.g., 1-minute, 15-minute, hourly), enabling researchers to control simulation fidelity depending on their study objectives.
- **F2:** Asset models shall dynamically adapt to different temporal granularities, ensuring consistent behaviour and performance across simulation resolutions.
- **F3:** The simulator shall support modular integration of household assets (e.g., refrigerators, washing machines), allowing users to include or exclude components based on their research needs.
- **F4:** The system shall enable stochastic modelling of energy consumption, production, and exogenous variables (e.g., weather, occupancy), supporting realistic scenario variability.
- **F5:** The simulation must allow the configuration of energy management objectives (e.g., self-consumption maximization, peak shaving) at the building level, reflected in agent reward functions and decision logic.
- **F6:** The platform shall implement a household-centric EV modelling approach, supporting intermittent EV presence and realistic load profiles aligned with residential patterns.
- **F7:** Simulation outputs and agent performance metrics shall be exportable in dashboard-compatible and analysis-ready formats (e.g., JSON, CSV), supporting downstream processing in external tools.

4.1.2 Usability (U)

Usability requirements ensure the platform is accessible, learnable, and usable by researchers and practitioners with varying degrees of technical expertise.

- **U1:** Simulation configurations shall be defined using structured and human-readable JSON files, enabling quick scenario creation without modifying source code.
- **U2:** Logging and output metrics (e.g., energy consumption, costs, emissions) shall be well-structured and organised to support post-simulation visualisation and analysis.
- **U3:** The codebase and documentation must follow clean, modular conventions with intuitive naming and consistent structure, lowering the barrier for new contributors.
- **U4:** A command-line interface (CLI) or script-based launcher shall be provided to run simulations without requiring direct interaction with internal APIs.

4.1.3 Reliability (R)

Reliability requirements govern system stability, fault tolerance, and deterministic behaviour.

- **R1:** The environment shall produce deterministic outputs when stochasticity is disabled.
- **R2:** The system shall implement error handling for invalid/missing inputs (e.g., configuration files).
- **R3:** Agents shall complete simulations without crashes due to invalid actions or states.
- **R4:** All modifications shall maintain the output quality of the original system.

4.1.4 Performance (P)

This category refers to system efficiency in running simulations, training agents, and processing data.

- **P1:** The simulation shall complete within 10 minutes for small-scale cases (e.g., ≤5 buildings and < 2000 timesteps).
- **P2:** The system shall execute without memory leaks or crashes during medium-duration simulations.

4.1.5 Supportability (S)

These requirements address ease of maintenance, extensibility, and integration with other tools.

- **S1:** The project shall include documentation for setup, configuration, and API usage.
- **S2:** The system shall allow adding new buildings/agent types with minimal core modifications.
- **S3:** Unit/integration tests shall cover core components (e.g., reward functions, data loading).
- **S4:** The system shall be compatible with TensorFlow, PyTorch, and stable-baselines3.

4.1.6 Plus (+)

This section includes additional constraints, such as architectural decisions, design limitations, and technology dependencies.

- **+1:** The solution shall use CityLearn as the base simulation environment.
- **+2:** Implementation shall use Python 3.9+ with pip dependency management.
- **+3:** Modifications shall adhere to OpenAI Gym interface standards.
- **+4:** Research metrics (e.g., peak demand reduction) shall be exportable to CSV/JSON.

With the functional and non-functional requirements now defined, the next step is to situate the enhanced simulation platform within the broader system context. Understanding how these requirements translate into architectural roles and interactions is essential to assess the platform’s contribution to the OPEVA project and its integration within the overall digital ecosystem.

4.2. Positioning of the Platform within the OPEVA Project

To fully understand the simulator's contribution, it is important to contextualise how CityLearn fits within the larger OPEVA ecosystem, as previously introduced in Section 1.1. This section clarifies the proposed solution's role and how it integrates into the project's overarching architecture, following the C4+1 Model [101].

Furthermore, the architectural positioning of CityLearn allows for the integration with adjacent systems via APIs or message queues (e.g., MQTT, Kafka), allowing simulation environments to incorporate real-time external inputs such as grid tariffs, carbon pricing, or user preferences from demand response platforms.

In the context of OPEVA, this architectural flexibility enables CityLearn to serve as an early-stage digital twin for virtualised energy communities, offering a testbed for coordinated control schemes, decentralised market mechanisms, and user-centric optimisation approaches. Its extensibility further supports experimentation with new device types, forecasting models, or control paradigms, aligning the platform with the iterative, experimental nature of applied energy research.

To align the components handled in this dissertation, the Level 1 Logical View of the C4+1 Model is introduced in Figure 10. Additionally, the subsequent diagrams in Figure 11 and Figure 12 situate the developed components within the overall system design.

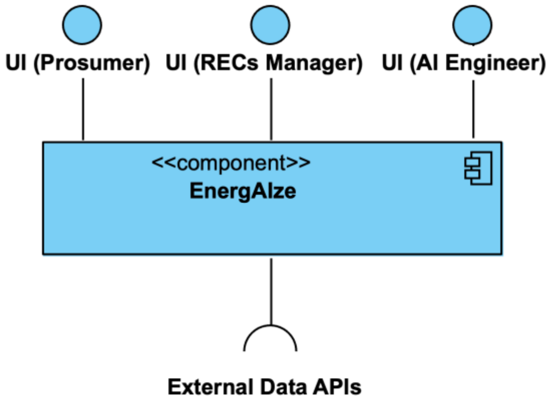


Figure 10 – Logical Level 1 View of the EnergAIze System

EnergAIze, as shown in the aforementioned figures, is the system mentioned previously, developed by the OPEVA project, to coordinate the energy use of individual installations, which can be grouped into RECs , by utilizing an AI-driven model [102]. This model operates at regular intervals, analysing real-time energy conditions during each cycle. To conduct this analysis, the system gathers data from multiple APIs, described in Figure 11, to reconstruct the energy usage profile of each installation at a specific point in time. The objectives defined in the scope of this thesis have been integrated into the EnergAIze system through the Simulator Component, where they are responsible for managing building-level energy coordination and demand response, based on input data from the other system components.

Below is a list of the interfaces provided by EnergAIze and their respective roles within the system.

- The **Prosumer User Interface (UI)** is a mobile application through which end-users can monitor household activity, select their optimisation preferences, add energy assets, and define flexibility settings.
- The **RECs Manager UI** is tailored for community managers. It enables administrative actions such as onboarding new members, removing inactive ones, creating communities, and assigning users to them.
- The **AI Engineer UI** supports technical users in tasks such as preparing datasets, training machine learning models, and deploying AI-based optimisation solutions for REC communities.

Logical View Level 2 (Figure 11) presents a detailed breakdown of the EnergAIze's components.

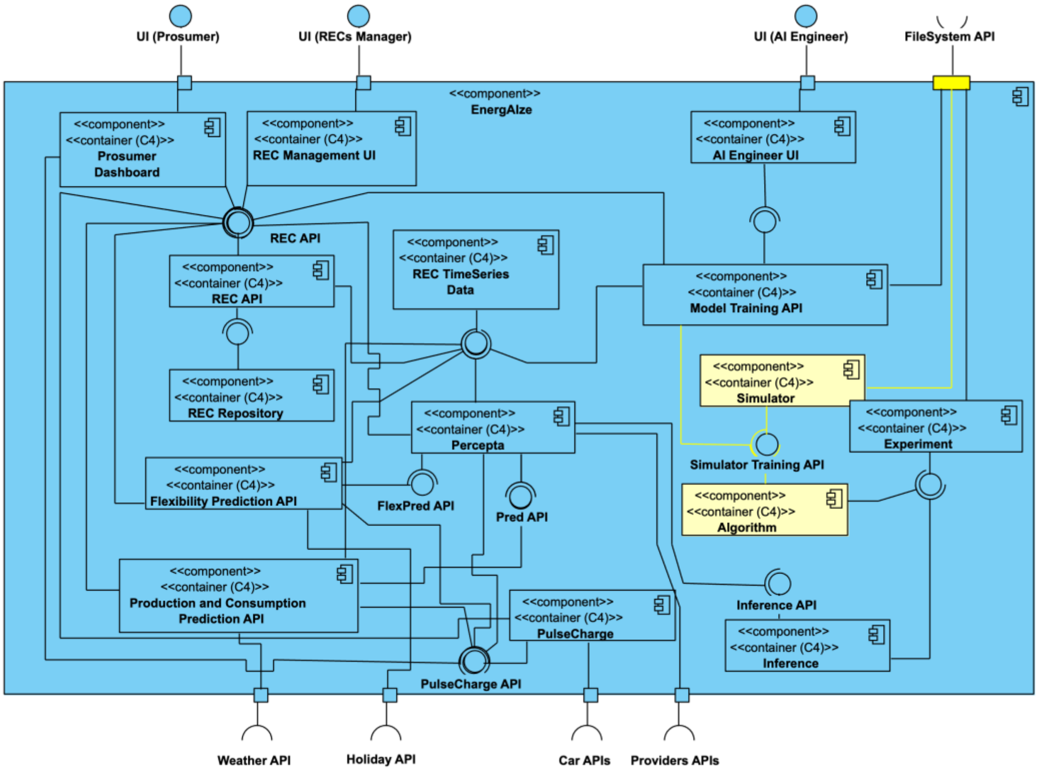


Figure 11 - Logical Level 2 View of the EnergAIze System

As previously mentioned, this dissertation focuses exclusively on the Simulator component, and by extension, the Algorithm module (both highlighted in yellow). This narrow scope highlights the scale and complexity of the broader Digital Twin system being constructed, of which only a small subset is currently addressed. The Simulator exposes the Simulator Training API, utilised by the Algorithm and the Model Training module, and relies on the FileSystem API, also highlighted in yellow, to manage essential resources such as CSV datasets, schema definitions, and simulation outputs.

To provide a clearer understanding of the deployment architecture within the OPEVA project, a physical view is presented on Figure 12. Centred on SoftCPS's cloud infrastructure, this architecture hosts key containerised services including MLFlow for experiment tracking, the Model Training API, and most notably, the Simulator, which executes RL algorithms crucial for energy optimisation.

A strong emphasis is placed on experiment reproducibility and lifecycle management. By leveraging MLFlow and version control tools, every training run, model artifact, and output dataset is logged and traceable, ensuring performance improvements can be audited and algorithmic behaviour remains transparent to researchers and stakeholders.

Supporting services such as the REC API, databases (MongoDB, MySQL), EV charging manager (PulseCharge), and message brokers (RabbitMQ) provide the data backbone and integration layer that facilitate cross-component communication. This architecture enables the Simulator to ingest up-to-date information, ranging from user preferences to device states and grid signals, and in turn, produce outputs that drive actionable decisions within the wider REC management platform.

As illustrated in Figure 12, this physical view positions the Simulator, highlighted in yellow, serves as the central engine within a modular and scalable DT infrastructure, empowering intelligent, adaptive control of RECs across the OPEVA ecosystem.

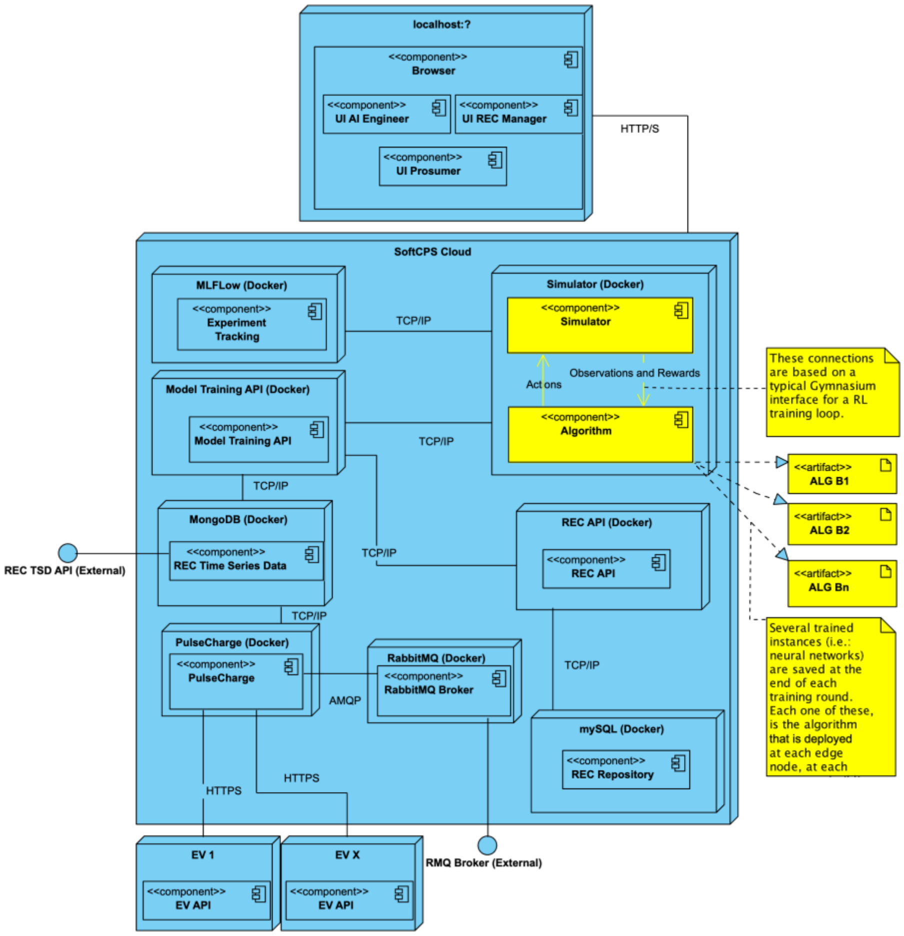


Figure 12 - Physical Level 2 View of the EnergAIze System

Having positioned the Simulator within the broader OPEVA architecture, Section 4.3 now turns to the core enhancements made to the CityLearn framework. These developments represent a shift from the current (AS-IS) implementation to a future-ready (TO-BE) architecture, focused on advancing modularity, scalability, and the ability to model and manage complex REC dynamics effectively.

4.3. Core Architecture Extensions to CityLearn

The transition from the current system (AS-IS) to the future state (TO-BE) highlights architectural improvements centred on modularity, scalability, and the enablement of intelligent energy community management.

The AS-IS domain model, shown in Figure 13, captures the fundamental relationships in the CityLearn environment. Each building operates within a many-to-one structure relative to the environment, and contains a portfolio of devices, including generation units (e.g., PV panels), storage systems (e.g., batteries, thermal tanks), and flexible consumption loads (e.g., heat pumps, electric water heaters, deferrable appliances). EVs are modelled as mobile storage units linked to chargers with time-varying availability profiles derived from empirical datasets.

Agents interface with these buildings by observing time-series data, ranging from device states and environmental variables to occupancy signals, and producing control actions based on defined reward functions. These may target specific objectives such as comfort, emissions reduction, or cost minimization.

Observations and actions are exchanged at each timestep, forming a closed feedback loop that continuously adapts to dynamic conditions. Unlocking higher fidelity in control actions and enabling the simulation of real-world phenomena like intraday solar variability or minute-level load ramping. This design not only increases the granularity of learning but also enhances the realism of policy training.

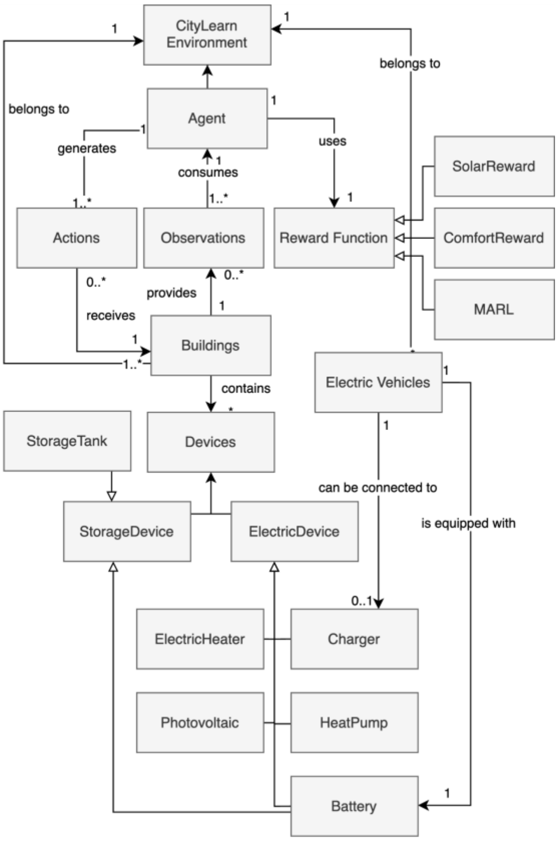


Figure 13 - Domain Model AS-IS of CityLearn (adapted from [103])

As discussed in Section 2.5, the CityLearn platform provides a foundation for simulating energy communities using RL, with a modular architecture that includes the Environment Engine, Agent Orchestrator, Reward Calculator, and Data Pipeline. While the original implementation supports cooperative control strategies and time-series simulation based on hourly timesteps, it lacks several capabilities required for more realistic and fine-grained energy community modelling.

This section builds upon that foundation by aligning the simulation design with the system requirements introduced in Section 1.3.1. These include support for sub-hourly control resolution (SO3, SO4), integration of additional household assets (SO5), multi-objective agent behaviour (SO6), and stochastic realism (SO7). The proposed enhancements are also evaluated through the lens of FURPS+, ensuring that functional features (e.g., multi-agent heterogeneity) are developed alongside critical non-functional attributes such as usability, configurability, modularity, and performance.

The configuration schema, CityLearn's central mechanism for defining simulation parameters, must be extended to enable these new features. These changes are crucial not only for fulfilling functional goals, but also for maintaining backward compatibility and supporting scalable experimentation, setting the stage for the architectural modifications that follow.

4.3.1. Extended Configuration Schema

Building upon the Data Pipeline, it is essential to explore how CityLearn ingests data and where this data originates from. At the heart of CityLearn's flexibility is its declarative configuration system, which will have to be significantly enhanced to support the requirements that have been enumerated on Section 1.3.1. The native configuration structure is organised into three main components: global settings, individual building descriptors, and time-series data inputs.

Global Configuration

The global schema file (schema.json) governs simulation-wide parameters, such as the number of timesteps per hour, the list of building descriptors, and the devices that each building contains. This relationship can be systematised as the Figure 14.

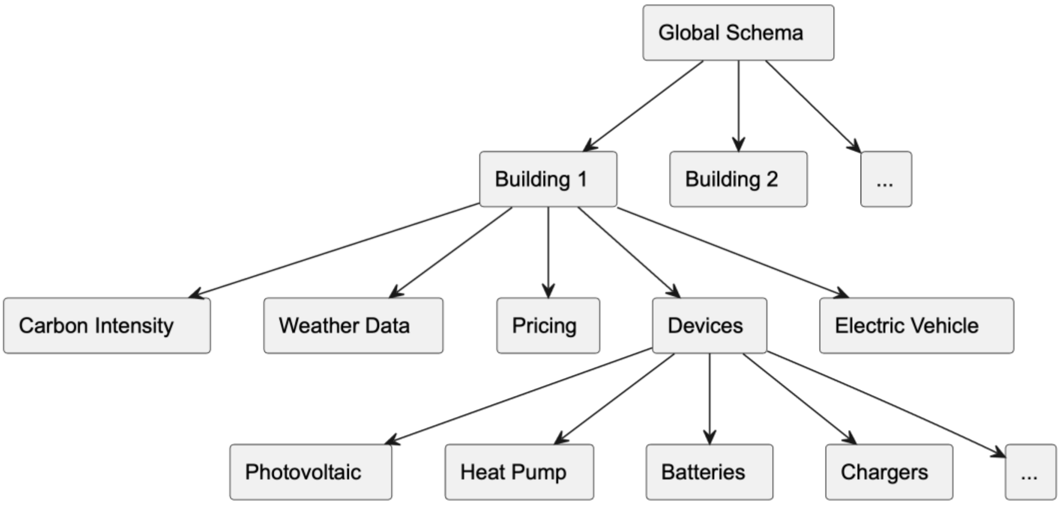


Figure 14 - Configuration Schema Hierarchy in CityLearn

The global configuration specifies simulation-wide attributes such as the number of seconds per time step, which has been extended to support sub-hourly intervals (e.g., 15-minute steps). This modification enables finer temporal resolution, allowing more granular control of HVAC systems, batteries, and load responses, crucial for modelling modern demand-side flexibility strategies. Additionally, the global configuration lists all buildings involved in the simulation and sets up the global reward structure, including custom or agent-specific metrics. Control-related settings, such as agent type, architecture (centralised vs. decentralised), and hyperparameters, are also specified here, allowing users to integrate custom controllers or RL agents.

Each building in the simulation is described by a dedicated set of data files. Building data files are CSVs containing time-series data such as end-use electrical loads, occupancy, solar generation, and indoor environmental conditions. These can be derived from detailed simulation tools like EnergyPlus or obtained from real-world building automation systems. Similarly, weather files provide outdoor environmental variables and are essential for contextualizing building thermal dynamics and for conditioning agent observations. It also contains carbon intensity and electricity pricing files, also in CSV format, that supply temporal profiles of CO_2 emissions and electricity rates, respectively, enabling the design of agents that optimise for cost or environmental impact. This dissertation's implementation will leverage these files to support heterogeneity in objectives across buildings, a departure from the cooperative paradigm of earlier CityLearn versions.

Crucially, all data files are referenced and structured through the schema.json file, which not only defines the simulation topology but also configures the observation and action spaces, device models, and dynamic behaviour of each building.

The extensions to the configuration schema directly will support several novel features, including sub-hourly simulation, building-specific reward functions, and the use of synthetic stochastic load profiles. These enhancements will have to be integrated into the schema in a backward-compatible manner, ensuring that existing workflows remain functional, like was referenced previously.

While a comprehensive listing of all data elements in the configuration files would be overly detailed for this context, Table 1 provides a focused overview of the most relevant schema extensions. It highlights the key features introduced, the corresponding schema elements responsible for their implementation, the specific system objectives from Section 1.3.1 that they support and the FURPS+ from Section 4.1 connected to them.

Table 1 - Key Configuration Schema Extensions and Their Alignment with System Objectives

Extension Feature	Description	Enabled By Schema Element	Linked System Objective	Linked FURPS+ Requirements
Improved Temporal Granularity	Allows simulation timestep granularity below 1 hour (e.g., 15-min intervals)	seconds_per_time_step	SO3, SO4	F1, P1, U2
Adaptation of Existing Models for Temporal Granularity	Adapts asset models for consistent performance across time resolutions.	-	SO4	F2 and S4

Extension Feature	Description	Enabled By Schema Element	Linked System Objective	Linked FURPS+ Requirements
Building-Specific Reward Functions	Enables heterogeneous agent goals (e.g., cost, emissions, comfort)	reward_function, reward_weights	SO6	F4, S2, U1
Modular Asset Integration	Custom device models (e.g., fridges, washing machines)	observation_variables, action_variables	SO5	F3, F2, S4
Incorporation of Stochasticity	Enables realistic, probabilistic modelling of uncertain behaviours	Is not enabled. Requires additional schema support	SO7	R1, F3, P2
Data Export Functionality	Exports simulation data and metrics in standard formats for easy integration and reproducibility.	–	SO2	F7
Refined EV Modelling	Shifts EV modelling to household-centric, capturing intermittent presence and realistic use.	EV-related schema files	SO6	F6

To address these limitations, a set of foundational architectural extensions is proposed, each aligned with the research objectives SO2–SO8. For each extension, the approach begins with a targeted analysis of the current system to identify structural or functional constraints. This is followed by a systematic evaluation of alternative design strategies, considering their relative merits in terms of flexibility, implementation complexity, and fidelity to real-world conditions. Based on this evaluation, a preferred solution is selected, and a corresponding system design is developed. The first, and most foundational, of these extensions is the introduction of **variable temporal granularity**.

4.3.2. Improved Temporal Granularity

AS-IS System Analysis

The current system architecture is limited to fixed hourly time steps across all simulations. Although a seconds_per_time_step parameter is defined in schema.json, it is not actively used in any time-dependent computations. This parameter is loaded in the environment module via schema['seconds_per_time_step'] and propagated to various components such as base.py, building.py, and data.py. However, its role remains largely passive, suggesting potential for further development.

A major limitation in the existing architecture is the lack of support for minute-level timestamps in the building-level input data. As outlined in Section 4.3.1, the simulation framework depends on schema.json, which includes temporal fields such as month, hour, and day_type (with values ranging from 1 to 7, representing weekdays). Crucially, there is no column for minutes, which restricts the system's temporal resolution to whole-hour intervals. This limitation significantly affects the model's ability to accurately represent the behaviour of assets, such as PV systems, batteries, and EVs, that operate and respond on sub-hourly timescales, where finer temporal granularity is essential to capture rapid changes in generation, consumption, and flexibility. Therefore, efforts are needed to incorporate minute-level data into CityLearn's data loading process and to extend the datasets accordingly.

The building-level data is loaded within the _load_building method of the citylearn.py module, part of the environment. This method reads energy simulation data as defined in the schema and initialises the corresponding data structures. Although the system supports partial propagation of configuration parameters (e.g., seconds_per_time_step), it lacks mechanisms to interpret or adjust input data dynamically based on the defined simulation time step. As a result, the system is constrained to operate on fixed hourly intervals, which limits its adaptability and accuracy in modelling dynamic, real-world conditions.

Addressing this limitation is critical to enhancing the simulator's fidelity and expanding its applicability to scenarios where minute-level or even finer time resolutions are required. This enhancement would enable more precise modelling of transient behaviours, such as intraday solar variability, rapid load changes, or EV charging patterns, and support more sophisticated control strategies that operate on shorter decision intervals. Implementing support for finer-grained timestamps also opens the door to integrating more granular real-time data streams and forecasting inputs, thereby improving the realism of simulation outcomes.

Design Alternatives Considered

Considering this initial analysis, multiple architectural options were studied evaluated. A fixed multi-resolution approach, supporting only a predefined set of time steps such as 15, 30, or 60 minutes, was initially considered due to its simplicity and predictable behaviour. However, this method proved too rigid and was ultimately dismissed due to its inability to support arbitrary user-defined resolutions. A more sophisticated strategy, involving adaptive resolution that could dynamically adjust Δt during simulation runtime, offered potential for improved efficiency and responsiveness but was deemed too complex and difficult to control deterministically. The approach ultimately selected involved a fully variable resolution system, allowing users to specify any Δt value at configuration time. This method maximised flexibility while enabling consistent, physically accurate results through time normalization. These design considerations are systematised into Table 2.

Table 2 - Systematisation of Alternatives for the Integration of Improved Temporal Granularity into CityLearn

Approach	Description	Advantages	Disadvantages	Outcome
Fixed Multi-Resolution	Support predefined steps (e.g., 15, 30, 60 min)	Simple to implement, bounded behaviour	Limited flexibility; step changes only	Rejected
Fully Variable Resolution	Allow arbitrary Δt per simulation	Maximum flexibility, high precision	Requires careful normalization, added logic	Selected
Adaptive Resolution	Adjust Δt dynamically during simulation	Efficient and adaptive	Complex implementation, non-determinism	Future Consideration

Proposed Design (<u>TO-BE</u> State)

The transition from fixed-hourly to variable temporal resolution represents a fundamental architectural shift in the simulation framework. This transformation is best understood through two complementary perspectives here proposed: the dynamic sequence of operations (Figure 15) and the structural component evolution (Figure 16).

Figure 15 reveals how temporal flexibility is achieved through coordinated system behaviour. The process begins with user configuration, where the specification of seconds_per_time_step (e.g., 900 seconds for 15-minute intervals) and the addition of minute-level timestamps establish the foundation for sub-hourly precision. As the loader processes these enhanced schemas, it dynamically calculates the normalization factor α: the ratio between the current time step and the canonical 3600-second baseline. This α-factor is then propagated through all subsequent energy calculations.

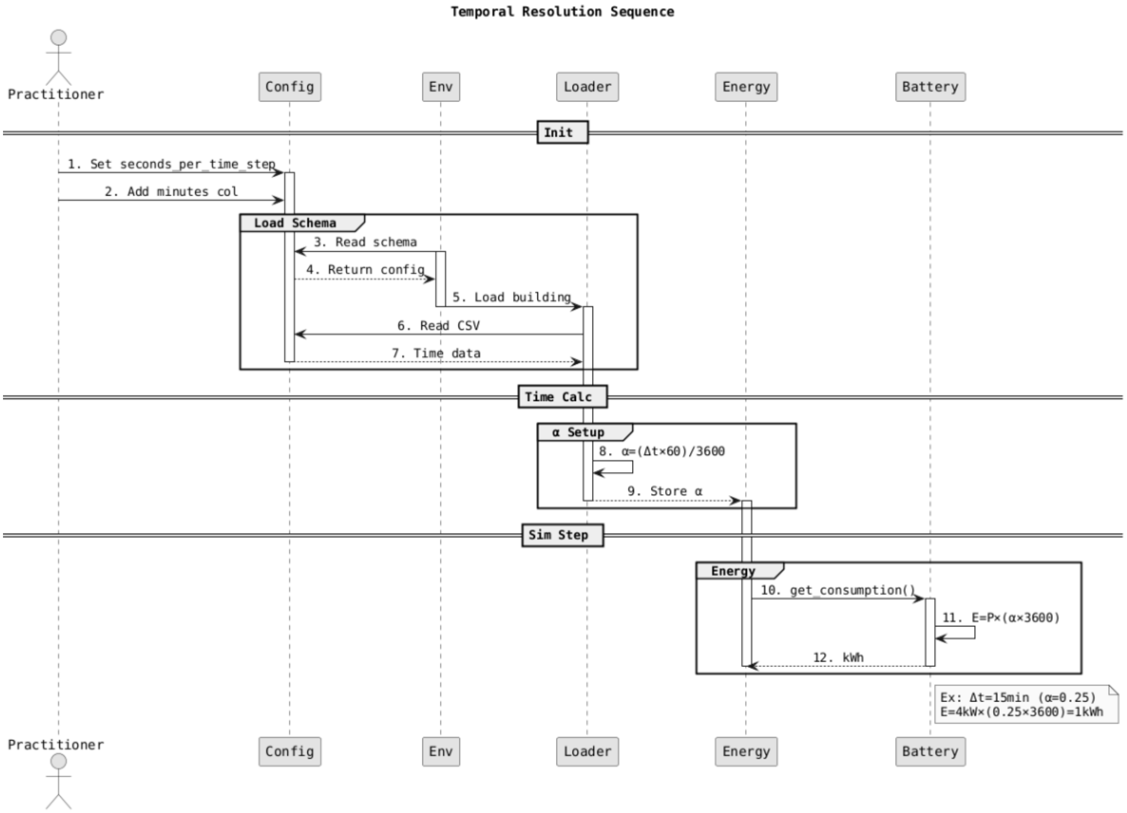


Figure 15 - System diagram: time step setup, data loading, and kWh consumption calculation

Figure 16 exposes the architectural modifications enabling this behavioural change. The schema component evolves from its rudimentary hourly-monthly structure to incorporate minute-level timestamps and explicit time step configuration. This expanded temporal representation flows through enhanced data loaders that automatically detect and normalise time intervals, ultimately reaching energy models that now operate in α-normalised space. Crucially, the architecture maintains backward compatibility. When operating at hourly resolution (α=1), the system behaves identically to its predecessor, ensuring existing simulations remain valid while unlocking new capabilities.

To incorporate minute-level resolution, the load_building() function and underlying schema are extended to parse and validate the newly added minute field, while the schema's structure is augmented to store sub-hourly intervals, alongside traditional hourly-monthly fields. The minutes will then be used to calculate the Δt from the input data's temporal granularity (defaulting to 3600s) for backward compatibility but switching to smaller intervals (e.g., 900s for 15-minute data) when detected. The Figure 16 highlights this duality: the schema's evolution (now timestamp-aware) and the load function's adaptability (now Δt-aware) work in aggregation to enable resolution-agnostic processing.

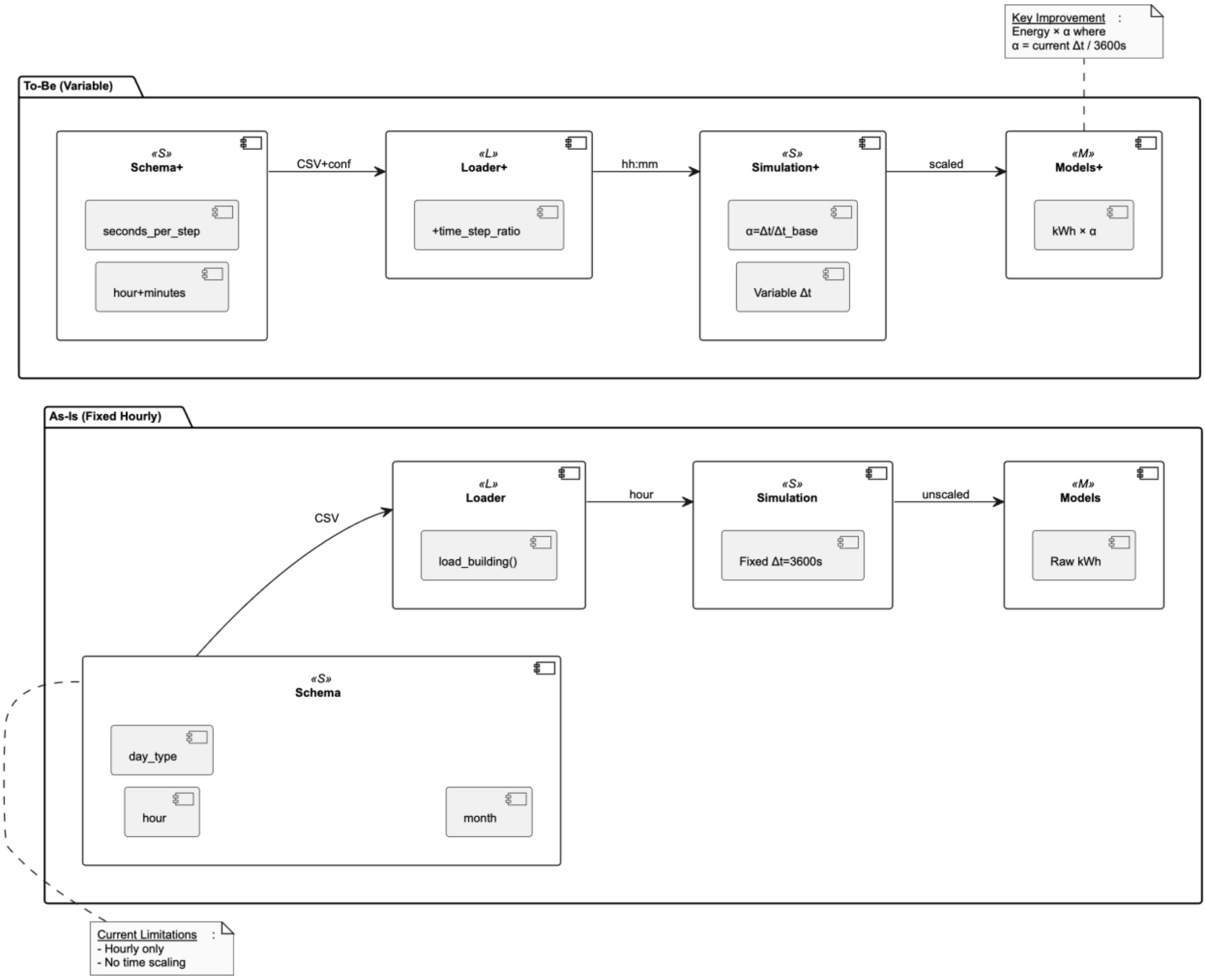


Figure 16 - Component diagram comparing AS-IS (Fixed Hourly) and TO-BE (Variable Time Step) energy simulation architectures

Timestep Normalization Factor

This normalization factor is critical to all energy computations. For instance, the updated battery model no longer relies on a hardcoded hourly resolution; instead, it calculates energy as the product of power and a dynamically determined Δt. A 4 kW load over a 15-minute interval now correctly computes as 1 kWh, maintaining the physical consistency of the simulation. Internally, the α factor is derived by computing the actual time difference between successive entries, accounting for both hour and minute values, and dividing the configured seconds_per_time_step by this interval. This approach is shown in Listing 1, through pseudocode, which illustrates the precise computation of α based on timestamp differentials.

The pseudocode shows how α is calculated step-by-step. First determining the time difference between timestamps (handling edge cases like midnight crossings), then converting it to seconds, and finally computing the ratio between the configured time step and actual elapsed time. This ensures energy calculations remain accurate whether working with 5-minute data, hourly inputs, or irregular intervals.

```
FUNCTION GetTimeRatio(current_hour, next_hour, current_min, next_min, time_step_seconds

    // 1. Calculate basic time difference in hours
    hour_diff = (next_hour - current_hour) * 60  // Convert to minutes

    // 2. If minute data exists, calculate precise difference
    IF current_min AND next_min EXIST THEN
        total_current = (current_hour * 60) + current_min
        total_next = (next_hour * 60) + next_min
        minute_diff = total_next - total_current

        // Handle crossing midnight (e.g., 23:45 to 0:15)
        IF minute_diff < 0 THEN
            minute_diff = minute_diff + 1440  // Add 24 hours in minutes
        END IF
    ELSE
        minute_diff = hour_diff  // Fall back to hourly difference
    END IF

    // 3. Convert to seconds and calculate ratio (α)
    time_diff_seconds = minute_diff * 60
    α = time_step_seconds / time_diff_seconds
    RETURN α
END FUNCTION
```

Listing 1 – Time Ratio (α) Calculation Pseudocode Algorithm

System-Wide Propagation

From a structural standpoint, the system will have to be expanded to propagate the time_step_ratio (α) variable throughout the environment. It will pass from the environment loader to each building, and further down into each asset model such as HVAC units, batteries, and PV systems. These models have will scale their energy outputs and consumptions by α, ensuring consistent behaviour across all time resolutions. For instance, methods returning heating demand or electricity consumption now multiply raw values by this ratio, making energy quantities invariant to the temporal resolution.

Architectural Implications

This flexible time-handling architecture not only meets the immediate needs of modelling diverse energy systems with high responsiveness but also lays the groundwork for future enhancements. These may include real-time simulations, finer-grained control strategies, or hybrid approaches that dynamically adjust resolution based on system states. In sum, the introduction of variable temporal granularity represents a foundational step toward a more extensible, and physically coherent simulation platform, fulfilling sub-objectives SO3 and SO4.

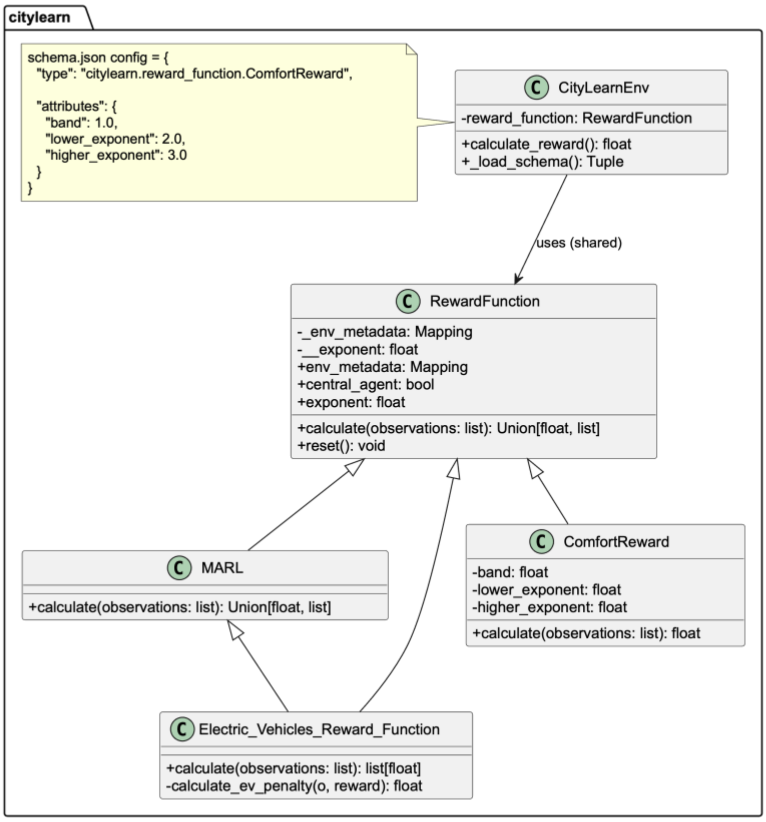
4.3.3. Building-Specific Reward Functions

AS-IS System Analysis

Shifting focus to building-level needs, modern energy communities combine diverse stakeholders (residents, businesses, aggregators, among others) each with unique (and sometimes conflicting) priorities. However, CityLearn's current architecture enforces a global reward function, assuming all agents share a common goal. This limitation hinders the simulation of real-world dynamics where trade-offs, distributed benefits, and fairness must be negotiated.

One key limitation lies in CityLearn's use of a single RewardFunction instance, applied uniformly across all buildings. This uniformity ignores stakeholder heterogeneity and simplifies agent behaviour to the point where it no longer reflects the diverse reality of real-world energy communities.

As illustrated in Figure 17, which represents the current AS-IS class structure, the CityLearnEnv class interfaces with a single reward function, such as ComfortReward or Electric_Vehicles_Reward_Function, and applies it across the estate. This monolithic approach presents several challenges: it prevents modelling of individual vs. community trade-offs, enforces homogenous agent behaviour, and makes distributional equity analysis impossible.



```
schema.json config = {
  "type": "citylearn.reward_function.ComfortReward",

  "attributes": {
    "band": 1.0,
    "lower_exponent": 2.0,
    "higher_exponent": 3.0
  }
}
```

Figure 17 - Class Diagram depicting CityLearn's Reward Function Class Hierarchy

Design Alternatives Considered

Recognising these limitations, three alternative approaches were systematically evaluated (summarised in Table 3). The first alternative, **weighted reward components**, preserved a single global reward function while incorporating building-specific weighting factors. In this formulation, individual priorities were approximated through scalar coefficients applied to common performance metrics (e.g., energy cost, self-sufficiency, or emissions). Although this approach offers computational efficiency and minimal architectural disruption, it fundamentally lacks the capacity to represent divergent behavioural policies. In essence, all agents still converge towards a shared optimisation landscape, limiting the method's suitability for scenarios requiring true policy heterogeneity or asymmetric stakeholder objectives.

The second approach explored, **hierarchical rewards**, introduced a two-tiered reward structure in which agents received both local (building-specific) and global (community-level) feedback. This structure draws from multi-level reinforcement learning and cooperative game theory, aiming to model both self-interest and collective welfare. While conceptually appealing, practical implementation brought significant challenges. Chief among these were the calibration and dynamic weighting of the reward tiers, which proved sensitive to shifting operational contexts and potentially undermined agent learning stability. Due to these operational complexities and their implications for system efficiency, this approach was deferred for future exploration rather than immediate deployment.

Ultimately, **individual reward functions** emerged as the most viable and accurate approach. By assigning a distinct reward class to each building agent, this method enables fully decoupled optimisation pathways. This autonomy is essential for accurately simulating heterogeneous energy communities, where participants may pursue divergent goals (e.g., cost minimisation vs. carbon reduction) due to behavioural, economic, or policy differences. While this architecture imposes a higher configuration burden, it delivers superior representational fidelity. Furthermore, it aligns with emerging paradigms in distributed control and decentralised energy markets, where participant autonomy is a foundational assumption. The additional computational overhead was deemed a reasonable trade-off for the enhanced realism and policy-testing capacity this method enables.

Table 3 - Systematisation of Alternatives Considered for the Integration of Building-Specific Reward Functions

Approach	Description	Advantages	Disadvantages	Selected
Weighted reward components	Single reward function with building-specific weights	Simple implementation, minimal architectural change	Limited expressiveness, all buildings still optimise the same metrics	No
Individual reward functions	Allow each building to specify its own reward class	Maximum flexibility, true heterogeneity	More complex configuration, potential for incompatible rewards	Yes
Hierarchical rewards	Building rewards plus community-level rewards	Models both individual and collective incentives	Complex balancing mechanism needed	No (future work)

Proposed Design (<u>TO-BE</u> State)

This selected design is shown in Figure 18, which introduces a new MultiBuildingRewardFunction class acting as a dispatcher. This system automatically supports two modes: a backward-compatible mode with a shared reward function, and a new heterogeneous mode where each building can be assigned a distinct reward function. In the new architecture, reward routing is dynamic. The MultiBuildingRewardFunction delegates reward computation to building-specific classes, such as Electric_Vehicles_Reward_Function for Building 1 and ComfortReward for Building 2. Configuration is managed through an updated schema.json, which now supports per-building reward definitions.

The reward function thus evolves from Equation 1.

$$R_t = \sum_{\{b \in B\}} r_{b(t)} \quad (1)$$

to a more granular formulation, in Equation 2.

$$R_t^b = f_b\left(s_t^b, a_t^b\right) \forall b \in B \quad (2)$$

Here, each building b calculates its reward R_t^b based on its own state s_t^b and actions a_t^b. A new configuration layer enables assignment of reward classes at the building level, while the dispatcher ensures that each reward is correctly routed and computed.

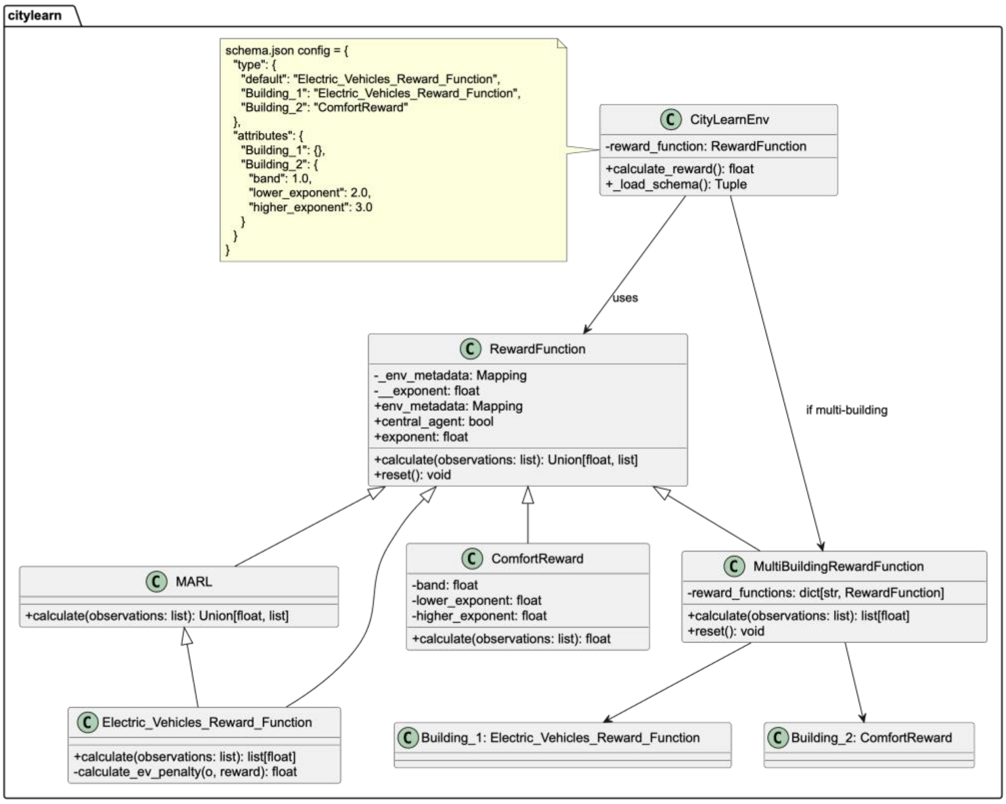

```
schema.json config = {
  "type": {
    "default": "Electric_Vehicles_Reward_Function",
    "Building_1": "Electric_Vehicles_Reward_Function",
    "Building_2": "ComfortReward"
  },
  "attributes": {
    "Building_1": {},
    "Building_2": {
      "band": 1.0,
      "lower_exponent": 2.0,
      "higher_exponent": 3.0
    }
  }
}
```

Figure 18 - Class Diagram: Configurable Multi-Building Rewards in CityLearn via schema.json

This enhancement introduces a more realistic and nuanced approach to incentive modelling. It enables simulations to reflect cooperative strategies, competitive tensions, and fairness considerations, all of which are essential for accurately representing RECs. Thus, this refactoring fulfils sub-objective SO6 while preserving backward compatibility for existing use cases.

4.3.4. Modular Asset Integration

AS-IS System Analysis

Subsequentially, this section examines the integration of two key residential appliances, refrigerators and washing machines, each presenting distinct modelling challenges. Expanding CityLearn's asset library requires consideration of thermodynamic principles and control system architecture. While refrigerators align naturally with existing thermal systems due to their thermodynamic equivalence with heat pumps, washing machines introduce new temporal control dimensions that necessitate architectural extensions. The following analysis justifies these design decisions through both theoretical foundations and practical implementation considerations.

Refrigerators: Thermodynamic Equivalence and Implementation

Refrigerators are essential residential appliances with significant implications for energy consumption and thermal load management. Thermodynamically, a refrigerator operates by extracting heat from its interior and expelling it into the surrounding environment [104]. This process closely mirrors the operation of a heat pump; the two systems are functionally equivalent but applied in opposite contexts. A heat pump transfers heat indoors for warming, while a refrigerator removes heat for cooling. Both rely on the vapor compression cycle and share similar performance characteristics.

A critical metric for evaluating these systems is the **coefficient of performance (COP)**, defined as in Equation 3.

$$COP = \frac{Q_{\text{out}}}{W_{\text{in}}} \quad (3)$$

where Q_{out} represents the heat extracted (in cooling mode) and W_{in} is the electrical work input. The COP reflects system efficiency, with higher values indicating better performance.

Leveraging CityLearn's Existing Framework

Analysing CityLearn's codebase it was identified that it included a well-structured HeatPump class, which already encapsulates the thermodynamic logic needed to model both heating and cooling devices. Key features of this class include:

- **Dynamic COP calculation:** The HeatPump class contains the get_cop method, which dynamically calculates the COP for both heating and cooling modes using Carnot-inspired efficiency equations. For cooling (refrigeration), the COP is computed as:

  ```
  cooling_cop = efficiency * (target_cooling_temperature + 273.15) / (outdoor_temp
  - target_cooling_temperature)
  ```

which is the code equivalent to the formula present at Equation 4.

$$COP_{\text{cooling}} = \eta \cdot \frac{T_{\text{cool}}+273.15}{T_{\text{out}}-T_{\text{cool}}} \quad (4)$$

- Configurable **cooling/heating modes**, allowing the system to operate as a refrigerator by setting an appropriate target temperature (e.g., 4°C to 8°C).
- The electrical input power required by the heat pump is calculated dynamically during **simulation** or **runtime**.

Given these capabilities, the HeatPump class can accurately simulate a refrigerator's energy consumption and thermal behaviour without modification. By constraining the system to cooling mode and setting a refrigeration-appropriate temperature, the existing implementation becomes a functionally equivalent model.

Justification for Reuse

The physical and operational similarities between heat pumps and refrigerators, combined with CityLearn's built-in support for dynamic COP modelling, make repurposing the HeatPump class both practical and theoretically effective. This approach avoids redundant code while maintaining accuracy and consistency within the simulation framework.

Integration of Washing Machine

Thereafter, washing machines were analysed. These assets, particularly those with temporal flexibility, offer latent potential for demand-side management (DSM) [105]. Most existing models treat appliances as part of an undifferentiated demand curve, failing to expose their control affordances.

Design Alternatives Considered

To integrate washing machines into the control framework, three strategies were evaluated: (1) incorporating them into the building's aggregate load, (2) modelling them as independent controllable assets, and (3) coupling CityLearn with an external simulator.

Strategy 1 treated washing machines as undifferentiated components of the building's overall energy demand. This approach required minimal changes to the control pipeline but made it impossible to impose appliance-level constraints (e.g., scheduling windows, runtime limits) or issue targeted control actions.

Strategy 3 proposed a co-simulation setup, where a separate simulator would model the appliance in detail and communicate with CityLearn through an interface. While this allowed for high model fidelity, it significantly increased system complexity, introduced synchronisation challenges, and posed performance risks during training and evaluation.

Strategy 2, selected for implementation, models the washing machine as a first-class controllable entity within CityLearn. This enables direct control of its operation, enforces appliance-level constraints, and preserves compatibility with CityLearn's modular architecture.

It achieves a practical balance between control flexibility, system complexity, and maintainability. A summary of the alternatives is provided in Table 4.

Table 4 - Systematisation of Alternatives Considered for the Integration of Washing Machines

Approach	Description	Advantages	Disadvantages	Selected
Building-integrated model	Model appliances as part of building load	Simplicity, minimal architectural changes	Limited expressiveness, inability to control individual appliances	No
First-class asset model	Model appliances as separate controllable assets	Explicit control, flexibility for future extensions	More complex observation/action space	Yes
External co-simulation	Use separate appliance simulator connected to CityLearn	Maximum detail in appliance models	Integration complexity, potential performance issues	No

Proposed Design (<u>TO-BE</u> State)

This approach introduces the washing machine directly into the CityLearn domain model, as seen in Figure 19. Each appliance instance is associated with a specific building, maintains its own internal state and constraints, and contributes to the building's observation space.

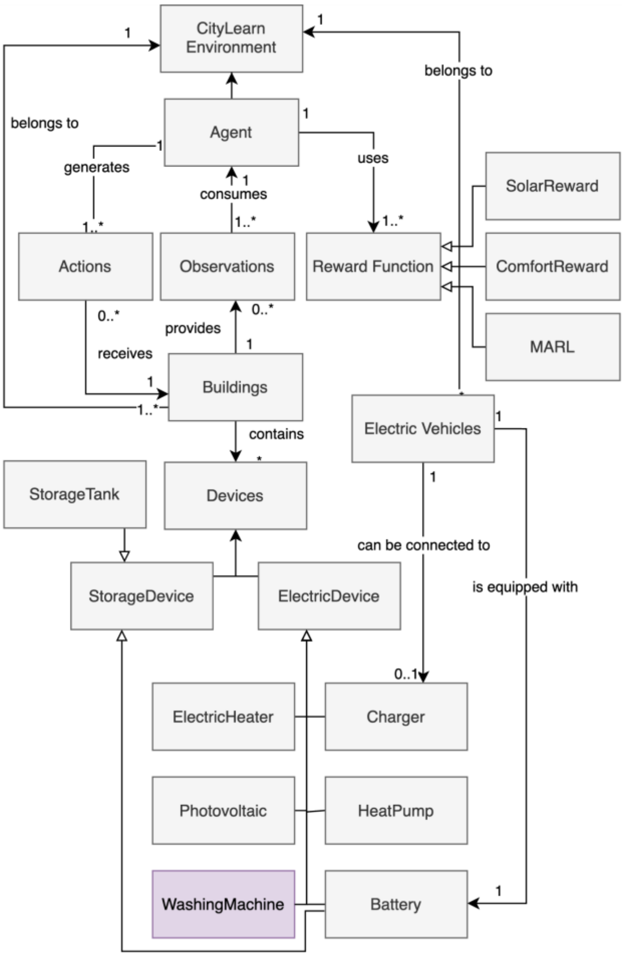


Figure 19 - CityLearn's Domain Model with added WashingMachine (adapted from [103])

Additionally, the washing machine asset schema could be defined by Table 5:

Table 5 - Washing Machine Schema Attributes

Name	Description
day_type	Indicates whether the day is a weekday or weekend
hour	Current hour of the simulation
wm_start_time_step	Earliest start time for the washing machine cycle
wm_end_time_step	Latest end time for the washing machine cycle
load_profile	Power profile required to complete the washing cycle

In this design, each washing machine is linked to a specific building and modelled as a discrete asset with its own internal state, control logic, and constraints. The simulation considers key temporal indicators such as the current simulation hour and whether the day is a weekday or weekend.

Additionally, each appliance defines user-specified flexibility windows, parameters that dictate the earliest and latest allowable start times for a wash cycle. The decision-making process is governed by the control algorithm, which can vary the timing and frequency of cycle initiation based on external factors such as electricity prices or grid conditions. As a result, some washing cycles may run to completion without interruption, while others may be strategically deferred or distributed across flexibility windows to optimise overall performance.

Lastly, the load profile is defined as a sequence (or array) of float values, each representing the amount of energy (in kilowatt-hours) consumed by the appliance at each time step during the wash cycle. These profiles can be derived from empirical data collected from actual machines, capturing the non-uniform energy consumption patterns typical of modern washing processes (e.g., initial water heating, intermittent agitation, spin cycles).

Additionally, the appliance contributes directly to the observation space of the agent. Relevant observation variables include the permitted time window for initiating a wash cycle, exposed through the configuration seen in Listing 2.

```
"washing_machine_start_time_step": {
  "active": true,
  "shared_in_central_agent": true
},
"washing_machine_end_time_step": {
  "active": true,
  "shared_in_central_agent": true
}
```

Listing 2 - Washing Machine Observation Variables Configuration

The agent's action space is also extended to include control over the washing machine through a single binary decision variable, as shown below in Listing 3.

```
"washing_machine": {
  "active": true
}
```

Listing 3 - Washing Machine Action Variables Configuration

This action indicates whether the washing machine should begin a cycle during the current time step, provided the decision falls within the user-defined flexibility window. Once triggered, the appliance transitions into an active state and executes its load profile.

By explicitly modelling both constraints and affordances of the washing machine, this design enables the simulation of more realistic and fine-grained demand response strategies. It allows RL agents to reason not only about energy usage, but also about temporal flexibility and user preferences, capabilities that are essential for advancing residential DSM. The resulting model meets the functional requirements specified in SO7 and provides a framework for the future integration of other discrete, time-constrained appliances within CityLearn. The next subsection addresses the inherent uncertainties in real-world energy systems and their impact on simulation fidelity.

4.3.5. Incorporation of Stochasticity

AS-IS System Analysis

Real-world energy systems are inherently uncertain [106]. Key inputs such as solar irradiance, electrical load, and market prices are subject to stochastic variation, driven by weather patterns, human behaviour, and economic dynamics. Simulation environments that rely solely on deterministic inputs fail to capture this variability, potentially overfitting agents to idealised conditions and leading to brittle or unrealistic demand-side control strategies.

Design Alternatives Considered

Three primary approaches were considered. Scenario-based modelling enables structured stress-testing but offers only coarse detail, limiting its ability to capture fine variations. Historical perturbation captures complex statistical structures but introduces dependency on extensive historical datasets. For initial prototyping, Gaussian noise injection was selected for its simplicity, configurability, and ability to approximate uncertainty without compromising performance or architectural compatibility. This was systematised into Table 6.

Table 6 - Systematisation of Alternatives Considered for the Incorporation of Data Stochasticity

Approach	Description	Advantages	Disadvantages	Selected
Simple Gaussian noise	Inject additive/multiplicative zero-mean noise	Low computational cost, configurable	Ignores temporal correlation, simplified model	**Yes (Phase 1)**
Historical perturbation	Use real-world data with stochastic resampling	High realism, preserves statistical structure	Data-hungry, inconsistent data availability	No
Scenario-based simulation	Execute distinct simulation runs with defined inputs	Easy comparison, interpretable extremes	Discrete coverage of the uncertainty space	No

Proposed Design (TO-BE State)

The proposed stochastic simulation framework introduces controlled variability by applying zero-mean Gaussian noise perturbations to deterministic inputs [107]. This approach preserves the fundamental system structure while generating subtly different conditions across simulation runs. The noise magnitude is precisely controlled through configurable standard deviation parameters (noise_std), which can be independently specified for different input streams (e.g., load profiles, solar generation, electricity prices) and individual agents (e.g., specific buildings or EVs).

The design incorporates several important features that maintain system integrity while adding stochastic capabilities. First, the configuration system allows for control over noise levels for each input channel and entity, enabling targeted variability studies. The configuration, seen in Listing 4, will be added into the global schema csv file, where one can select the noise value through the building or EV.

```
"building_1": {
  "noise_std": 0.1
},
"electric_vehicle_1": {
  "noise_std": 0.07
}
```

Listing 4 - Stochastic Capabilities Variable Configuration

Secondly, the implementation carefully minimises architectural impact by applying perturbations exclusively during data ingestion, leaving downstream simulation logic untouched. Additionally, this method preserves full temporal fidelity, maintaining the resolution and characteristic shape of the input signals while introducing realistic variability. This process is mathematically represented by Equation 5.

$$\tilde{x}(t) = x(t) + \epsilon(t), \epsilon(t) \sim \mathcal{N}(0, \sigma^2) \quad (5)$$

Where:

- $x(t)$ is the original deterministic input at time t,
- $\tilde{x}(t)$ is the perturbed input,
- $\epsilon(t)$ is Gaussian white noise with standard deviation σ.

This balanced combination of configurability, non-invasiveness, and temporal fidelity makes the solution particularly suitable for studying energy system behaviour under uncertainty.

Architectural View

In order to facilitate this, a dedicated noise injection module is positioned between the deterministic input loader and the input processor. This component draws its parameters from the simulation schema and applies the corresponding perturbations before inputs reach the learning agents. This integration is visually represented in the component diagram (Figure 20), which highlights the modular nature of the implementation.

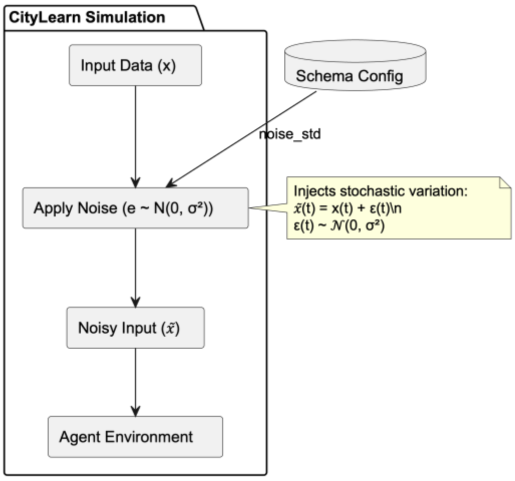


Figure 20 - Component Diagram depicting CityLearn Simulation with Noisy Inputs

By introducing configurable Gaussian noise to key inputs, the simulation gains the ability to emulate uncertainty and non-determinism in energy environments. This approach supports more resilient agent policies, aligns with CityLearn's modular architecture, and fulfil sub-objective SO8. Future iterations may extend this foundation with more sophisticated stochastic models, including time-correlated noise or empirical distributions derived from real-world data.

4.3.6. Data Export Functionality

AS-IS System Analysis

In the context of MARL for smart energy systems, the ability to systematically export and analyse simulation data is critical for validation, benchmarking, and policy improvement. The CityLearn environment, while adept at simulating complex interactions between buildings, EVs, and power grids, currently lacks a unified framework for exporting time-series data and KPIs. This gap introduces friction in downstream workflows, particularly for offline policy evaluation and comparative algorithm analysis, where reproducibility and interoperability are paramount [108]. Without a standardised export mechanism, researchers must resort to ad-hoc solutions, ranging from fragmented log files to custom binary dumps, which often sacrifice semantic clarity, storage efficiency, or compatibility with analytical tools.

This section presents the design and rationale for a centralised export system that addresses these limitations while adhering to the OpenAI Gym interface conventions, as was previously explored in Section 2.5, namely through the method render(). The solution prioritises extensibility, ensuring compatibility with both current needs (e.g., analysis for the OPEVA project) and future requirements (e.g., database integration or hierarchical data formats).

Design Alternatives Considered

Three primary architectures were evaluated for exporting simulation data: flat CSV files, hierarchical JSON documents, and direct database integration. Each approach was assessed based on its ability to balance four competing demands: fidelity to the simulation's semantic structure, computational and storage overhead, interoperability with analytical tools, and ease of implementation.

The CSV-based approach was selected as the most practical initial solution due to its broad compatibility and lower operational overhead, compared to the others. By organising data in a tabular format with composite column headers (e.g., Net_Electricity_Consumption-kWh), it strikes as a balance between structural clarity and implementation simplicity. To support frontend processing, each column explicitly includes its unit of measurement, facilitating the accurate interpretation and visualisation of the data. Although CSV lacks native support for hierarchical relationships, such as EV charging sessions nested within buildings, this limitation is addressed through a well-defined schema that flattens nested structures using dot-delimited field names, supplemented by a machine-readable schema descriptor for semantic context.

JSON export, though capable of preserving rich hierarchical relationships, was deemed less suitable for the initial phase due to its storage inefficiency and parsing complexity. Some benchmark tests, using already existing datasets in CityLearn, was done within a controlled development environment, which demonstrated CSV's superior performance for this application, this can be viewed in Appendix F. This performance difference, combined with CSV's natural alignment with our tabular time-series data structure, made it the clear choice for our high-volume simulation outputs, containing thousands of rows. Similarly, direct database integration, though advantageous for large-scale deployments, introduces significant implementation complexity and potential latency issues. Therefore, the database integration was reserved for future iterations currently being developed under a separate objective of the OPEVA project. These findings are systematically summarised in Table 7, which compares the trade-offs between CSV, JSON, and database storage approaches.

Table 7 - Systematisation of Alternatives Considered for the Integration Data Export Functionality in CityLearn

Approach	Description	Advantages	Disadvantages	Selected
CSV export	Export flat, tabular data in CSV format	Simple, universal compatibility	Limited structure for complex data	Yes
JSON export	Export hierarchical data in JSON format	Rich structure, native web support	Larger file size, more complex parsing	No
Database integration	Write results directly to a database	Query capabilities, scalability	External dependencies, complex setup	No (ongoing work by another team in the broader project may support future integration)

Proposed Design (TO-BE State)

The proposed export system is structured as a modular pipeline, decoupling data collection, aggregation, and serialization to accommodate future extensions. At its core lies the OpenAI Gym render() method, which orchestrates the export process at each simulation timestep. When invoked, render() iterates over all active entities, buildings, EVs, and grid components, invoking their respective as_dict() methods to capture their state variables as key-value pairs.

The serialization layer will currently default to CSV output but is designed to support alternative backends. The CSV serializer employs several optimisations to minimise overhead: buffered I/O reduces disk contention, delta encoding logs only changed fields in high-frequency modes, and episodic segmentation organises outputs into subdirectories for improved traceability. A companion schema.json file documents the semantics and units of each exported field, ensuring reproducibility and aiding third-party tooling.

Schema and Downstream Applications

The exported data schema is designed to accommodate both CityLearn’s core dynamics and project-specific extensions. Mandatory fields include timestamps (ISO 8601 format) [109], episode identifiers (Universally Unique Identifiers (UUIDs)), and entity-level metrics (e.g., building power demands, EV state-of-charge values). KPIs such as carbon intensity and demand variance are appended as derived columns, computed during the aggregation phase. This schema has been validated against the requirements of the OPEVA project, where it serves as the foundation for visualisation dashboards.

The simulation runtime autonomously manages temporal progression through an event loop that repeatedly invokes next_time_step() until termination conditions are met. This loop constitutes the core heartbeat of the CityLearn environment, advancing the virtual clock and synchronising state updates across all entities (buildings, EVs, grid components). Within this automated cycle, the export system intervenes to capture snapshots of the simulation state.

The render() mechanism is triggered as the first operation within each timestep transition, implementing a write-ahead logging pattern that guarantees data is persisted before any state modifications occur. The exporter operates as a passive observer, collecting data through the environment's as_dict() interface without perturbing the simulation physics. This interaction is illustrated in Figure 21.

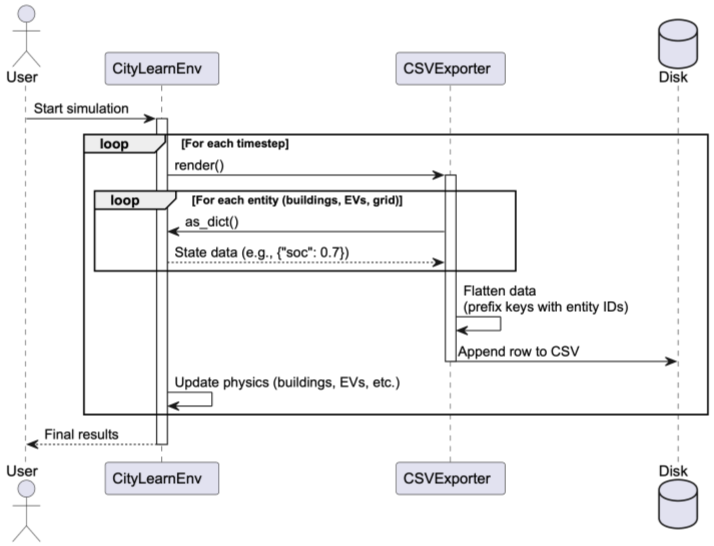


Figure 21 - Sequence Diagram demonstrating CityLearnEnv’s Simulation Loop and Data Export

Building on the preceding analysis and design considerations, the core export mechanism can be formalised, enhanced by the insights discussed earlier, into a systematic procedure, designated as Listing 5. This pseudocode algorithm takes three inputs: the simulation environment state (env), the current timestep (timestep), and an identifier for the active episode (episode_id). Its output consists of CSV files that record the full system state at each timestep, ensuring structured and reusable data for subsequent analysis.

```
Function render(env, timestep, episode_id):

    timestamp ← generate_iso8601_timestamp(env)
    export_dir ← create_export_directory(episode_id)

    # Export community-level metrics
    community_data ← env.as_dict()
    append_csv(export_dir + "/community.csv", timestamp, community_data)

    # Export data per building and subsystems
    For each building in env.buildings Do
        building_data ← building.as_dict()
        append_csv(export_dir + f"/building_{building.id}.csv", timestamp, building_data)

        battery_data ← building.battery.as_dict()
        append_csv(export_dir + f"/building_{building.id}_battery.csv", timestamp, battery_data)

        For each charger in building.chargers Do
            charger_data ← charger.as_dict()
            append_csv(export_dir + f"/building_{building.id}_charger_{charger.id}.csv", timestamp, charger_data)
        End For
    End For

    # Export electric vehicle data
    For each ev in env.electric_vehicles Do
        ev_data ← ev.as_dict()
        append_csv(export_dir + f"/ev_{ev.id}.csv", timestamp, ev_data)
    End For

    # Export pricing and market signals
    pricing_data ← env.pricing.as_dict(timestep)
    append_csv(export_dir + "/pricing.csv", timestamp, pricing_data)

End Function
```

Listing 5 - render() Pseudocode Algorithm

4.3.7. Refined EV Modelling

EV usage in real-world settings is highly intermittent, with frequent shifts between home charging and travel. Earlier vehicle-centric models tracked each EV’s SOC independently, capturing behaviour but poorly reflecting shared infrastructure and household routines. On the contrary, charger-centric models focus on infrastructure but often ignore actual vehicle presence. This mismatch is especially limiting in REC simulations, where EV availability is shaped more by occupant routines than charger status.

Proposed Design (TO-BE State)

To resolve this mismatch, the simulation is restructured around chargers rather than vehicles. A new ChargerSimulation class replaces the legacy ElectricVehicleSimulation shifting from a vehicle-centric to a charger-centric perspective. Each charger now acts as an independent state machine that models vehicle presence, reflecting real usage patterns, shown in Listing 6.

```
class ChargerSimulation:
    STATES = {
        1: 'Connected',    # Vehicle plugged in
        2: 'Incoming',     # Vehicle en route
        3: 'Commuting'     # Vehicle away

    }
```

Listing 6 - Charger Simulation State Machine

The state transitions follow Equation 6, where SOC evolves differently per state:

$$SOC_{t+1} = \begin{cases} f_{charge}(P_t) & \text{if state=1} \\ SOC_t \cdot \eta_{drift} + \epsilon & \text{if state}\in\{2,3\} \\ SOC_{arrival} & \text{at transition } 2\rightarrow 1 \end{cases} \tag{6}$$

The state-dependent SOC evolution uses three cases: (1) $\mathrm{f_{charge}(P_t)}$ (charging when plugged in), (2) $SOC_t \cdot \eta_{drift} + \epsilon$ (self-discharge with noise during transit), and (3) $SOC_{arrival}$ (preset value when reconnecting).

This shift, reframing simulation around chargers instead of vehicles, resolves a long-standing misalignment in EV modelling in CityLearn. By embedding behaviour in each charger's local state machine, the system reflects real-world usage driven by human routines without requiring centralised coordination. What previously demanded complex scheduling logic now emerges naturally from local rules. This minimalist change delivers disproportionate gains in realism and efficiency for REC simulations.

This design fulfils sub-objective SO10 by accurately modelling residential EV behaviour while preserving the efficiency required for large-scale REC simulations.

4.4. Conclusions taken from the chapter

This section has presented a comprehensive analysis and design of an enhanced simulation platform built upon the CityLearn framework, with the rationale for each architectural decision. The proposed improvements systematically address critical limitations in modelling energy communities through five key innovations: **Improved Temporal Granularity**, **Building-Specific Reward Functions** supporting heterogeneous agent objectives, expanded **Modular Asset Integration** including controllable appliances like washing machines and standardised data export in CSV format. The architecture further **Incorporates Stochastic Variability** to better capture real-world uncertainties while implementing an easily extensible **Data Export** mechanism to ensure reproducibility and empower REC managers with a tool for data analysis.

Together, these enhancements significantly advance the platform's capabilities, achieving greater realism, flexibility, and scalability while maintaining strict adherence to both functional and non-functional requirements. Crucially, the design preserves backward compatibility, allowing the efficient integration with existing workflows while introducing powerful new features for energy systems research.

5. Implementation of the solution

This chapter implements Chapter 4's architectural framework into the enhanced CityLearn platform. Through code examples and SOLID design principles, developed backward-compatible, modular components that meet functional (SO3-SO7) and non-functional FURPS+ requirements, with special emphasis on performance and maintainability.

This chapter is separated into the 5 big contributions of this dissertation:

1. **Data Export Functionality (5.1):** Which introduces a render() hook that implements hierarchical CSV export functionality with ISO 8601 timestamp support, cleanly separating data capture from simulation logic in accordance with the Single Responsibility Principle.
2. **Improved Temporal Granularity (5.2**): Which implements α-coefficient-based power normalization that gracefully handles sub-hourly intervals while maintaining backward compatibility with hourly datasets, demonstrating the Open/Closed Principle through extensible time-step management.
3. **Modular Asset Integration (5.3):** expands the simulator's capabilities by adding a new WashingMachine asset with configurable load profiles and trigger-based activation, following interface segregation principles.
4. **Building-Specific Reward Functions (5.4):** employs a MultiBuildingRewardFunction architecture that applies Dependency Inversion, allowing flexible, schema-driven reward structures tailored to individual buildings.
5. **Incorporation of Stochasticity (5.5):** physics-aware noise was implemented by extending the TimeSeriesData class in a way that stays true to Liskov Substitution, ensuring mathematically valid perturbations of environmental inputs.

Additionally, a testing suite was added with unit tests, integration tests, and GitHub Actions Continuous Integration (CI) to improve reliability. Foundational tests now cover new features and critical legacy code, addressing the previously untested codebase.

5.1. Data Export Functionality

Centralised data export in CityLearn was enabled by augmenting the simulation architecture with a render() call, integrated into the simulation's temporal loop, as seen in Listing 7.

```
def next_time_step(self):
    self.render()  # Data capture hook
    for building in self.buildings:
        building.next_time_step()
    for electric_vehicle in self.electric_vehicles:
        electric_vehicle.next_time_step()
    super().next_time_step()
    self.associate_electric_vehicles_to_chargers()
```

Listing 7 - CityLearn Time Step Progression Method

This extension introduces a data capture hook that executes at every simulation time step. It enables systematic snapshots of both global (community-level) and local (building-level) state without interfering with the forward progression of time. The render() method orchestrates the export process, ensuring that data is captured just before each state transition, reflecting the pre-transition state of the system.

The render() method itself handles the structured, hierarchical collection and persistence of simulation state. Because CityLearn natively operates in discrete time steps, an additional layer was added to convert these steps into ISO 8601 timestamps. This makes the exported data easier to interpret and aligns it with real-world temporal structures. The resulting timestamps are included in CSV files for both individual buildings and aggregate community metrics, as demonstrated in Listing 8.

```
def render(self):
    """Coordinates parallel data export for all simulation entities"""
    iso_timestamp = self._get_iso_timestamp()
    os.makedirs(self.export_dir, exist_ok=True)

    # Community-level metrics
    self._save_to_csv(
        f"community_ep{self.episode_num}.csv",
        {"timestamp": iso_timestamp, **self.as_dict()}
    )

    # Building-specific data streams
    for building in self.buildings:
        self._save_to_csv(
            f"{building.name}_ep{self.episode_num}.csv",
            {"timestamp": iso_timestamp, **building.as_dict()}
        )
```

Listing 8 - Parallelised Data Export Method

Each data record is appended to its respective file using _save_to_csv, which also ensures headers are written when a file is created. The function automatically constructs filenames to reflect the entity and episode, creating a self-organising log directory suited for both real-time and post-simulation analysis, as shown in Listing 9.

```
def _save_to_csv(self, filename, data):
        """
        Saves data to a CSV file, appending it if the file exists.
        """
        file_path = os.path.join(self.new_folder_path, filename)
        file_exists = os.path.isfile(file_path)

        with open(file_path, 'a', newline='') as csvfile:
            fieldnames = list(data.keys())
            writer = csv.DictWriter(csvfile, fieldnames=fieldnames)

            if not file_exists:
                writer.writeheader()
            writer.writerow(data)
```

Listing 9 - Atomic CSV Writer Method

Each export operation serialises a snapshot of the system's metrics into a CSV file, appending to a file designated by entity and episode. Timestamps are generated using a custom ISO format constructor designed specifically for this task. It reconstructs real-world timestamps using available simulation parameters such as hour, minute, day, and month, as shown in Listing 10.

```
def _get_iso_timestamp(self):
    sim = self._get_energy_simulation()
    month = sim.month[self.time_step]
    hour = sim.hour[self.time_step]
     // Optional for backward compatibility
    minutes = getattr(sim, "minutes", [0])[self.time_step]

    next_timestamp = self.time_step + 1
    next_month = sim.month[next_ts] if next_ts < len(sim.month) else month
    next_hour = sim.hour[next_ts] if next_ts < len(sim.hour) else hour
    next_minutes = getattr(sim, "minutes", [minutes])[next_ts] if next_ts <
len(sim.hour) else minutes

    if next_month != month:
        self.current_day = 1
        if month == 12 and next_month == 1:
            self.year += 1
    elif next_hour == 1 and next_minutes == 0:
        self.current_day += 1

    return f"{self.year:04d}-{month:02d}-
{self.current_day:02d}T{hour:02d}:{minutes:02d}:00"
```

Listing 10 - Temporal Context Manager Method

To ensure clarity and consistency in the exported data, each simulation entity implements an as_dict() method, which encapsulates the logic for exposing relevant metrics in a clean, flat dictionary format. Each key is a human-readable label corresponding to a specific metric, and each value reflects the entity's current reading at that time step. As shown in Listing 11, this method is implemented by all asset-representing classes and is invoked during the export process, as demonstrated in Listing 8.

```
def as_dict(self) -> dict:
    """Standardized metric extraction for serialization"""
    return {
        "Net Electricity Consumption-kWh":
            self.net_electricity_consumption[self.time_step],
        "Solar Generation-kWh":
            self.solar_generation[self.time_step],
        "EV Contribution-kWh":
            self.energy_production_from_ev[self.time_step]
    }
```

Listing 11 – As Dictionary Example implementation

This design adheres to the Information Expert principle: rather than centralizing knowledge of what to export within the rendering system, each component (e.g., buildings, batteries, EVs) is responsible for exposing its own state. This decoupling enhances modularity and maintainability, allowing for easy extension of metrics without modifying the core export logic.

The result is a flexible and scalable data pipeline, producing semantically labelled output. With consistent formatting, the exported data can be ingested directly into downstream systems for visualisation, statistical analysis, or machine learning.

Adopted as the default export mechanism in CityLearn, this system has proven to be extensible, and well-suited for other OPEVA initiatives, such as the dashboard data visualizing one [26]. Whether integrating new data formats, interfacing with external telemetry services, or expanding the set of tracked metrics, this architecture provides a principled and future-proof foundation for high-resolution simulation analysis.

5.2. Improved Temporal Granularity

The top priority was enhancing CityLearn's temporal resolution from fixed hourly intervals to minute-level granularity. This feature introduces the capacity to model short-term fluctuations and transient behaviours with increased precision which is crucial for advanced energy systems that operate on sub-hourly timescales, such as fast-charging EV stations or demand response mechanisms. The shift necessitated a cohesive redesign across schema handling, time-step normalization, and energy model computations, that was already exposed in Section 4.3.2.

Schema Extension and Temporal Precision

To support minute-level resolution in energy simulations, the EnergySimulation class, responsible for loading and parsing data from CSV files, was updated to accept an optional minutes parameter. This addition allows each time step to be timestamped to the minute, significantly improving the temporal fidelity of simulations.

Within the constructor, the minutes column is conditionally converted to a NumPy array, ensuring backward compatibility with legacy datasets that contain only hour-level timestamps. Integration into the data schema is fully backward compatible: data are unpacked using the to_dict('list') method, and the minutes field is handled gracefully if present.

By explicitly constructing simulation timestamps using the minutes field, the system now supports arbitrary resolutions, from coarse hourly intervals to high-frequency, minute-by-minute sampling. This improvement aligns the simulation framework more closely with real-world energy systems and provides a consistent basis for time-resolved data analytics.

Dynamic Time-Step Normalization

To accurately convert power (W) to energy (Wh or kWh) in simulations with non-uniform or sub-hourly time steps, a normalization factor, called the alpha (α) coefficient, is computed. This coefficient adjusts energy calculations based on the actual elapsed time between consecutive entries in the input data.

By default, time_delta is calculated using the difference in hour values between two consecutive time steps. However, if the minutes field is present (i.e. sub-hourly resolution is supported), the code refines this calculation by converting both the hour and minute fields into total minutes past midnight (t0 and t1) and computing their difference. This ensures the time delta reflects the true time separation between samples.

To handle edge cases such as midnight rollovers (e.g., from 23:59 to 00:01), a correction is applied: if time_delta is negative, it wraps around by adding 1440 minutes (i.e., one full day), preserving the integrity of the simulation timeline.

Once time_delta (in minutes) is known, it is converted to seconds and used to normalise the simulation's fixed time step (seconds_per_time_step). Specifically, the time_step_ratio is computed, as seen in Listing 12.

```
time_step_ratio = (
            # Computes the ratio of the current time step (in seconds)
relative to:
            # - 1 hour (3600s) if time_delta ≤ 1 hour, OR
            # - time_delta (converted to seconds) if time_delta > 1 hour
            # Returns None if either time_delta or seconds_per_time_step is
missing
            seconds_per_time_step / max(3600, time_delta * 60)
            if time_delta is not None and seconds_per_time_step
            else None
        )
```

Listing 12 – Time Step Ratio Computation Implementation

System-Wide Ratio Propagation

Once computed, the α coefficient is propagated system-wide via constructor dependency injection, a deliberate design decision that enforces immutability and modularity. For instance, when initializing a Building object within the simulation environment, the α value is passed explicitly, as seen in Listing 13.

```
building = Building(..., time_step_ratio=alpha)
```

Listing 13 – Time Step Ratio Constructor Initialisation

Each asset or subsystem that depends on time-aware calculations, such as batteries or EVs, receives the α ratio and applies it locally during energy computation. This propagation mechanism satisfies three key architectural goals:

1. **Single Source of Truth:** α is calculated once per time step during simulation setup.
2. **Immutable Propagation:** Components receive but do not modify the coefficient, ensuring consistency.
3. **Atomic Utilization:** Energy calculations apply α precisely at the point of conversion, maintaining mathematical integrity.

Energy Model Refactoring

All energy models were refactored to incorporate α during internal calculations. A representative example is the battery subsystem, whose charge method was modified to include the time-step ratio in its energy conversion logic, as shown in Listing 14:

```
def charge(self, energy: float):
        energy = energy * self.time_step_ratio
        energy_init = self.energy_init
```

```
        # The initial State Of Charge (SOC) is the previous SOC minus the
energy losses
        energy_final = min(energy_init + energy*self.round_trip_efficiency,
self.capacity) if energy >= 0\
            else max(0.0, energy_init + energy/self.round_trip_efficiency)

        self.__soc[self.time_step] = energy_final/max(self.capacity,
ZERO_DIVISION_PLACEHOLDER)
        self.__energy_balance[self.time_step] =
self.set_energy_balance(energy_final, energy_init)
```

Listing 14 - Energy Model Refactoring Implementation

This modification, which was replicated across all relevant subsystems, ensures that energy deltas faithfully reflect the actual duration of power flow, yielding physically accurate simulation outcomes. For instance:

- A 4 kW load over 15 minutes (α = 0.25) yields an energy delta of 1.0 kWh.
- A 2 kW generation over 45 minutes (α = 0.75) yields 1.5 kWh added to storage or offset from load.

These results are consistent with analytical expectations and enhance the physical realism of simulations, especially in dynamic or high-frequency contexts.

Summary and Impact

The incorporation of minute-level granularity and dynamic time normalization positions CityLearn to more accurately reflect the complexities of real-world energy ecosystems. By embedding time sensitivity at the core of every calculation, the system supports fine-grained control policies, real-time responsiveness, and precision-guided decision-making algorithms.

Crucially, these improvements were implemented without compromising compatibility with existing workflows, thereby preserving accessibility while significantly extending capability.

This advancement reinforces CityLearn’s role as an efficient platform for high-resolution energy modelling and experimentation. It also lays the foundation for future features such as adaptive time-stepping, real-time scenario forecasting, and the integration of telemetry data with irregular intervals. Through principled software design and the systemic integration of α-normalization, variable temporal granularity has become a first-class, scalable, and forward-compatible feature within the CityLearn ecosystem.

5.3. Modular Asset Integration

Following the refinement of time step granularity, development efforts in CityLearn moved towards the integration of new controllable appliances. This section presents the complete implementation of a washing machine model, encompassing both dishwashers and clothes washers, into the CityLearn simulation environment. The integration emphasises high-resolution control, observability, and schema conformity, with each design choice illustrated through corresponding code excerpts.

Washing Machine Model and Control Logic

To accurately simulate the behaviour of a residential washing machine as a controllable load, a dedicated WashingMachine class was introduced as a subclass of the generic Appliance base class. This component models a fixed-sequence load that can only be activated within user-defined operational windows, making it suitable for demand response experiments. Listing 15 shows the class initialisation, where core variables are defined.

```
class WashingMachine(Appliance):
    def __init__(self, start_time, end_time, load_profile):
        self.start_time = start_time
        self.end_time = end_time
        self.load_profile = load_profile
        self.active = False
        self.current_step = 0
```

Listing 15 - Washing Machine Class Initialisation Method

The class maintains internal state variables that track whether a cycle is currently active (self.active) and which step of the load profile is being executed (self.current_step). The load_profile array represents the energy draw of the washing machine over time once a cycle is initiated.

At each simulation step, the agent may choose to trigger a washing cycle via the start_cycle() method. This method first checks that a valid action has been issued (action_value > 0) and that the current time step falls within a valid operating window, as defined in the simulation schema, as seen in Listing 16.

```
def start_cycle(self, action_value: float):
    if not self.initiated and action_value > 0:
        start_time_step =
self.washing_machine_simulation.wm_start_time_step[self.time_step]
        end_time_step =
self.washing_machine_simulation.wm_end_time_step[self.time_step]

        if start_time_step <= self.time_step <= end_time_step:
            load_profile =
self.washing_machine_simulation.load_profile[self.time_step]
            self.__initiated = True
```

Listing 16 - Start Cycle Method Implementation

Once a cycle is triggered, the washing machine applies its load profile to the internal electricity consumption array, one time step at a time. This ensures the appliance behaves as a non-interruptible, sequential load (Listing 17).

```
for offset, load in enumerate(load_profile):
    step = self.time_step + offset
    if step < self.episode_tracker.episode_time_steps:
        self.__electricity_consumption[step] = load
```

Listing 17 - Electricity Consumption Variable Update

To preserve simulation consistency, the next_time_step() method includes logic to detect changes in the activation window between steps. If the current time step falls outside from the previously cached window, the model resets its internal initiation state, which safeguards it against unintended cycle restarts due to window shifts mid-episode, as seen in Listing 18.

```
if (prev_start != curr_start or prev_end != curr_end) and self.initiated:
    self.__initiated = False
```

Listing 18 - Activation Window Logic

This structure allows the washing machine to be treated as a flexible, schedulable appliance, while ensuring physically realistic behaviour by enforcing cycle atomicity. It also enables higher-level agents (e.g., RL controllers) to learn when to trigger cycles in response to time-varying electricity prices or grid signals.

Simulation Integration and Agent Control

After implementing the washing machine model, it was integrated into the simulation environment to enable meaningful interaction with RL agents. This was achieved by extending the Environment class to load and manage a WashingMachine instance based on configuration files provided in the input schema defined in Table 5 of Section 4.3.4.

During environment setup, the simulation loads a precomputed load profile for the appliance, using a helper method, as seen in Listing 19.

```
def _load_washing_machine(self, washing_machine_name, schema,
washing_machine_schema, episode_tracker):
    file_path = os.path.join(schema['root_directory'],
washing_machine_schema['washing_machine_energy_simulation'])
    df = pd.read_csv(file_path).iloc[

schema['simulation_start_time_step']:schema['simulation_end_time_step'] + 1
    ]
    simulation = WashingMachineSimulation(*df.values.T)
    return WashingMachine(
        washing_machine_simulation=simulation,
        episode_tracker=episode_tracker,
        name=washing_machine_name,
        seconds_per_time_step=schema['seconds_per_time_step'],
        random_seed=schema['random_seed']
    )
```

Listing 19 - Load Washing Machine Implementation

This function reads a CSV file containing the energy load profile and time window constraints for each episode, initialises a WashingMachineSimulation object, and passes it to the WashingMachine instance. This separation ensures clean encapsulation of appliance logic and simulation data.

Observation Pipeline and Schema Integration

To enable agents to make temporally aware decisions, the observation pipeline was modified to expose relevant washing machine metadata, specifically, the start and end time steps of its temporal availability windows. A purpose-built helper function populates this metadata during observation generation, inserting values into the agent’s observation space only when they align with keys explicitly defined in the simulation schema. This guarantees schema compliance and prevents leakage of extraneous or malformed data. During simulation, the following call integrates this logic into the broader observation system, as seen in Listing 20.

```
def update_washing_machine_observations(self, observations,
valid_observations, washing_machines):
    for wm in washing_machines:
        prefix = f"{wm.washing_machine_name}"
        if f"{prefix}_start_time_step" in valid_observations:
            observations[f"{prefix}_start_time_step"] =
wm.washing_machine_simulation.wm_start_time_step[self.time_step]
        if f"{prefix}_end_time_step" in valid_observations:
            observations[f"{prefix}_end_time_step"] =
wm.washing_machine_simulation.wm_end_time_step[self.time_step]
    return observations
```

Listing 20 - Update Washing Machine Observations

Observation Space Limits and Normalization

To preserve numerical stability during agent training and to promote generalizability across episodes, explicit upper and lower bounds were introduced for washing machine-specific observation fields to ensure the agent’s inputs remain within normalised ranges during training. These constraints are integrated into the environment's observation space definitions, with bounds set to reflect intra-day time steps ranging from 0 to 24. A sentinel value of -1 is reserved for cases where a valid schedule is undefined, following the already existent CityLearn conventions for fault toleration. This design ensures consistency across simulation episodes while preserving the expressive capacity of the observation space, as seen in Listing 21.

```
elif 'washing_machine' in key:
    for wm in self.washing_machines:
        if key == f'{wm.washing_machine_name}_start_time_step':
            low_limit[key] = -1
            high_limit[key] = 24
        elif f'{wm.washing_machine_name}_end_time_step' in key:
            low_limit[key] = -1
            high_limit[key] = 24
```

Listing 21 - Observation Space Limit implementation

Action Configuration and Load Profiles

The control interface for the washing machine within the simulation's action space adheres to a binary schema. A simple activation flag, "washing_machine": {"active": true}, suffices to trigger a full operational cycle. This abstraction closely mirrors real-world appliance usage patterns, where devices are either switched on or remain idle, and once initiated, follow a fixed consumption path to completion.

The washing machine's energy profile is modelled as a predefined list of power values spanning several time steps. A typical sequence might resemble [0.5, 0.7, 0.3, 0.2, 0.1], capturing the temporal structure of a standard washing cycle: an initial heating phase, followed by moderate energy use during washing, and concluding with a low-power spin or rinse phase. Importantly, the framework allows for substitution of this default profile with custom patterns derived from empirical measurements or user derived load patterns, thereby accommodating a wide array of appliance behaviours.

This appliance integration demonstrates CityLearn’s extensibility and architectural modularity. Through careful compartmentalisation of control logic, observability, schema integration, and simulation synchronisation, the platform can accommodate increasingly complex energy assets. The washing machine serves as a blueprint for future appliance models, enabling the study of flexible demand under realistic constraints and variable control scenarios.

5.4. Building-Specific Reward Functions

Following the integration of washing machines, the priority shifted to support building-specific reward functions. This extension enables heterogeneous agent objectives, supporting more nuanced simulations of cooperative, competitive, or independently optimised buildings. The result is a modular and extensible reward architecture that provides custom incentives per agent, without breaking compatibility with earlier single-reward configurations.

Schema-Driven Reward Assignment

To facilitate declarative configuration of building-specific rewards, the schema was extended to accept a dictionary under the reward_function key. This allows users to assign distinct reward types and parameters to each building, as illustrated in Listing 22.

```
"reward_function": {
  "type": {
    "Building_1": "citylearn.reward_function.ElectricVehiclesReward",
    "Building_2": "citylearn.reward_function.ComfortReward"
  },
  "attributes": {
    "Building_1": {
      "ev_weight": 0.5
    },
    "Building_2": {
      "comfort_band": [21, 25]
    }
  }
}
```

Listing 22 - Modular Reward Strategy: JSON-configurable specialisation

This declarative approach gives users flexibility to mix and match reward strategies, such as operational constraints, comfort metrics, or sustainability goals, by simply editing configuration files.

Architectural Refactoring for Reward Modularity

Upon loading the configuration, CityLearn dynamically instantiates the appropriate reward functions. The initialisation logic adapts automatically depending on whether a single global reward or building-specific rewards are specified. If the reward type is provided as a string, the system defaults to legacy mode, applying one reward function across all buildings. If instead a dictionary is provided, the system switches to the new multi-building mode, constructing a custom reward function for each building, as seen in Listing 23.

```
if isinstance(reward_function_type, str):
    reward_function = reward_function_constructor(None,
**reward_function_attributes)
else:
    # New mode: one reward function per building
    reward_function = MultiBuildingRewardFunction({
        b.name: reward_function_constructor(None,
**reward_function_attributes.get(b.name, {}))
        for b in buildings
    })
```

Listing 23 - Multi Building Reward Initialisation Implementation

MultiBuildingRewardFunction: Delegated Reward Logic

The MultiBuildingRewardFunction class is the core of this new modular architecture. While it adheres to the same interface as a traditional RewardFunction, its internal mechanics differ: it maintains a mapping between building names and their respective reward function instances. At runtime, the class receives a list of per-building observations and dispatches each to the corresponding reward function, as seen in Listing 24.

```
class MultiBuildingRewardFunction(RewardFunction):
    def __init__(self, reward_functions: Dict[str, RewardFunction]):
        self.reward_functions = reward_functions

    def calculate(self, observations: List[Mapping[str, Any]]) ->
List[float]:
        return [
            self.reward_functions[b_name].calculate([obs])[0]
            for b_name, obs in zip(self.reward_functions.keys(),
observations)]
```

Listing 24 - New MultiBuildingRewardFunction class Implementation

This structure ensures complete separation of reward semantics between buildings. One agent might be incentivised to minimise EV charging during peak hours, while another is tuned exclusively to maintain thermal comfort. This design enables the simulation of heterogenous behavioural patterns and facilitates experimentation with diverse incentive structures.

Simulation Integration and Execution

Despite the architectural changes, integration with CityLearn's primary simulation loop remains transparent. Within the step() method of CityLearnEnv, reward calculation occurs after observation gathering. Regardless of whether a single or multi-building reward strategy is used, the reward interface remains unchanged, as seen in Listing 25.

```
reward_observations = [b.observations(include_all=True, normalize=False) for
b in self.buildings]
reward = self.reward_function.calculate(observations=reward_observations)
self.__rewards.append(reward)
```

Listing 25 - Reward Calculation Left Unchanged

The application of polymorphism here ensures that downstream components, such as logging, agent learning algorithms, or evaluation scripts, do not need to be aware of whether a single or multiple reward functions are in play.

Summary and Implications

The adoption of building-specific reward functions transforms CityLearn from a globally incentivised simulation into a platform capable of modelling multi-agent environments with diverse, potentially conflicting objectives. By embedding reward logic at the building level and introducing a dispatch mechanism for reward routing, this enhancement enables differentiated agent behaviour aligned with sub-objective SO6, allowing for more realistic scenarios and nuanced policy experimentation. Crucially, it maintains backward compatibility, ensuring legacy configurations based on a single reward function remain functional. The update also support a range of coordination dynamics, including fairness, competition, and hybrid strategies, by enabling agents to pursue distinct goals. Additionally, schema-driven, declarative modelling of reward types and parameters improves both reproducibility and usability. Together, these changes lay the foundation for more sophisticated multi-agent simulations in CityLearn, enabling fine-grained behaviour modelling and exploration of incentive designs in distributed energy systems.

5.5. Incorporation of Stochasticity

Building on the flexibility introduced in the previous subsection, this dissertation enhances the realism and experimental efficiency of CityLearn by introducing stochastic variability in inputs, a feature designed to simulate real-world unpredictability while preserving system consistency. By injecting controlled noise into data streams such as energy loads and weather profiles, the framework facilitates resilient policy testing without requiring architectural modifications to downstream components. This balance of modularity, realism, and reproducibility makes it particularly well-suited for evaluating control strategies under realistic uncertainty.

5.5.1. Declarative Noise Configuration

The system introduces stochastic variability through a declarative configuration mechanism. Each building or EV entry in the simulation schema may include a noise_std parameter, representing the standard deviation of zero-mean Gaussian noise applied to its input streams. This modular design enables fine-grained control over uncertainty injection per agent, supporting experimental flexibility. For example, noise is specified directly in the schema, as shown in Listing 26.

```
"Building_1": {
    "energy_simulation": "Building_1.csv",
    "weather": "weather.csv",
    "carbon_intensity": "carbon_intensity.csv",
    "pricing": "pricing.csv",
    "noise_std": 17.0
}
```

Listing 26 – Example of Stochastic Deviation Schema Configuration in Buildings

This is also the case for the EV schema, as seen in Listing 27.

```
"Electric_Vehicle_1": {
      "include": true,
      "energy_simulation": "Electric_Vehicle_1.csv",
      "inactive_observations": [],
      "inactive_actions": [],
      "noise_std": 0.0,
}
```

Listing 27 – Example of Stochastic Deviation Schema Configuration in EVs

During environment initialisation, the noise_std parameter is extracted and passed to relevant data subsystems, enabling consistent, per-agent noise injection, as shown in Listing 28.

```
noise_std = building_schema.get('noise_std', 0.0)
…
electric_vehicle_noise_std = electric_vehicle_schema.get('noise_std', 0.0)
```

Listing 28 - Stochastic Parameter Initialisation

5.5.2. Hierarchical Building Initialisation

The hierarchical building initialisation process, governed by the_load_building() method, which constructs each building instance and supports schema-defined stochastic perturbations. It parses the optional noise_std parameter from the schema to determine the magnitude of variability. If not provided, noise_std defaults to 0, resulting in no noise injection. This value is propagated to downstream subsystems, ensuring consistent noise application across both energy and weather profiles (Listing 29 - Listing 31).

```
def _load_building(self, index: int, building_name: str, schema: dict,
                   episode_tracker: EpisodeTracker,
                   pv_sizing_data: pd.DataFrame,
                   battery_sizing_data: pd.DataFrame,
                   **kwargs) -> Building:
    building_schema = schema['buildings'][building_name]
    noise_std = building_schema.get('noise_std', 0.0)  # Configurable
variability
…
```

Listing 29 - Load Building Method Implementation

```
energy_simulation = EnergySimulation(
    **energy_simulation.to_dict('list'),
    seconds_per_time_step=schema['seconds_per_time_step'],
    noise_std=noise_std
)
```

Listing 30 - Energy Simulation Noise Injection

```
weather = pd.read_csv(
    os.path.join(schema['root_directory'], building_schema['weather']))
weather = Weather(
    **weather.to_dict('list'),
    noise_std=noise_std
)
```

Listing 31 - Weather Noise Injection

5.5.2. Weather Data Stochastic Processing

Following injection into the subsystems, key components, such as EnergySimulation, Weather, CarbonIntensity, and Pricing are extended to accept and apply noise directly during instantiation. The weather module applies noise independently to both observed and forecasted variables, as seen in Listing 32.

```
class Weather(TimeSeriesData):
    """Enhanced weather modeling with configurable uncertainty"""

    def __init__(self, outdoor_dry_bulb_temperature: Iterable[float],
                 outdoor_relative_humidity: Iterable[float],
                 diffuse_solar_irradiance: Iterable[float],
                 direct_solar_irradiance: Iterable[float],
                 noise_std: float = 0.0):
self.outdoor_dry_bulb_temperature +=
NoiseUtils.generate_gaussian_noise(self.outdoor_dry_bulb_temperature,
self.noise_std)
self.outdoor_dry_bulb_temperature_predicted_1 +=
NoiseUtils.generate_gaussian_noise(outdoor_dry_bulb_temperature_predicted_1,
self.noise_std)
```

Listing 32 - Weather Noise Processing Implementation

5.5.3. Noise Generation Utilities

All stochastic variability in CityLearn is handled by the NoiseUtils class, a centralised utility designed for physics-aware noise generation. This toolkit ensures that injected perturbations are both statistically meaningful and physically plausible, enabling realistic simulation of sensor noise and environmental disturbances, as seen in Listing 33.

```
class NoiseUtils:
    """Physics-aware noise generation toolkit"""

        def generate_gaussian_noise(input_data: Union[np.ndarray,
Iterable[float]],
                              noise_std: float) -> np.ndarray:
        """Generates Gaussian noise matching input shape.
        Args:
            input_data: numpy array or iterable (list/tuple of numbers)
            noise_std: Noise standard deviation (ignored if <= 0)

        Returns:
            Zero-mean noise array with same shape as input
        """
        arr = np.asarray(input_data)  # Handles both ndarray and Iterable
        if noise_std <= 0:
            return np.zeros(arr.shape)
        return np.random.normal(loc=0, scale=noise_std, size=arr.shape)
```

Listing 33 - Noise Logic Implementation

The generate_gaussian_noise() method applies zero-mean Gaussian perturbations with a user-defined standard deviation using NumPy’s random.normal(). The output noise is shaped to match the input data and can be directly added to time series such as weather conditions or energy profiles.

The utility is designed for extensibility, enabling future enhancements such as variable-specific noise distributions or temporally correlated perturbations. By centralizing noise generation, NoiseUtils improves consistency across modules and simplifies the implementation of experiments involving uncertainty.

5.5.4. Research Implications

The research implications of this stochastic framework are substantial. By enabling systematic noise injection, CityLearn facilitates control testing under sensor disturbances, supports sensitivity analyses for characterizing building archetypes, and enables anomaly detection research through synthetic fault generation. Planned enhancements, such as equipment-specific noise profiles, spatially correlated urban-scale models, and empirical uncertainty datasets, will further expand these capabilities.

In summary, CityLearn’s stochastic input subsystem represents a significant advancement toward experimental rigour and real-world applicability. Its declarative configuration, modular propagation, and physics-aware noise generation collectively support algorithm benchmarking, uncertainty-aware policy design, and synthetic anomaly testing. By integrating variability at the schema level while maintaining consistency across subsystems, the framework solidifies its position as a research-grade tool for urban energy systems, capable of probing the complexities of distributed control under realistic uncertainty.

5.6. Testing and Validation

The CityLearn framework was developed using a rigorous, multi-tiered testing methodology to ensure reliability, physical consistency, and correctness in urban energy simulations. Three primary testing approaches were implemented: unit testing of individual components, integration testing of subsystem interactions, and system-level validation for physical plausibility. This multi-layered strategy resulted in comprehensive verification across all framework aspects.

The testing process achieved 73% code coverage, across the core components examined in this dissertation, encompassing 67 individual tests, with particular emphasis on energy conservation principles and temporal consistency in multi-agent scenarios. A CI pipeline was established to enforce code quality, reproducibility, and test-driven development throughout the implementation lifecycle. Validation outcomes demonstrated the framework's resilience, achieving complete verification of core physics models and a marked improvement in bug detection efficiency.

5.6.1. Unit Testing with Pytest

Unit testing formed the foundation of the verification strategy, leveraging Pytest to validate core components in isolation. Tests focused on key physics-driven modules, including battery charge/discharge dynamics, power conversion efficiency, and stochastic weather perturbation behaviour. Mock objects and fixtures were employed to isolate components and eliminate dependencies, ensuring deterministic test behaviour.

A representative test case validated interpolation across the battery's power-dependent efficiency curve under variable loading conditions. The test confirmed proper efficiency calculations at minimum (0.83 at 0% load), medium (0.9 at 70% load), and interpolated (0.865 at 50% load) power levels, as seen in Listing 34. Edge case testing included validation of depth-of-discharge limits, ensuring batteries never exceeded their specified discharge thresholds even under extreme conditions, as seen in Listing 35.

```
def test_battery_efficiency_curve():
    """Verify power-dependent efficiency interpolation"""
    battery = Battery(
        power_efficiency_curve=[[0, 0.83], [0.7, 0.9], [1, 0.85]],
        nominal_power=10.0
    )

    # Test curve interpolation points
    assert battery.get_current_efficiency(0.0) == 0.83    # Minimum load
    assert battery.get_current_efficiency(7.0) == 0.9     # 70% load
    assert battery.get_current_efficiency(5.0) == pytest.approx(0.865)
```

Listing 34 - Test Battery Curve Test Implementation

```
def test_depth_of_discharge_limit():
    """Verify battery respects discharge limits"""
    battery = Battery(depth_of_discharge=0.8, capacity=100)
    battery.force_set_soc(0.25)  # 25% SOC (5% below 20% DoD limit)

    # Attempt deep discharge
    battery.charge(-1000)  # Try to fully discharge

    assert battery.soc >= 0.2 - 1e-6  # Never exceeds DoD limit
```

Listing 35 - Edge case Test Implementation

5.6.2. Test Reproducibility Considerations

To ensure reproducibility of tests involving stochastic processes, all noise generation components were configured to support deterministic seeding. During testing, fixed random seeds were applied to modules such as weather perturbation, carbon intensity, and stochastic pricing, ensuring consistent behaviour across test runs and enabling precise debugging. This design supports both statistical realism in runtime simulations and full reproducibility in validation contexts.

5.6.3. Integration Testing

Integration testing evaluated interactions among major system components, with particular emphasis on energy flow between EVs, charging infrastructure, and building subsystems. This phase bridges the gap between isolated unit functionality and system-level performance, ensuring components interact as expected when composed in real-world scenarios.

Beyond nominal functionality, integration tests were crafted to uncover subtle failures arising from interface mismatches, implicit assumptions, and accumulated numerical drift. The primary objective was to ensure that the physical laws governing energy conservation remained intact even as energy passed through multiple system layers, each introducing potential losses or delays.

A representative integration test demonstrated complete energy tracking during a charging session. The test verified that a 5kW charge command resulted in appropriate energy values at each system interface: 5.0kW drawn from the grid, 4.5kW after charger efficiency losses, and approximately 4.275kW stored in the battery after accounting for battery efficiency. These tests ensured proper energy conservation throughout the entire system, as seen in Listing 36.

```
def test_charger_ev_integration():
    """Verify complete energy accounting in charging cycle"""
    ev = ElectricVehicle(battery=Battery(capacity=50))
    charger = Charger(max_power=10, efficiency=0.9)

    # Test charging session
    charger.plug_car(ev)
    charger.update_soc(0.5)  # 5kW charge

    # Verify energy accounting
    assert ev.battery.energy_balance == pytest.approx(4.5 * 0.95)  # Charger
eff * battery eff
    assert charger.electricity_consumption == pytest.approx(5.0)    # Grid
draw
```

Listing 36 - Integration Test Code Implementation

5.6.4. Continuous Integration Pipeline

GitHub Actions was employed to implement a CI pipeline, establishing automated verification of all code changes. The CI pipeline performed three primary functions: (i) environment validation via dependency resolution and caching (see Listing 37), (ii) automated execution of the test suite with coverage enforcement (see Listing 38), and (iii) backward compatibility verification for legacy dataset support. The complete CI configuration is available in Appendix G.

The CI configuration enforced strict quality gates, including a minimum 90% test coverage threshold for all newly introduced code segments. This automated approach reduced regression detection time from days to minutes, catching most bugs before human code review. The CI system became particularly valuable when implementing temporal resolution improvements, rapidly identifying inconsistencies across modules operating at different temporal resolutions.

```
- name: Cache pip dependencies
        uses: actions/cache@v3
        with:
          path: ~/.cache/pip
          key: ${{ runner.os }}-pip-${{ hashFiles('**/requirements.txt') }}
          restore-keys: |
            ${{ runner.os }}-pip-
```

Listing 37 - Snippet of Python caching in GitHub Actions

```
- name: Run tests
  run: |
    pytest --cov=citylearn --cov-report=xml
    coverage report --fail-under=90
```

Listing 38 - Snippet depicting the CI/CD step for Running Tests

5.7. Conclusions

This chapter implemented five key architectural advancements based on the analysis and design from Chapter 4. Together, these innovations transform CityLearn into a sophisticated energy simulation platform. The implementation begins with a **Data Export** system featuring ISO 8601 timestamp support, which effectively resolves persistent data fragmentation issues.

Building on this foundation, the introduction of **Improved Temporal Granularity** through α-coefficient normalization enables physically consistent simulations across multiple timescales, from rapid sub-hourly fluctuations to seasonal patterns. This enhanced temporal precision proves particularly valuable for modelling modern energy assets such as high-speed EV chargers and responsive demand-side systems.

The platform’s capabilities are further advanced through two key features: the successful validation of **Modular Asset Integration** via the washing machine implementation, which establishes a reusable pattern for integrating diverse energy assets and the introduction of building-specific reward functions that transform multi-agent coordination through polymorphic dispatch mechanisms. Additionally, the **Incorporation of Stochastic** variability significantly enhances experimental rigour by enabling testing under realistic uncertainty conditions while maintaining essential reproducibility.

These architectural improvements underwent verification through 67 designed unit and integration tests, covering 73% of critical system components. The testing framework enforces fundamental physics principles, particularly energy conservation validation for battery systems and charging infrastructure.

Automated CI pipelines were created to improve the standards of CityLearn, namely automatically ensuring backward compatibility. The resulting platform achieves an optimal balance between academic precision and practical utility, capable of handling both basic deterministic scenarios and complex stochastic cases.

By advancing both the fidelity and flexibility of energy modelling, these architectural improvements lay the groundwork for exploring complex system behaviours and emergent dynamics. They also enable scaling across diverse use cases, from localised microgrid optimisation to large-scale urban simulations. Chapter 6 will demonstrate a Use Case done to show how these enhancements expand simulation capabilities in temporal accuracy, modelling flexibility, and experimental rigour, establishing CityLearn as both a powerful research tool and a blueprint for next-generation energy DTs.

6. Case Study

This case study serves as the empirical core of this dissertation, designed to evaluate the cumulative value of the platform enhancements proposed in previous chapters. It presents a high-resolution comparative experiment using the CityLearn simulation environment. The aim of this section is to examine the performance implications of sub-hourly control resolution, stochastic input variability, appliance-level load flexibility, and personalised, heterogeneous reward functions. Through a controlled comparison between a traditional baseline simulation and an enhanced framework that embodies the contributions of this work, this study illuminates how the CityLearn platform can evolve into a more realistic, adaptive, and policy-relevant tool for next-generation demand-side energy management.

6.1. Simulation Setup

The simulated environment models a REC consisting of one year of operational data on electricity demand and PV generation for 17 single-family residential buildings, each outfitted with a suite of Distributed Energy Resources (DERs), including rooftop PV panels, battery energy storage systems, and heat pump-based HVAC systems. Additionally, eight EVs are integrated, representing the dynamic and increasingly electrified nature of residential energy demand. Coordination and optimisation across these assets are achieved through a decentralised RL control framework, specifically employing the Soft Actor-Critic (SAC) algorithm for continuous action-space optimisation.

6.1.1. Common Parameters (Applicable to Both Scenarios)

All buildings are modelled after the Sierra Crest housing development in Fontana, California, which was used at the CityLearn Challenge 2022 Workshop [101]. The EV fleet is programmed using real-world stochastic charging and mobility data, ensuring behavioural fidelity to empirical usage patterns [100]. Both scenarios, baseline and enhanced, leverage the SAC RL algorithm [102] enabling consistent comparison of control performance across different model configurations.

6.1.2. Baseline Scenario (Original CityLearn Framework)

The baseline scenario reproduces the default CityLearn environment and serves as a reference point for evaluating the efficacy of the proposed enhancements. The simulation operates at a fixed hourly timestep and spans a full calendar year, totalling 8760 timesteps. A single, global reward function is employed, focusing exclusively on minimizing total electricity cost. Load modelling is deterministic and aggregated at the building level, abstracting away intra-building variability and appliance-specific dynamics. As such, this configuration reflects the prevailing paradigm in current energy management simulators, static, cost-centric, and coarse in temporal resolution.

6.1.3. Enhanced Scenario (Proposed Framework)

The enhanced scenario contains all methodological and architectural contributions introduced in this dissertation. It operates at a 15-minute timestep, simulating a three-month period. Crucially, this results in the same number of decision points (8,760), aligning with the annual hourly baseline for comparability. The choice of a 15-minute interval reflects a deliberate balance, offering improved temporal resolution and system realism without compromising computational performance.

Reward functions are diversified to reflect heterogeneous user needs and building characteristics. Buildings 1 through 3 employ the baseline IndependentSACReward, serving as internal controls. Building 4 utilises a ComfortReward function that prioritises indoor thermal comfort. Building 5 adopts a SolarPenaltyReward to encourage self-consumption of solar energy and discourage grid reliance. Building 6 combines both objectives in a composite SolarPenaltyAndComfortReward, representing a real-world trade-off between sustainability and comfort. The remaining buildings (7 through 17) revert to the baseline reward function, preserving statistical balance and allowing for meaningful isolation of the effects of personalised reward structures.

Stochastic variability is introduced by applying Gaussian noise (±10%) to both solar generation and household load profiles, modelling real-world environmental and behavioural uncertainty. Additionally, appliance-level control is integrated via washing machines modelled as deferrable loads. Usage patterns for these appliances are based on empirical data obtained from the Smart PDM project, specifically Use Case 11: Home Appliances [103], developed by Instituto Superior de Engenharia do Porto (ISEP), Cleanwatts (formerly named DPS), and Sonae. A dedicated adaptor layer, documented in Appendix H, transforms raw measurement data, from this project, (recorded on July 7, 2022) into models compatible with CityLearn.

6.2. Test and Evaluation Methodology

To assess the effectiveness of the methods and contributions outlined earlier in this thesis, a structured evaluation framework has been implemented. This involves defining a reference scenario (baseline) and a set of performance indicators that reflect various aspects of system behaviour.

6.2.1. Key Performance Indicators for Context Evaluation

These metrics, known as KPIs, offer a detailed view of system efficiency and behaviour, acting as critical tools for benchmarking against existing approaches in the field. They also support the objectives of reproducibility, clarity, and knowledge transfer. The analysis is based on the final run (episode) from the dataset, capturing the complete operational cycle.

Within the context of MultiMARL), an episode constitutes an uninterrupted sequence of interactions (states, decisions, and resulting feedback) from start to finish. For this study, each episode spans a full operational year, consisting of 8,760 hourly intervals (from $t = 0$ to $t = N-1$), providing a comprehensive testbed for performance analysis.

The eight KPIs defined below are designed to be minimised by the algorithms under evaluation:

1. **Electricity Consumption (D)** - Captures cumulative non-negative energy imported from the main grid by an individual dwelling throughout the episode. This is expressed in Equation 7, where only positive consumption values contribute to the total, thus filtering out self-generation feedback.

$$D = \sum_{t=0}^{N-1} \max(0, E_t^{dwelling}) \tag{7}$$

2. **Electricity Price (C)** – This indicator calculates the overall cost of electricity consumed during an episode. It multiplies the time-varying price signal P_t, by the dwelling's net energy draw at each timestep. In cases where the dwelling produces surplus energy, a correction factor is applied to reflect the reduced cost or income from excess generation (Equation 8).

$$C = \sum_{t=0}^{N-1} (E_t^{dwelling} * P_t) \ if \ (E_t^{dwelling} > 0) \ else \ (E_t^{dwelling} * P_t * 0.8) \tag{8}$$

3. **Carbon Emissions (G)** – Evaluates the carbon footprint of energy consumed based on emission intensity data (O_t) per unit of energy. The product of energy drawn and the respective emission factor is summed across all timesteps, offering insight into the environmental cost of electricity use (Equation 9).

$$G = \sum_{t=0}^{N-1} \max(0, E_t^{dwelling} * O_t) \tag{9}$$

4. **Zero Net Energy/Self-Consumption (Z)** – This metric tracks the net flow of electricity in and out of a dwelling. It is an indicator of self-sufficiency and the balance between grid imports and local renewable generation. A Z value nearing zero implies balanced consumption and production; negative values indicate surplus production exported to the grid (Equation 10).

$$Z = \sum_{t=0}^{N-1} (E_t^{dwelling}) \tag{10}$$

5. **Average Daily Peak (P)** – As shown in Equation 11, the average daily peak of electricity consumed (E_t^{EC}) during the last year of the simulation episode is measured, where d is the day of the year index.

$$P = \frac{\sum_{d=0}^{364} \sum_{t=0}^{23} (E_{24d+t}^{EC})}{365} \tag{11}$$

6. **Ramping (R)** – Quantifies the rate of change in energy consumption between successive time intervals at the Energy Community level. High ramping values indicate volatility, which can strain grid infrastructure. The KPI sums absolute differences in consecutive hourly consumption, as described in Equation 12.

$$\mathrm{R} = \sum_{t=0}^{N-1} \left| E_{t}^{EC} - E_{t-1}^{EC} \right| \tag{12}$$

7. **1 - Load Factor (1-L)** – This metric assesses the evenness of energy usage patterns over each month by comparing average and peak monthly demands. A lower 1−L value denotes more consistent and efficient energy use, while higher values signal erratic consumption. The formulation in Equation 13 expresses this as the inverse of a standard load factor, averaged annually.

$$1 - L = \frac{\left(\sum_{0}^{11} 1 - \frac{\frac{\left(\sum_{0}^{729} E_{730m+t}^{EC}\right)}{730}}{\max\left(E_{730m+t}^{EC}\right)} \right)}{12} \tag{13}$$

Among these KPIs, *P*, *R*, and 1−*L* are Energy Community level KPIs. On the other hand, *D*, *C*, *G*, and *Z* are dwelling-level KPIs. These are computed using the hourly net electricity consumption of individual buildings and are subsequently averaged and reported at the Energy Community level.

6.2.2. Normalization Methodology

To ensure analytical rigour, transparency, and fairness in comparing different scenarios, a structured normalization protocol was devised. One of the key considerations was maintaining a consistent analytical framework: therefore, both scenarios under evaluation were constructed to feature the same number of control actions, specifically, 8,760 timesteps, corresponding to hourly intervals across an entire year, even if their calendar durations differed due to specific modelling or data constraints.

To facilitate a meaningful comparison of performance, the KPIs were expressed as normalised values relative to a well-defined reference case. This reference scenario, or baseline, represents the original system operation prior to the application of any smart management strategies, in other words, it reflects the raw, unoptimised data without the influence of the advanced features and interventions discussed in this dissertation.

By comparing the KPIs of the enhanced scenarios against this baseline, it becomes possible to systematically identify and quantify the impacts, improvements, and potential trade-offs introduced by the deployment of the smart management strategies. This process of normalization is mathematically defined in Equation (14), shown below:

$$\mathrm{KPI}_{\mathrm{ratio}} = \frac{\mathrm{KPI}_{\mathrm{control}}}{\mathrm{KPI}_{\mathrm{baseline(no\ control)}}} \tag{14}$$

A KPI ratio below 1.0 indicates an improvement in performance for the enhanced scenario relative to the baseline, while a ratio above 1.0 suggests a deterioration in performance. This normalization framework provides a basis for comparing results across a wide range of performance dimensions and is particularly important when dealing with heterogeneous systems, for instance, when evaluating buildings with varying objectives, usage patterns, or reward structures.

6.3. Key Comparisons & Results

After running the simulations with both the baseline and enhanced simulation scenarios, they were compared using a comprehensive KPI suite through RL Cost Functions [110], as visualised in Table 8 through Table 10. For brevity reasons only the values of the first 10 buildings are presented in table form. The full data is present in Appendix I in Figure 38 through Figure 40. The metrics, spanning energy consumption, cost, emissions, flexibility, and comfort, were derived, as mentioned before, using timestep-normalised ratios to ensure analytic fairness and remove biases stemming from temporal resolution.

6.3.1. Energy and Carbon Performance

When comparing both scenarios, a reduction in carbon emissions error emerges at the district level, with total emissions dropping from 2.714 units in the baseline to just 0.718 units in the enhanced scenario, a 73.5% decrease. The outcome, captured visually in Table 8 through 10, and through Figures Figure 38 to Figure 40 of Appendix I, is especially pronounced in buildings with solar-centric reward shaping (notably Buildings 5 and 6), each demonstrating reductions exceeding 4.2 units. These results validate the capacity of reward differentiation to induce individualised, sustainability-aligned agent behaviour.

In parallel, electricity consumption exhibits a similarly decline in the cost function KPI, from 2.77 units to 0.714 units. The simultaneity of lower demand and lower emissions is a powerful indicator that the improvements are not simply shifting burdens but genuinely enhancing energy efficiency. The enhanced control architecture, featuring sub-hourly resolution, deferrable load integration, and stochastic variability, achieves systemic reductions through intelligent scheduling, real-time adaptation, and context-aware actuation.

Table 8 – Baseline Scenario Simulation KPI table

KPI	District	Building_1	Building_2	Building_3	Building_4	Building_5	Building_6	Building_7	Building_8	Building_9	Building_10
all_time_peak_average	0.333										
carbon_emissions_total	2.714	3.18	1.186	2.687	3.431	3.232	1.107	3.163	1.177	1.221	2.372
cost_total	2.990	3.371	1.229	1.32	4.051	3.438	1.188	4.575	1.229	1.303	2.577
daily_one_minus_load_factor_average	1.023										
daily_peak_average	0.002										
electricity_consumption_total	2.77	3.128	1.209	1.322	3.602	3.237	1.146	4.366	1.221	1.279	2.446
monthly_one_minus_load_factor_average	1.056										
zero_net_energy	-6.819	-0.132	1.005	1.282	16.456	-0.345	1.09	-0.288	1.187	1.225	4.021

Perhaps most significantly, the zero net energy (ZNE) performance marked advancement. Table 9 highlights the overall trend, and a closer look at Figure 39 in Appendix I shows that 11 of the 17 buildings (65% of buildings) transitioned toward more favourable ZNE states, indicative of increased solar self-sufficiency. District-wide, the ZNE metric improved by 3.688 units, reflecting a meaningful reduction in dependency on grid-delivered energy and aligning with the broader goals of RECs.

Table 9 – Enhanced Scenario Simulation KPI table

KPI	District	Building_1	Building_2	Building_3	Building_4	Building_5	Building_6	Building_7	Building_8	Building_9	Building_10
all_time_peak_average	0.006										
carbon_emissions_total	0.718	0.646	0.996	0.895	0.063	0.224	0.997	0.214	0.996	0.996	0.21
cost_total	0.71	0.649	0.996	0.895	0.047	0.178	0.997	0.195	0.996	0.996	0.177
daily_one_minus_load_factor_average	0.837										
electricity_consumption_total	0.714	0.55	0.996	0.895	0.057	0.195	0.997	0.134	0.996	0.996	0.188
monthly_one_minus_load_factor_average	0.836										
zero_net_energy	-3.131	0.267	1.004	1.103	7.389	2.357	1.003	2.796	1.003	1.003	3.419

6.3.2. Cost Implications

A direct consequence of energy reduction is seen in electricity cost, which dropped from 2.900 to 0.71 at the district level. The effect is especially visible in solar-optimised buildings, where autonomous strategies resulted in peak shaving and higher solar self-consumption, amplifying tariff-based savings.

While Table 10 summarises the overall trends, Figure 40 of Appendix I provides a more detailed breakdown, showing that 12 out of 17 buildings (71% of buildings) achieved a cost reduction exceeding 20%, with Building 12 achieving a cost savings of over 12.4 units. These results validate that higher-resolution control and reward differentiation are not merely theoretical improvements, they translate into substantial economic benefits.

Table 10 - Exported KPIs comparison between Baseline and Enhanced Simulations

KPI	District	Building_1	Building_2	Building_3	Building_4	Building_5	Building_6	Building_7	Building_8	Building_9	Building_10
all_time_peak_average	0.328										
carbon_emissions_total	1.996	2.234	0.146	0.042	4.251	3.008	1.014	0.181	0.222	0.146	0.145
cost_total	2.19	2.422	0.266	0.035	3.984	3.262	1.198	0.222	0.368	0.240	0.199

KPI	District	Building_1	Building_2	Building_3	Building_4	Building_5	Building_6	Building_7	Building_8	Building_9	Building_10
daily_one_minus_load_factor_average	0.186										
daily_peak_average	0.002										
electricity_consumption_total	2.056	2.178	0.220	0.267	3.335	3.042	1.149	0.225	0.264	0.258	0.165
zero_net_energy	-3.868	-0.399	0.001	0.279	9.865	-2.792	-3.054	0.184	0.222	-4.207	0.077

6.3.3. Flexibility and Peak Load Mitigation

The system’s ability to flexibly manage demand is evidenced by improvements in the daily load factor average, which decreased from 1.023 to 0.837, a relative improvement of 18.6%. This metric, which penalises sharp peaks and irregular usage, reflects the platform’s success in creating flatter, more predictable demand curves. The SAC-based controllers, operating at a 15-minute resolution, leveraged the increased actuation opportunities to shift deferrable loads to low-tariff, low-carbon periods without disrupting comfort or grid reliability.

Additionally, despite the changes introduced between the baseline and enhanced runs, key metrics, such as daily peak average and ramping intensity, remained largely unchanged.

6.3.4. Effect of Finer Time Step Granularity

Having seen the broader results, it’s valuable to examine the role of each individual feature in shaping the outcomes. One of the most impactful methodological advancements in the enhanced scenario is the reduction of the control timestep from one hour to fifteen minutes. It fundamentally alters the temporal structure of the simulation, dramatically enhancing the system's responsiveness, precision, and realism. Hourly control intervals often smooth over short-term fluctuations in load, solar generation, and user behaviour, masking important dynamics, as previously seen in Section 2.7.2. In contrast, a 15-minute resolution captures these transient events, allowing the control system to adapt more effectively to real-time changes. This isn’t merely a case of more frequent decisions, it represents a qualitative shift in how decisions are generated, contextualised, and implemented.

Empirical results strongly support this methodological shift. As illustrated in Figure 22 the electricity consumption profile with a 15-minute timestep (solid blue line) shows clearer, more stable control than the coarser 1-hour baseline (dashed orange line). The X-axis shows the 15-minute case timestamps for consistency, even though the 1-hour case would align a few days later under identical conditions. The enhanced control strategy results in reduced oscillations, improved load-following, and a smoother net consumption pattern throughout the observed period. Zooming in around midnight on July 3rd, 2024, the orange dashed line (without timestep adjustment) shows a sharp, erratic peak, while the solid blue line (with time_step ratio) remains smoother and more stable. This reflects the benefit of a finer control interval, giving the agent, in this 15-minute granularity case, four times more opportunities to adjust its actions. However, this doesn’t guarantee perfect decisions; occasional missteps still occur, not due to the timestep itself, but because of the agent’s intrinsic learning behaviour.

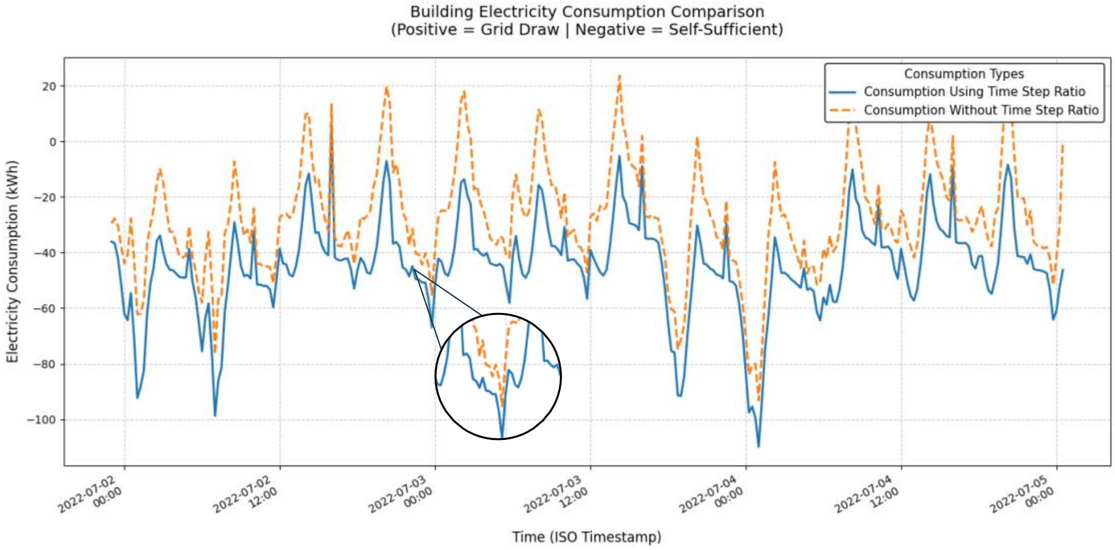


Figure 22 - Electricity Consumption Over Time With and Without Time Step Ratio

Consequently, the benefits of high-resolution control extend well beyond cosmetic improvements to the consumption curve. Sub-hourly decision intervals allow RL agents to anticipate load peaks, shift deferrable consumption, and synchronise demand with short-lived surges in solar generation. Crucially, these consumption reductions are not the product of crude load suppression, but of intelligent, context-aware control decisions enabled by a richer actuation window. With a finer actuation window, agents can continuously optimise battery use, HVAC settings, and EV charging in ways that are both environmentally responsive and user conscious.

These gains are particularly evident in buildings governed by complex, multi-objective reward functions, such as Building 6, which balances thermal comfort with solar self-consumption. In such settings, the ability to make nuanced, temporally sensitive decisions is essential. The finer resolution provides the temporal bandwidth necessary to negotiate competing priorities, offering an operational subtlety that is simply unattainable at hourly intervals.

Beyond traditional performance metrics, finer timestep granularity enhances systemic flexibility. Indicators such as load factor and ramping intensity reflect more stable and predictable consumption patterns. The system becomes less reactive and more anticipatory, flattening demand profiles in a manner that reduces grid stress while preserving comfort. This smoother behaviour is not only beneficial from an energy systems engineering perspective, but also a prerequisite for future improvements, namely the compatibility with real-time pricing schemes and dynamic tariffs, which increasingly operate on sub-hourly and even sub-minute schedules.

In summary, the transition to a 15-minute timestep represents far more than a technical refinement. It is a structural enhancement that elevates the CityLearn platform to a new standard of realism, control sophistication, and policy relevance. By aligning the simulation framework with the operational cadence of modern smart grids, this improvement unlocks previously inaccessible opportunities for demand-side optimisation. The evidence is clear: finer temporal granularity is not only a better approach, but also an essential enabler for the future of intelligent, decentralised energy management.

6.3.5. Effect of Building-Specific Rewards

Beyond temporal resolution, another key factor influencing agent performance is how rewards are defined and distributed. In RL environments like CityLearn, the reward structure significantly shapes agent behaviour, learning speed, and long-term performance. To investigate the impact of using building-specific reward functions, a customised reward assignment was implemented. Each building was allocated a distinct reward function, using the configuration detailed in Section 6.1.3. These included a range of formulations, from standard SAC-based objectives that emphasise energy cost and emissions minimization, to specialised metrics like the ComfortReward, which prioritises indoor climate stability. This design aims to better align individual building goals with broader community objectives. Figure 23 confirms that each building correctly initialised and operated under its designated reward function, enabling a fair evaluation of how differentiated incentives affect overall system dynamics and agent collaboration.

```
[Reward Setup] Building: Building_1, Type: citylearn.reward_function.IndependentSACReward, Attributes: {}
[Reward Setup] Building: Building_2, Type: citylearn.reward_function.IndependentSACReward, Attributes: {}
[Reward Setup] Building: Building_3, Type: citylearn.reward_function.IndependentSACReward, Attributes: {}
[Reward Setup] Building: Building_4, Type: citylearn.reward_function.ComfortReward, Attributes: {}
[Reward Setup] Building: Building_5, Type: citylearn.reward_function.SolarPenaltyReward, Attributes: {}
[Reward Setup] Building: Building_6, Type: citylearn.reward_function.SolarPenaltyAndComfortReward, Attributes: {}
[Reward Setup] Building: Building_7, Type: citylearn.reward_function.IndependentSACReward, Attributes: {}
[Reward Setup] Building: Building_8, Type: citylearn.reward_function.IndependentSACReward, Attributes: {}
[Reward Setup] Building: Building_9, Type: citylearn.reward_function.IndependentSACReward, Attributes: {}
[Reward Setup] Building: Building_10, Type: citylearn.reward_function.IndependentSACReward, Attributes: {}
[Reward Setup] Building: Building_11, Type: citylearn.reward_function.IndependentSACReward, Attributes: {}
[Reward Setup] Building: Building_12, Type: citylearn.reward_function.IndependentSACReward, Attributes: {}
[Reward Setup] Building: Building_13, Type: citylearn.reward_function.IndependentSACReward, Attributes: {}
[Reward Setup] Building: Building_14, Type: citylearn.reward_function.IndependentSACReward, Attributes: {}
[Reward Setup] Building: Building_15, Type: citylearn.reward_function.IndependentSACReward, Attributes: {}
[Reward Setup] Building: Building_16, Type: citylearn.reward_function.IndependentSACReward, Attributes: {}
[Reward Setup] Building: Building_17, Type: citylearn.reward_function.IndependentSACReward, Attributes: {}
```

Figure 23 - Multi-Reward Function Setup

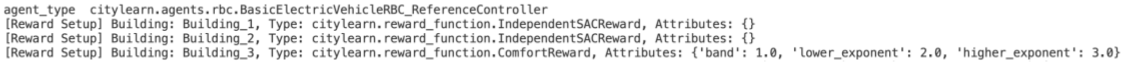
Table 11 presents the calculated rewards of a certain time step for each building under the configuration shown in Section 6.1.3. Despite the configuration's flexibility, this experiment revealed a key limitation: the dataset used lacked indoor temperature readings required by the ComfortReward class.

Table 11 - Rewards Calculated using the Setup of Section 6.1.3

Building	Custom Reward
Building_1	-1.7051
Building_2	-1.6418
Building_3	-0.0002
Building_4	NAN
Building_5	-0.5788
Building_6	NAN
Building_7	-0.5783
Building_8	-0.0002
Building_9	-0.4731
Building_10	-0.4095
Building_11	-1.3215
Building_12	-0.0002
Building_13	-1.3354
Building_14	-0.2869
Building_15	-0.002
Building_16	-0.6397
Building_17	-0.6889

As a result, while the system technically initialised the ComfortReward for the specified building, it failed to compute its values correctly due to missing input data, specifically, the indoor_dry_bulb_temperature field.

To validate that the custom reward functions, particularly those dependent on thermal comfort metrics, were implemented correctly, a separate experiment was run using a different dataset with full indoor climate information. This configuration is shown in Figure 24.



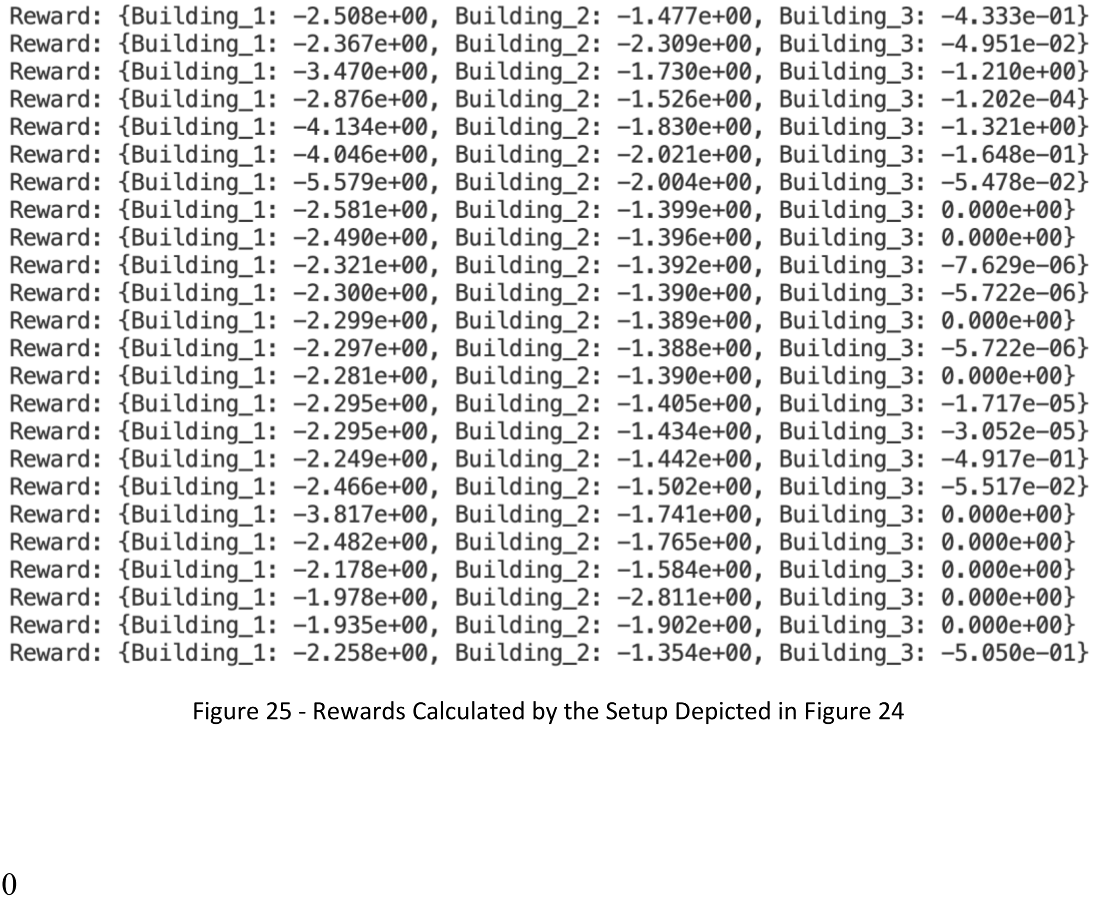
```
agent_type  citylearn.agents.rbc.BasicElectricVehicleRBC_ReferenceController
[Reward Setup] Building: Building_1, Type: citylearn.reward_function.IndependentSACReward, Attributes: {}
[Reward Setup] Building: Building_2, Type: citylearn.reward_function.IndependentSACReward, Attributes: {}
[Reward Setup] Building: Building_3, Type: citylearn.reward_function.ComfortReward, Attributes: {'band': 1.0, 'lower_exponent': 2.0, 'higher_exponent': 3.0}
```

Figure 24 - Multi-Reward Function Setup with the 2023 Dataset

This setup uses the 2023 dataset, which includes comprehensive environmental variables, allowing ComfortReward and other advanced reward structures to function as intended.

In Figure 25, one can see that the corresponding reward traces indicate proper differentiation between buildings based on their assigned reward functions. Notably, buildings with ComfortReward exhibit distinct reward dynamics reflective of their sensitivity to indoor temperature deviations. Meanwhile, those using IndependentSACReward follow smoother patterns more aligned with energy-centric objectives.

This evaluation confirms the ability of the CityLearn platform to support heterogeneous reward formulations at the building level, providing a modular foundation for multi-objective and agent-specific learning strategies. However, it also highlights the critical dependency between reward class design and dataset completeness, as it wasn't possible to confirm these reward functions with the database previously presented.



```
Reward: {Building_1: -2.508e+00, Building_2: -1.477e+00, Building_3: -4.333e-01}
Reward: {Building_1: -2.367e+00, Building_2: -2.309e+00, Building_3: -4.951e-02}
Reward: {Building_1: -3.470e+00, Building_2: -1.730e+00, Building_3: -1.210e+00}
Reward: {Building_1: -2.876e+00, Building_2: -1.526e+00, Building_3: -1.202e-04}
Reward: {Building_1: -4.134e+00, Building_2: -1.830e+00, Building_3: -1.321e+00}
Reward: {Building_1: -4.046e+00, Building_2: -2.021e+00, Building_3: -1.648e-01}
Reward: {Building_1: -5.579e+00, Building_2: -2.004e+00, Building_3: -5.478e-02}
Reward: {Building_1: -2.581e+00, Building_2: -1.399e+00, Building_3: 0.000e+00}
Reward: {Building_1: -2.490e+00, Building_2: -1.396e+00, Building_3: 0.000e+00}
Reward: {Building_1: -2.321e+00, Building_2: -1.392e+00, Building_3: -7.629e-06}
Reward: {Building_1: -2.300e+00, Building_2: -1.390e+00, Building_3: -5.722e-06}
Reward: {Building_1: -2.299e+00, Building_2: -1.389e+00, Building_3: 0.000e+00}
Reward: {Building_1: -2.297e+00, Building_2: -1.388e+00, Building_3: -5.722e-06}
Reward: {Building_1: -2.281e+00, Building_2: -1.390e+00, Building_3: 0.000e+00}
Reward: {Building_1: -2.295e+00, Building_2: -1.405e+00, Building_3: -1.717e-05}
Reward: {Building_1: -2.295e+00, Building_2: -1.434e+00, Building_3: -3.052e-05}
Reward: {Building_1: -2.249e+00, Building_2: -1.442e+00, Building_3: -4.917e-01}
Reward: {Building_1: -2.466e+00, Building_2: -1.502e+00, Building_3: -5.517e-02}
Reward: {Building_1: -3.817e+00, Building_2: -1.741e+00, Building_3: 0.000e+00}
Reward: {Building_1: -2.482e+00, Building_2: -1.765e+00, Building_3: 0.000e+00}
Reward: {Building_1: -2.178e+00, Building_2: -1.584e+00, Building_3: 0.000e+00}
Reward: {Building_1: -1.978e+00, Building_2: -2.811e+00, Building_3: 0.000e+00}
Reward: {Building_1: -1.935e+00, Building_2: -1.902e+00, Building_3: 0.000e+00}
Reward: {Building_1: -2.258e+00, Building_2: -1.354e+00, Building_3: -5.050e-01}
```

Figure 25 - Rewards Calculated by the Setup Depicted in Figure 24

6.3.6. Impact of Stochastic Variability

Building upon this deterministic analysis, the framework was further evaluated by introducing stochastic variability into the environment. Specifically, Gaussian noise (±10%) was applied to solar generation and household consumption profiles to emulate the uncertainty arising from weather patterns and user behaviour. This step simulates more realistic operational conditions, pushing the system closer to real-world scenarios. To assess the sensitivity of the enhanced framework to real-world unpredictability, we introduced stochastic variability through Gaussian noise (±10%) applied to solar generation and household consumption profiles. This simulates the inherent uncertainty present in energy systems due to weather conditions and user behaviour.

Figure 26 provides a comprehensive analysis of how such variability impacts indoor temperature data, a critical comfort and operational metric.

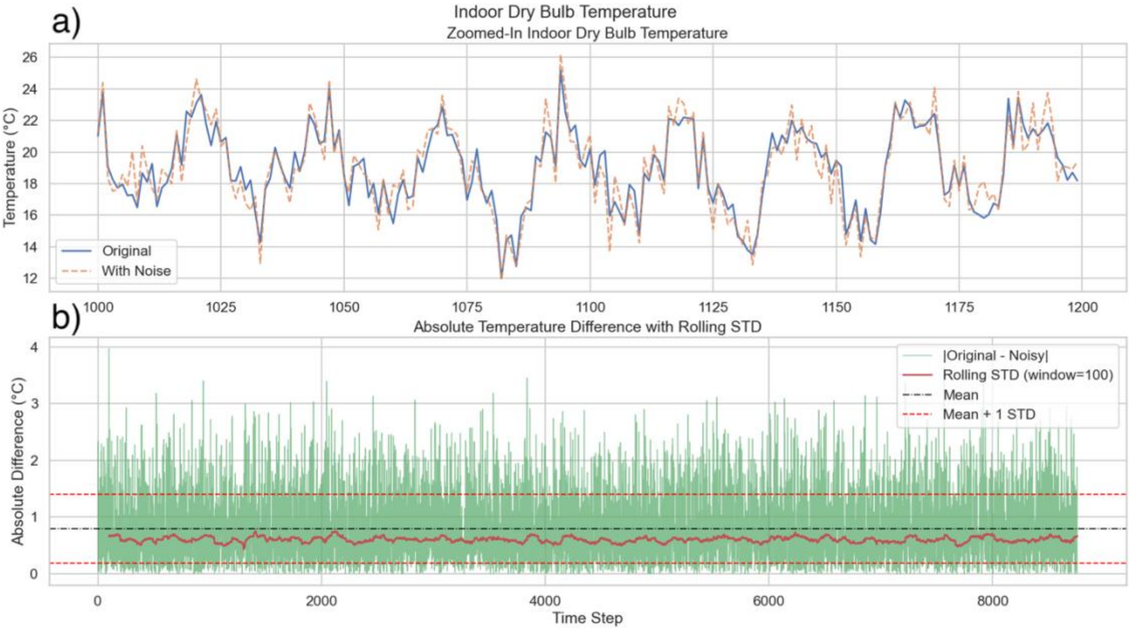


Figure 26 - Effect of Noise on Indoor Temperature Time Series and Absolute Differences

In Figure 26 a) A zoomed-in comparison of indoor dry bulb temperature shows the original (blue, solid) versus noisy (orange, dashed) time series. Despite fluctuations, the overall signal fidelity is preserved, suggesting efficiency of thermal control to moderate noise.

In Figure 26 b), the absolute differences between the original and noisy temperature signals are shown in green, remaining mostly below 1.5°C with only occasional spikes reaching 3–4°C. The rolling standard deviation remains stable over time, indicating that the variability introduced by the noise does not accumulate or intensify as the simulation progresses. The black dashed line represents the mean absolute difference, approximately 0.6–0.7°C, while the red dashed line marks the mean plus one standard deviation, serving as a threshold for more pronounced deviations. The fact that most values fall below this threshold suggests that the noise remains well-contained and statistically predictable. These results indicate that the stochastic input, while introducing minor fluctuations, does not meaningfully distort the underlying temperature signal, a conclusion that is further supported by the distributional analysis in Figure 27.

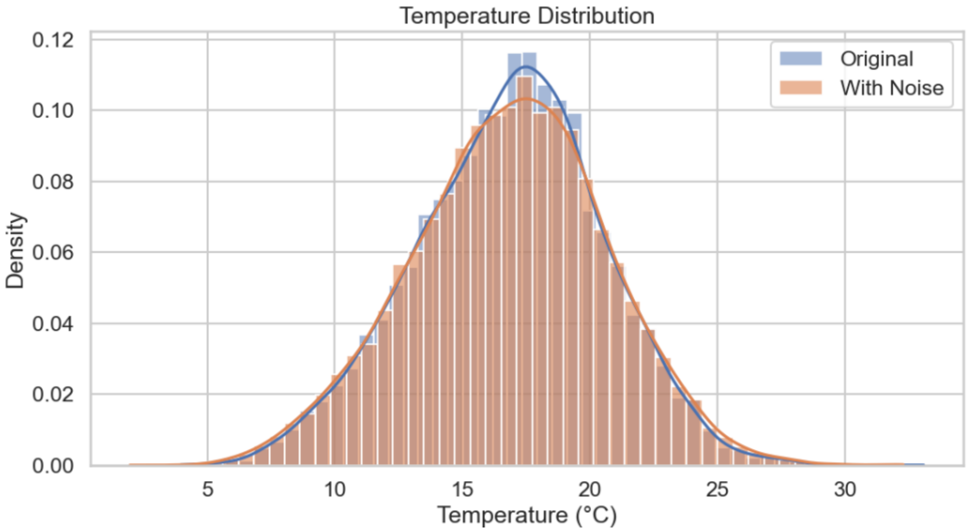


Figure 27 - Temperature Distribution Comparison Between Original and Noisy Data

This histogram with overlaid KDE curves [111] shows a near-identical distribution shape for both datasets. The Gaussian noise introduces variability without skewing the temperature profile substantially. The peak density remains centred around 17–18°C, affirming that the probabilistic structure of temperature control is retained despite the injected uncertainty.

What emerges from this use case is a form of adaptive inertia, where the control agents, trained without explicit exposure to stochastic inputs, exhibit an innate capacity to absorb and dampen short-term volatility. The RL framework, particularly in the configuration used here with soft actor-critic agents and sub-hourly actuation, demonstrates stability not through brute-force responsiveness but through anticipatory control patterns that internalise system dynamics over time.

This outcome carries direct implications for the operational deployment of learning-based controllers in residential energy communities. The simulation confirms that controller performance does not collapse under mild to moderate input perturbations, validating their application in physical systems where sensor noise, behavioural unpredictability, and weather dynamics are ever-present. More importantly, it justifies the practice of embedding stochastic variability into the training and benchmarking pipeline itself. By integrating noise as a core modelling element rather than a post hoc sensitivity test, future iterations of the framework can push toward even greater levels of generalization, aligning simulated behaviour more tightly with real-world operational complexity.

Continuing the evaluation, as established in Section 4.3.4, the performance of the heat pump is highly sensitive to ambient conditions, and its COP is intrinsically linked to the outdoor dry bulb temperature.

Introducing stochasticity to the temperature profile, simulating uncertainty or forecast error, directly alters the COP values at each time step. Because the COP serves as the divisor in the heat pump's input power calculation, as seen in Listing 39.

```
input_power = output_power / cop
```

Listing 39 - Input Power formula

A higher COP results in lower electricity consumption, while a lower COP increases demand. This sensitivity means even modest temperature noise can have a **nonlinear, cumulative impact** on energy flows. In our experiments, as seen in Figure 28, introducing stochastic variations, particularly at ±10% noise levels, led to observable shifts in heating electricity demand, demonstrating a **snowballing effect** through the system.

This can be interpreted as follows: a deviation in temperature reduces COP, increased power is needed to meet the same thermal demand, which means greater electricity consumption is recorded, which makes it for a potentially more costly or less efficient operation. This behaviour exemplifies the central role of temperature in shaping the heat pump's energy profile, making it one of the most vulnerable points of sensitivity in the simulation pipeline.

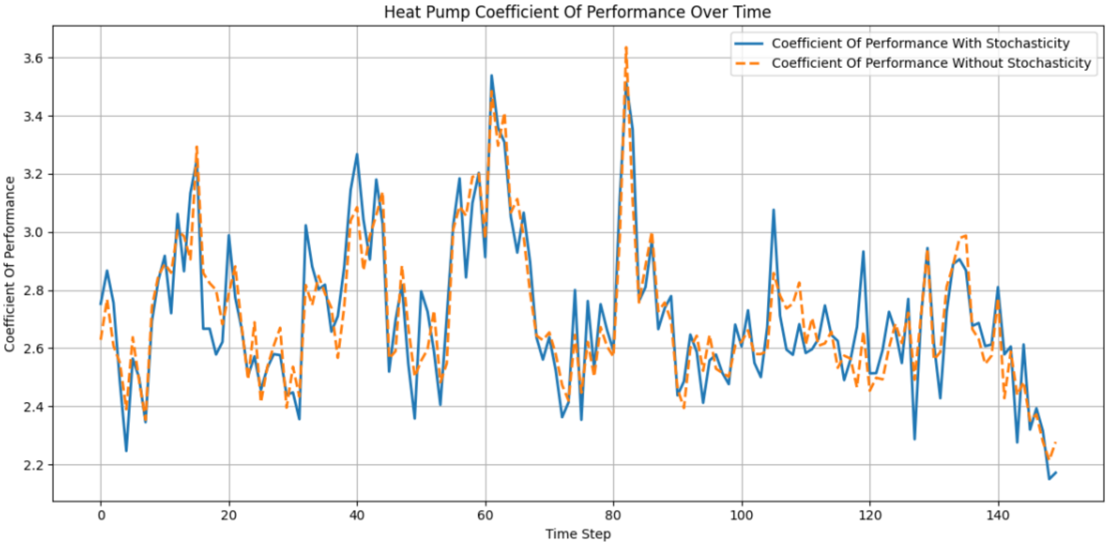


Figure 28 - Heat Pump Coefficient of Performance Over Time

It is important to contextualise this result. The application of stochasticity is intended to emulate uncertainty in predicted data, as would be the case in model predictive control or real-time forecasts. In such scenarios, the use of ±10% noise is arguably permissible, as it represents the range of uncertainty planners must contend with. However, when exact values are available (e.g., in retrospective analysis or validation), applying large stochastic variations becomes less appropriate. Instead, modest perturbations, on the order of observed real-world sensor variance, are more realistic and defensible.

Nevertheless, even within these bounds, the impact remains: COP variability propagates, influencing the entire energy simulation pipeline. As the COP fluctuates, so too does the derived input power, which in turn reshapes electricity consumption metrics, efficiency assessments, and even system-level optimisation outcomes. Thus, it can be concluded that temperature stochasticity, while conceptually small, has a disproportionately large influence not only on the performance of individual components like the heat pump, but also on the macro-scale results of the simulation.

This underlines a broader principle: data uncertainty must be treated as a first-class citizen in energy systems modelling. Even marginal changes in inputs can produce materially different outcomes, particularly when feedback loops and irregularities in the data are present.

6.3.7. Benefits of Appliance-Level Control (Washing Machines)

Building on this emphasis on precision and responsiveness in modelling, another avenue for enhancing simulation fidelity lies in the integration of fine-grained control strategies. One notable example is appliance-level control, specifically, the inclusion of washing machines as deferrable loads within the CityLearn framework. This extension introduces a new layer of flexibility and responsiveness to the demand side. To evaluate its impact, Figure 29 presents the first 300 simulation timesteps, contrasting the net electricity consumption (blue line) with a counterfactual scenario in which washing machines are excluded from active scheduling (orange dashed line).

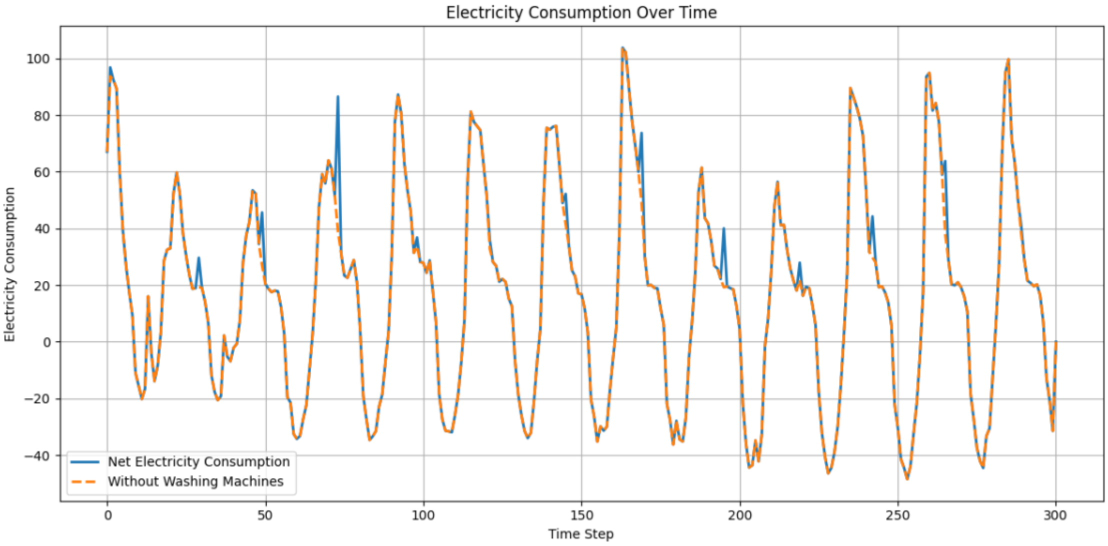


Figure 29 - Impact of the Washing Machine on the Electricity Consumption

Although the overall shape of the two curves in Figure 29 remains largely consistent, a closer inspection reveals transient deviations during specific time intervals. These deviations correspond to periods when the agent strategically advances or delays the operation of the washing machine to align with lower electricity prices or higher PV availability. While the magnitude of these shifts is relatively modest, given the comparatively low energy demand of washing machines versus HVAC systems or EVs, their contribution to peak shaving and intra-day load balancing is nontrivial over extended periods.

This form of localised, appliance-level flexibility enables subtle demand shifts that, when aggregated across households and multiple device types, yield noticeable improvements in grid stability and economic efficiency. The RL agent leverages this flexibility dynamically, without relying on static rule-based schedules or predefined operating windows. Instead, it responds adaptively to real-time pricing signals, weather forecasts, and system states, optimising decisions based on evolving conditions.

Notably, the inclusion of washing machine control does not introduce disruptive alterations to the aggregate load profile. Instead, it fine-tunes consumption behaviour, smoothing fluctuations and increasing temporal alignment with renewable generation. This demonstrates that even seemingly minor loads can meaningfully contribute to the overall performance of demand response strategies.

Moreover, the integration of such devices enhances the granularity of the control strategy, enabling a more distributed and resilient form of demand-side management. As highlighted in other sections of this work, while the influence of a single device may be incremental, the compounded impact across heterogeneous appliances and time scales substantiates the value of appliance-aware control. This supports a broader vision of intelligent energy systems where every controllable load plays a role in optimising performance at both the building and community levels.

6.3.8. Discussion & Insights

This case study has provided empirical validation for a reimagined simulation paradigm that aligns more closely with the operational, behavioural, and regulatory complexities of real-world residential energy systems. By integrating sub-hourly control granularity, stochastic variability, appliance-level load modulation, and heterogeneously defined reward functions within the CityLearn framework, this dissertation has demonstrated not only technical feasibility but also measurable performance gains across multiple axes of energy management.

The performance improvements can be attributed not to an indiscriminate increase in model complexity, but to specific, principled enhancements. For instance, sub-hourly actuation increased the temporal resolution of control, enabling agents to anticipate and capitalise on short-lived fluctuations in price and solar generation. However, without intelligent reward shaping or stochastic resilience, this alone would have led to overfitting or brittle policies. The results suggest a synergistic relationship between the features that were developed in this dissertation.

Moreover, this study proves that effective energy management isn’t just about cutting costs. It’s about balancing competing priorities like comfort, sustainability, and grid flexibility. Traditional systems treat energy optimisation as a single goal (usually cost reduction), but real-world households have different needs: some prioritise savings, others want perfect temperatures, and some focus on green energy. By allowing personalised AI-driven control, where each building can weigh these trade-offs differently, the simulation shows how communities can achieve broader success. Homes that value comfort may use slightly more energy, while solar-focused buildings reduce emissions but might pay higher short-term costs. One-size-fits-all energy strategies miss the bigger picture, but flexible, adaptive systems can harmonise these trade-offs for smarter, fairer, and more efficient grids.

Importantly, the incorporation of stochastic perturbations did not degrade performance in an uncontrolled manner. Instead, the system exhibited what may be termed “graceful degradation”, maintaining the distributional properties of KPIs and showing minimal skew in sensitive variables like indoor temperature or COP. This reveals that the learning agents, particularly those trained with the SAC algorithm, implicitly internalise environmental regularities in ways that render them able to moderate uncertainty. In operational contexts where perfect foresight is unattainable, this is a crucial attribute.

The experiment also underscores the value of appliance-level control, which, though modest in impact when considered in isolation, acts as a system-wide lubricant for load balancing and peak mitigation. Washing machine scheduling served as a proxy for a broader class of deferrable residential loads, suggesting that incremental controllability across appliances can, when scaled, become a cornerstone of demand response strategies.

Together, these findings support a shift toward control frameworks that are heterogeneous, high-resolution, and sensitive to uncertainty, better reflecting the messy, dynamic nature of real buildings and real occupants. The goal of the improved simulation architecture isn’t just to mimic complexity, but to harness it, enabling optimisation that naturally aligns with human behaviour, physical limits, and broader policy goals.

6.4. Conclusion

This dissertation presents a significant advancement in the state of the art for simulation-based demand-side energy management. By systematically incorporating sub-hourly resolution, agent-level reward heterogeneity, stochastic environmental inputs, and appliance-level flexibility into the CityLearn platform, this work transforms a conventionally idealised testbed into an extensible research environment capable of supporting the next generation of energy policy, market design, and control theory research.

The comparative case study provided insights on how these enhancements yield substantial, multidimensional benefits. Emissions and energy consumption were reduced, thermal comfort was preserved even under noisy conditions, and cost savings were delivered without sacrificing systemic resilience. More importantly, the modular architecture of the enhanced framework enables researchers to toggle complexity on a per-component basis, offering a valuable tool for sensitivity analysis, control benchmarking, and generalization studies.

In broader terms, this work advocates for a paradigm shift: from rigid, single-objective control strategies toward adaptive, personalised, and context-aware energy optimisation. Such a shift is not merely technical, it is foundational. As energy systems transition toward decentralisation, decarbonisation, and digitalization, the ability to simulate and manage heterogeneity becomes not a luxury, but a necessity.

Future research can build on this foundation by integrating dynamic pricing schemes, user-in-the-loop control, and multi-agent coordination across districts and utility interfaces. Additionally, by designing reward systems that also account for fairness, public health outcomes, and how people change their behaviour over time, these simulations can evolve from purely technical tools into valuable systems that help guide broader societal decisions.

In conclusion, this dissertation not only enhances the capabilities of an existing simulation platform, but it also offers a blueprint for how simulation itself can evolve to meet the profound challenges of climate, complexity, and computation in the 21st-century energy landscape.

7. Conclusions

After the analysis of the results from the use case, this chapter describes the conclusions of the dissertation, how the research questions were answered, how the objectives were accomplished, and the limitations and future directions of the proposed solution.

7.1. Summary

The digital transformation of energy systems is crucial for addressing sustainability and efficiency challenges within RECs. This dissertation explored how DT technology can enhance the simulation, planning, and optimisation of energy systems in such communities. To that end, the CityLearn platform was extended to incorporate features that align it more closely with DT principles, specifically improving its scalability, flexibility, and accuracy for community-level energy management.

Key enhancements include refined temporal granularity, modular integration of energy assets (such as household appliances and EVs), support for stochastic variability, and personalised building-level objectives. These improvements allow for more realistic and dynamic simulations of REC behaviour, contributing to better design and validation of EMSs.

The work conducted validates that adapting simulation tools with DT characteristics enables more accurate and meaningful insights into the operation of RECs. The enhanced CityLearn version provides a resilient framework for future research and deployment in real-world contexts, advancing both the theoretical understanding and practical capabilities in smart energy system modelling.

7.2. Accomplished Objectives

The work presented in this dissertation successfully fulfilled the main objective and all the originally defined specific sub-objectives, with each one contributing directly to the enhancement of the CityLearn platform and the advancement of DT simulation for RECs. Below is a breakdown of each objective and where its fulfilment can be found throughout the document:

- **SO1** – A detailed review of the current landscape of Renewable Energy Communities, Digital Twins, and energy simulation frameworks was conducted in Chapter 2 (Sections 2.3, 2.4, 2.5, 2.6 and 2.7), identifying technical gaps and opportunities to apply DT principles to REC simulations.

- **SO2** – A centralised export module for simulation results and KPIs was designed and implemented to support cross-project collaboration and external dashboard integration. The export system is detailed in Sections 4.3.6 and 5.1.

- **SO3** – The simulator was extended to support multiple temporal granularities (e.g., 15-minute steps), and the implications of finer time steps on computational performance and simulation fidelity were studied. This work is presented in Sections 2.7.2, 4.3.2, 5.2, and evaluated in the case study in Section 6.3.4.

- **SO4** – Asset models were adapted to support variable time resolutions without compromising behavioural accuracy. These changes enabled better reflection of real-world dynamics and are documented in Section 4.3.3 and 5.2, with supporting results discussed in Chapter 6.3.4.

- **SO5** – A modular simulation architecture was developed to integrate and manage diverse household assets such as EVs, refrigerators, and washing machines. These were modelled to interact within dynamic energy communities, and their behaviour is covered in Sections 2.7.3, 4.3.4 and 5.3, with results shown in 6.3.7.

- **SO6** – Individual building-level objectives (e.g., economic savings, maximizing self-consumption) were supported via custom reward functions, as detailed in Section 2.7.4, 4.3.3 and implemented in Section 5.4. The impact of these configurations was evaluated in Section 6.3.5.

- **SO7** – To replicate real-world uncertainty, stochastic elements were introduced at various layers of the simulation, including PV generation, temperature profiles, and appliance use. These mechanisms are described in Sections 4.3.5 and 5.5, and their effect is analysed in Section 6.3.6.

- **SO8** – The simulation framework was validated using unit and integration tests, Continuous Integration pipelines, and reproducibility safeguards. These are explained in detail in Section 5.6.

- **SO9** – Throughout the project, active collaboration was maintained with CityLearn's maintainers (University of Texas) to align extensions, contribute code upstream, and ensure compatibility.

- **SO10** – The electric vehicle model was redesigned to reflect real-world mobility more accurately, shifting from static vehicle-charger pairing to a setup involving mobile EVs and stationary chargers. This SO was addressed in Section 4.3.7.

In addition to its direct contributions to the OPEVA project, this work also formed the foundation of a scientific research paper:

• [Journal] Pina, R., Fonseca, T., ______ - "Smart Grid Optimisation in Renewable Energy Communities via Digital Twin Technology", to be published to Energy Scientific Journal 2025.

7.3. Research Questions and Objectives Achieved

This section revisits each of the research questions outlined in Chapter 1 and assesses how the dissertation's findings have addressed them. It also reflects on the objectives associated with each question and how they were fulfilled throughout the different phases of this work.

RQ1 – What limitations exist in current simulation tools for RECs, and how can a Digital Twin approach address them?

As explored in Chapter 2, many existing platforms either target building-level detail or lack the scalability required for community-wide analysis. Tools such as Energym, BOPTEST, and SAM offer high accuracy in isolated contexts but fall short in replicating the dynamic, interconnected nature of RECs. They typically lack real-time adaptability, user-oriented configuration, or integration with broader IoT ecosystems.

This research addressed these limitations by re-engineering CityLearn into a flexible DT-oriented framework. By adding support for stochasticity, fine-grained time steps, and modular energy assets, the resulting simulation environment brings it closer to a DT and enables testing across a wider spectrum of real-world conditions.

This question was addressed through Objectives **SO1**, **SO2**, and **SO4**, which guided the exploration of the literature, and the design specifications needed to elevate a traditional simulation into a dynamic and DT-aligned system.

RQ2 – How can temporal granularity be refined to better support the dynamics of REC simulation?

Temporal resolution is critical for energy simulations, as coarse time steps often fail to capture transient fluctuations in demand, storage, and generation. After the analysis of literature in section 2.7.2, chapter 4.3.2 introduced a configurable time-step mechanism, allowing the simulation to move beyond the default hourly basis of CityLearn and adapt to the needs of different case studies, e.g., sub-hourly resolution for EV charging or appliance cycling.

This was implemented while maintaining computational tractability and interoperability with RL agents, in Chapter 5.2. The results (Chapter 6.3.4) show that increased resolution allows for finer control and better predictive accuracy in EMS scenarios.

Objective **SO3** and **SO5** were central to this research question, resulting in substantial improvements to the platform’s temporal modelling capabilities.

RQ3 – What benefits arise from integrating diverse energy assets, such as household appliances and EVs, into simulation environments?

A true Digital Twin must reflect the full complexity of the physical system it represents, including the behavioural diversity of energy-consuming devices. In the original CityLearn framework, critical assets such as washing machines and refrigerators were either missing or abstracted. To bridge this gap, Section 4.3.4 introduced modular appliance-level integration, with washing machines serving as the primary case study.

A custom WashingMachine class was implemented to model realistic appliance behaviour using empirical power profiles, activation windows, and user-definable schedules. This allowed the simulation of demand-side management strategies such as cycle deferral and time-sensitive load shifting. As evaluated in Section 6.3.7, the addition of this asset contributed to peak load reduction and enabled finer control over building-level flexibility, particularly when coordinated through RL agents.

During development, it was also confirmed that refrigerators were indirectly supported through the HeatPump class hierarchy, which guided a design decision to reuse and extend existing modelling logic where appropriate. This maintained consistency and avoided redundancy, while clearly classifying fridges as passive loads, distinct from schedulable appliances.

Altogether, the integration of these diverse assets enriched the simulation environment with greater heterogeneity and realism, allowing more accurate assessments of energy strategies within RECs. This work fulfilled Objectives **SO3**, **SO5**, and **SO6**, and laid the groundwork for expanding the simulation framework to accommodate increasingly complex and personalised energy ecosystems.

RQ4 – To what extent does stochastic variability improve simulation realism and flexibility?

Incorporating stochastic behaviour into the simulation framework, covered in **Sections 4.3.5 and 5.5**, significantly increased the realism and efficiency of modelled scenarios. Noise injection mechanisms were implemented at multiple levels: PV generation, indoor temperature profiles, weather variables, and user-appliance interactions. These were configured via declarative YAML schemas, allowing for easy definition of variability ranges, probability distributions, and building-specific deviations (Listings Listing 26–Listing 33).

The impact of these additions was demonstrated in **Section 6.3.6**, where simulations with and without stochasticity were compared. The results showed that even moderate noise levels led to observable shifts in energy consumption patterns, thermal comfort metrics, and battery dispatch behaviour. Importantly, the variability surfaced edge cases and non-obvious interdependencies that deterministic simulations would have missed, such as misaligned control actions during sudden drops in solar availability or noise-induced oscillations in appliance scheduling.

This capability proved essential for stress-testing EMS algorithms, revealing how sensitive they are to real-world volatility. It also allowed for more nuanced benchmarking, since controller performance can now be evaluated not just on mean outcomes, but also on variance, stability, and worst-case behaviour.

Objective **SO5** guided the implementation of these stochastic features with a focus on modularity, allowing users to define variability without modifying core logic. This positions the framework for future work on probabilistic forecasting, efficient optimisation, and risk-aware control policies, key areas for real-world REC deployment.

Centralised Export

Although not explicitly defined as a research question of the project, the development of a centralised export mechanism, represented as SO2, emerged as a key enabler for cross-functional collaboration. This feature was particularly motivated by the need to support ongoing work within the OPEVA program, which aims to provide user-facing analytical tools built on top of simulation outputs. The project includes a separate dissertation [26], done by a colleague, focused on designing a frontend interface for data visualisation and exploration, leveraging the exported simulation data, examples of which are shown in Appendix I.

To facilitate this, the framework was extended with capabilities for structured export of KPIs, agent states, and simulation traces in standardised formats such as CSV and JSON (Listings Listing 7–Listing 11). The export module was designed to be modular and non-intrusive, automatically triggered during simulation without requiring post-processing scripts. This ensures full compatibility with external analytics pipelines, web-based dashboards, or machine learning workflows.

Beyond usability, this feature enhances reproducibility, supports multi-run aggregation, and lays the groundwork for integration with frontend visual tools, such as those being developed for OPEVA. By externalizing simulation data in structured form, researchers and practitioners can now interact with results in a more accessible and collaborative manner, aligning the simulation framework with open science principles and interdisciplinary use cases.

In sum, while not a research question per se, the centralised export system represents a strategic contribution to the long-term scalability and usability of the platform. It bridges the technical backend of simulation with the informational frontend of user interfaces, further solidifying the role of this work as a foundational layer in broader REC-related research and development efforts.

7.4. Limitations and Future Work

While the enhanced CityLearn framework represents substantial progress toward a functional DT environment for RECs, its current implementation still presents key limitations that constrain its practical deployment and broader usability. These limitations are especially pertinent when considering the real-world integration of DTs in energy systems that require ongoing adaptability, user accessibility, and seamless interoperability with diverse data ecosystems. The following areas have been identified for future improvement:

- **Real-Time Data Integration:** The current framework operates in an offline simulation environment and does not yet support streaming real-time data, limiting its ability to function as a live DT capable of dynamically interacting with physical systems. As noted in Section 4.2, the architecture is being developed with real-time integration in mind, with the adoption of protocols such as MQTT already underway. This upcoming milestone is critical, as it will enable the creation of a closed-loop system in which the digital model and the physical infrastructure continuously inform one another, fulfilling the fundamental DT paradigm [112].

- **Limited Support for Non-Expert Users:** Although the framework is inheritably technical, it currently requires considerable programming expertise and familiarity with reinforcement learning environments to operate, which poses a barrier to non-technical stakeholders such as community energy coordinators or REC administrators. To improve accessibility, future iterations should introduce intuitive user interfaces, low-code configuration workflows, and guided deployment processes. Promising developments are already emerging in parallel efforts, such as the OPEVA initiative, which has introduced a web-based dashboard aimed at improving usability.

- **Lack of Decision-Support Tools:** As discussed throughout Section 6, interpreting agent-driven outcomes remains complex, particularly for stakeholders without a background in optimisation or control systems. The addition of built-in decision-support features, such as strategy comparison visualisations, counterfactual scenario testing, and cost, benefit analysis tools, would allow users to assess the impact of different control strategies before real-world implementation, thereby making it a compelling argument for the adoption of the solution by economists and policymakers.

- **Interoperability with External Platforms:** Despite its strengths as a simulation tool, CityLearn currently lacks standardised interfaces for integration with operational technologies such as SCADA systems [113], Home Energy Management Systems (HEMS) [114], and microgrid controllers [115]. Incorporating support for industry-standard protocols (e.g., OpenADR [116], IEEE 2030.5 [117]) and middleware solutions (e.g., FIWARE [118], Node-RED [119]) would enable the framework to operate as an active digital layer within existing energy infrastructures, a prerequisite for real-world pilot deployment in RECs.

- **Scalability and Performance:** While the framework already supports multi-building environments, scaling it to accommodate the complexity of full REC or district-level deployments will lead to computational bottlenecks. Throughout this dissertation, decentralisation has been emphasised as a foundational design principle. Extending this approach to the simulation architecture, through the adoption of distributed computing methods, would enhance scalability and ensure the framework can perform effectively at larger geographic and operational scales, aligning with the vision of Smart Energy Cities.

Addressing these limitations will be key to transforming CityLearn from a sophisticated simulation tool into a robust, scalable, and user-oriented Digital Twin platform. With continued development, the framework will be well-positioned to support real-time optimisation, participatory decision-making, and operational resilience in Renewable Energy Communities.

7.5. Final Remarks

This dissertation marks a significant step toward bridging the gap between traditional simulation tools and practical DT applications for RECs, by enhancing the CityLearn platform with finer temporal resolution, modular asset integration, and stochastic modelling.

These developments are not just technical, but also redefine how RECs can be analysed, optimised, and governed. The improved framework supports more accurate modelling, user-driven objectives, and uncertainty-aware control, bringing the vision of intelligent, resilient, and participatory energy communities closer to reality.

Importantly, this work contributes directly to real-world initiatives like the EU's OPEVA project, underscoring its relevance beyond academia. With its focus on an open and extensible architecture, the platform is positioned for continued iterations and interdisciplinary collaboration.

As the energy sector moves toward greater decentralisation and digitalisation, the need for real-time, user-friendly tools becomes critical. This project provides a strong foundation for such tools, advancing not only simulation capabilities but also the practical deployment of DTs in shaping the energy systems of the future.

References

[1] 'OPEVA – OPtimization of Electric Vehicle Autonomy'. Accessed: Dec. 21, 2024. [Online]. Available: https://opeva.eu/

[2] Q. Hassan *et al.*, 'The renewable energy role in the global energy Transformations', *Renew. Energy Focus*, vol. 48, p. 100545, Mar. 2024, doi: 10.1016/j.ref.2024.100545.

[3] S. Ahmed, A. Ali, and A. D'Angola, 'A Review of Renewable Energy Communities: Concepts, Scope, Progress, Challenges, and Recommendations', *Sustainability*, vol. 16, no. 5, p. 1749, Feb. 2024, doi: 10.3390/su16051749.

[4] *COMMISSION STAFF WORKING DOCUMENT IMPACT ASSESSMENT Accompanying the document Proposal for a Directive of the European Parliament and of the Council on the promotion of the use of energy from renewable sources (recast)*. 2016. Accessed: Dec. 22, 2024. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex:52016SC0418

[5] 'EnR-Study-RenewableEnergyCommunities.pdf'. Accessed: Dec. 22, 2024. [Online]. Available: https://enr-network.org/wp-content/uploads/EnR-Study-RenewableEnergyCommunities.pdf

[6] J.-V. Andrei, A. Zaharia, G. Graci, and L. Chivu, 'Energy transition or energy diversification? Assessing the complexity of energy ecosystem towards transiting a climate neutral society', *Environ. Sci. Pollut. Res.*, vol. 30, no. 50, pp. 108477–108511, Sep. 2023, doi: 10.1007/s11356-023-30031-8.

[7] K. Nweye, B. Liu, P. Stone, and Z. Nagy, 'Real-world challenges for multi-agent reinforcement learning in grid-interactive buildings', *Energy AI*, vol. 10, p. 100202, Nov. 2022, doi: 10.1016/j.egyai.2022.100202.

[8] T. Luan, 'A Comprehensive Review of Simulation Technology: Development, Methods, Applications, Challenges and Future Trends', *Int. J. Emerg. Technol. Adv. Appl.*, vol. 1, no. 5, pp. 9–14, Jun. 2024, doi: 10.62677/IJETAA.2405119.

[9] P. Ladosz, L. Weng, M. Kim, and H. Oh, 'Exploration in deep reinforcement learning: A survey', *Inf. Fusion*, vol. 85, pp. 1–22, Sep. 2022, doi: 10.1016/j.inffus.2022.03.003.

[10] A. Rasheed, O. San, and T. Kvamsdal, 'Digital Twin: Values, Challenges and Enablers From a Modeling Perspective', *IEEE Access*, vol. 8, pp. 21980–22012, 2020, doi: 10.1109/ACCESS.2020.2970143.

[11] S. Mihai *et al.*, 'Digital Twins: A Survey on Enabling Technologies, Challenges, Trends and Future Prospects', *IEEE Commun. Surv. Tutor.*, vol. 24, no. 4, pp. 2255–2291, 2022, doi: 10.1109/COMST.2022.3208773.

[12] P. Scharnhorst *et al.*, 'Energym: A Building Model Library for Controller Benchmarking', *Appl. Sci.*, vol. 11, no. 8, p. 3518, Apr. 2021, doi: 10.3390/app11083518.

[13] D. Blum *et al.*, 'Building optimization testing framework (BOPTEST) for simulation-based benchmarking of control strategies in buildings', *J. Build. Perform. Simul.*, vol. 14, no. 5, pp. 586–610, Sep. 2021, doi: 10.1080/19401493.2021.1986574.

[14] D. B. Crawley *et al.*, 'EnergyPlus: creating a new-generation building energy simulation program', *Energy Build.*, vol. 33, no. 4, pp. 319–331, Apr. 2001, doi: 10.1016/S0378-7788(00)00114-6.

[15] P. Fritzson, *Principles of Object Oriented Modeling and Simulation with Modelica 3.3: A Cyber-Physical Approach*, 1st ed. Wiley, 2014. doi: 10.1002/9781118989166.

[16] N. Blair *et al.*, 'System Advisor Model, SAM 2014.1.14: General Description', NREL/TP--6A20-61019, 1126294, Feb. 2014. doi: 10.2172/1126294.

[17] R. Buyya and M. Murshed, 'GridSim: a toolkit for the modeling and simulation of distributed resource management and scheduling for Grid computing', *Concurr. Comput. Pract. Exp.*, vol. 14, no. 13–15, pp. 1175–1220, Nov. 2002, doi: 10.1002/cpe.710.

[18] M. A. Cuesta, T. Castillo-Calzadilla, and C. E. Borges, 'A critical analysis on hybrid renewable energy modeling tools: An emerging opportunity to include social indicators to optimise systems in small communities', *Renew. Sustain. Energy Rev.*, vol. 122, p. 109691, Apr. 2020, doi: 10.1016/j.rser.2019.109691.

[19] U. Ur Rehman, P. Faria, L. Gomes, and Z. Vale, 'Future of energy management systems in smart cities: A systematic literature review', *Sustain. Cities Soc.*, vol. 96, p. 104720, Sep. 2023, doi: 10.1016/j.scs.2023.104720.

[20] M. Mostafa and S. Lazarova-Molnar, 'Enhancing Reliability of Energy Systems with Digital Twins: Challenges and Opportunities', presented at the The 8th International Conference on System Reliability and Safety (ICSRS 2024), Sicily, Italy, Nov. 2024.

[21] I. Antonopoulos *et al.*, 'Artificial intelligence and machine learning approaches to energy demand-side response: A systematic review', *Renew. Sustain. Energy Rev.*, vol. 130, p. 109899, Sep. 2020, doi: 10.1016/j.rser.2020.109899.

[22] F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee, 'Digital Twin in Industry: State-of-the-Art', *IEEE Trans. Ind. Inform.*, vol. 15, no. 4, pp. 2405–2415, Apr. 2019, doi: 10.1109/TII.2018.2873186.

[23] K. Nweye *et al.*, 'CityLearn v2: energy-flexible, resilient, occupant-centric, and carbon-aware management of grid-interactive communities', *J. Build. Perform. Simul.*, pp. 1–22, Nov. 2024, doi: 10.1080/19401493.2024.2418813.

[24] M. Attaran and B. G. Celik, 'Digital Twin: Benefits, use cases, challenges, and opportunities', *Decis. Anal. J.*, vol. 6, p. 100165, Mar. 2023, doi: 10.1016/j.dajour.2023.100165.

[25] I. David, G. Shao, C. Gomes, D. Tilbury, and B. Zarkout, 'Interoperability of Digital Twins: Challenges, Success Factors, and Future Research Directions', in *Leveraging Applications of Formal Methods, Verification and Validation. Application Areas*, T. Margaria and B. Steffen, Eds., Cham: Springer Nature Switzerland, 2025, pp. 27–46. doi: 10.1007/978-3-031-75390-9_3.

[26] J. Oliveira, 'Desenvolvimento de Interface Gráfica para um Simulador de Comunidades de Energia', Instituto Superior de Engenharia do Porto, Porto, 2025.

[27] J. vom Brocke, A. Hevner, and A. Maedche, 'Introduction to Design Science Research', 2020, pp. 1–13. doi: 10.1007/978-3-030-46781-4_1.

[28] 'Regulation - 2022/868 - EN - EUR-Lex'. Accessed: Dec. 21, 2024. [Online]. Available: https://eur-lex.europa.eu/eli/reg/2022/868/oj/eng

[29] 'Despacho n.º 11171/2020 | DR'. Accessed: Dec. 21, 2024. [Online]. Available: https://diariodarepublica.pt/dr/detalhe/despacho/11171-2020-148327696

[30] 'IEEE Code of Ethics'. Accessed: Dec. 21, 2024. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html

[31] R. B. Grady and D. L. Caswell, *Software metrics: establishing a company-wide program*. USA: Prentice-Hall, Inc., 1987.

[32] M. A. Russo, D. Carvalho, N. Martins, and A. Monteiro, ‘Forecasting the inevitable: A review on the impacts of climate change on renewable energy resources’, *Sustain. Energy Technol. Assess.*, vol. 52, p. 102283, Aug. 2022, doi: 10.1016/j.seta.2022.102283.

[33] L. S. Paraschiv and S. Paraschiv, ‘Contribution of renewable energy (hydro, wind, solar and biomass) to decarbonization and transformation of the electricity generation sector for sustainable development’, *Energy Rep.*, vol. 9, pp. 535–544, Sep. 2023, doi: 10.1016/j.egyr.2023.07.024.

[34] W. L. Filho *et al.*, ‘Prosumers and sustainable development: An international assessment in the field of renewable energy’, *Sustain. Futur.*, vol. 7, p. 100158, Jun. 2024, doi: 10.1016/j.sftr.2024.100158.

[35] J. Parra-Domínguez, E. Sánchez, and Á. Ordóñez, ‘The Prosumer: A Systematic Review of the New Paradigm in Energy and Sustainable Development’, *Sustainability*, vol. 15, no. 13, p. 10552, Jul. 2023, doi: 10.3390/su151310552.

[36] S. A. Qadir, H. Al-Motairi, F. Tahir, and L. Al-Fagih, ‘Incentives and strategies for financing the renewable energy transition: A review’, *Energy Rep.*, vol. 7, pp. 3590–3606, Nov. 2021, doi: 10.1016/j.egyr.2021.06.041.

[37] European Environment Agency., *Energy prosumers in Europe: citizen participation in the energy transition.* LU: Publications Office, 2022. Accessed: Dec. 12, 2024. [Online]. Available: https://data.europa.eu/doi/10.2800/030218

[38] J. Schnidrig, A. Chuat, C. Terrier, F. Maréchal, and M. Margni, ‘Power to the People: On the Role of Districts in Decentralized Energy Systems’, *Energies*, vol. 17, no. 7, p. 1718, Apr. 2024, doi: 10.3390/en17071718.

[39] M. Mahmood, P. Chowdhury, R. Yeassin, M. Hasan, T. Ahmad, and N.-U.-R. Chowdhury, ‘Impacts of digitalization on smart grids, renewable energy, and demand response: An updated review of current applications’, *Energy Convers. Manag. X*, vol. 24, p. 100790, Oct. 2024, doi: 10.1016/j.ecmx.2024.100790.

[40] M. J. B. Kabeyi and O. A. Olanrewaju, ‘Smart grid technologies and application in the sustainable energy transition: a review’, *Int. J. Sustain. Energy*, vol. 42, no. 1, pp. 685–758, Dec. 2023, doi: 10.1080/14786451.2023.2222298.

[41] M. Khalid, ‘Smart grids and renewable energy systems: Perspectives and grid integration challenges’, *Energy Strategy Rev.*, vol. 51, p. 101299, Jan. 2024, doi: 10.1016/j.esr.2024.101299.

[42] M. H. Raza, Y. M. Rind, I. Javed, M. Zubair, M. Q. Mehmood, and Y. Massoud, ‘Smart Meters for Smart Energy: A Review of Business Intelligence Applications’, *IEEE Access*, vol. 11, pp. 120001–120022, 2023, doi: 10.1109/ACCESS.2023.3326724.

[43] R. A. Jabr and I. Džafić, ‘Distribution Management Systems for Smart Grid: Architecture, Work Flows, and Interoperability’, *J. Mod. Power Syst. Clean Energy*, vol. 10, no. 2, pp. 300–308, Mar. 2022, doi: 10.35833/MPCE.2021.000542.

[44] M. Moniruzzaman, A. Yassine, and R. Benlamri, ‘Blockchain and cooperative game theory for peer-to-peer energy trading in smart grids’, *Int. J. Electr. Power Energy Syst.*, vol. 151, p. 109111, Sep. 2023, doi: 10.1016/j.ijepes.2023.109111.

[45] K. M. Tan, T. S. Babu, V. K. Ramachandaramurthy, P. Kasinathan, S. G. Solanki, and S. K. Raveendran, ‘Empowering smart grid: A comprehensive review of

energy storage technology and application with renewable energy integration', *J. Energy Storage*, vol. 39, p. 102591, Jul. 2021, doi: 10.1016/j.est.2021.102591.
[46] L. Das, S. Munikoti, B. Natarajan, and B. Srinivasan, 'Measuring smart grid resilience: Methods, challenges and opportunities', *Renew. Sustain. Energy Rev.*, vol. 130, p. 109918, Sep. 2020, doi: 10.1016/j.rser.2020.109918.
[47] B. Raj, S. Kumar, S. Padhi, A. Sarkar, A. Mondal, and R. krithi, 'Brownout Based Blackout Avoidance Strategies in Smart Grids', *IEEE Trans. Sustain. Comput.*, vol. PP, pp. 1–1, Aug. 2020, doi: 10.1109/TSUSC.2020.3014077.
[48] A. Achnib and B. Amghar, 'Optimal Control-Based Energy Management in a Real Smart Grid', in *2023 IEEE 3rd International Conference on Smart Technologies for Power, Energy and Control (STPEC)*, Dec. 2023, pp. 1–6. doi: 10.1109/STPEC59253.2023.10431289.
[49] S. Ahmed and A. M. Măgurean, 'Renewable Energy Communities: Towards a new sustainable model of energy production and sharing', *Energy Strategy Rev.*, vol. 55, p. 101522, Sep. 2024, doi: 10.1016/j.esr.2024.101522.
[50] European Commission. Joint Research Centre., *Energy communities and energy poverty: the role of energy communities in alleviating energy poverty.* LU: Publications Office, 2023. Accessed: Dec. 12, 2024. [Online]. Available: https://data.europa.eu/doi/10.2760/389514
[51] R. Mishra, B. K. R. Naik, R. D. Raut, and S. K. Paul, 'Circular economy principles in community energy initiatives through stakeholder perspectives', *Sustain. Prod. Consum.*, vol. 33, pp. 256–270, Sep. 2022, doi: 10.1016/j.spc.2022.07.001.
[52] B. Lormeteau, 'Regulatory framework of collective self-consumption operations', 2022, pp. 110–125. doi: 10.4324/9781003257547-9.
[53] 'THE 17 GOALS | Sustainable Development'. Accessed: Dec. 12, 2024. [Online]. Available: https://sdgs.un.org/goals
[54] M. M. Mundu, S. N. Nnamchi, J. I. Sempewo, and D. E. Uti, 'Simulation modeling for energy systems analysis: a critical review', *Energy Inform.*, vol. 7, no. 1, p. 75, Aug. 2024, doi: 10.1186/s42162-024-00374-8.
[55] M. Soori, B. Arezoo, and R. Dastres, 'Virtual manufacturing in Industry 4.0: A review', *Data Sci. Manag.*, vol. 7, no. 1, pp. 47–63, Mar. 2024, doi: 10.1016/j.dsm.2023.10.006.
[56] M. S. Es-haghi, C. Anitescu, and T. Rabczuk, 'Methods for enabling real-time analysis in digital twins: A literature review', *Comput. Struct.*, vol. 297, p. 107342, Jul. 2024, doi: 10.1016/j.compstruc.2024.107342.
[57] Q. Hassan, S. Algburi, A. Z. Sameen, H. M. Salman, and M. Jaszczur, 'A review of hybrid renewable energy systems: Solar and wind-powered solutions: Challenges, opportunities, and policy implications', *Results Eng.*, vol. 20, p. 101621, Dec. 2023, doi: 10.1016/j.rineng.2023.101621.
[58] S. Djebali, G. Guerard, and I. Taleb, 'Survey and insights on digital twins design and smart grid's applications', *Future Gener. Comput. Syst.*, vol. 153, pp. 234–248, Apr. 2024, doi: 10.1016/j.future.2023.11.033.
[59] X. Liu *et al.*, 'A systematic review of digital twin about physical entities, virtual models, twin data, and applications', *Adv. Eng. Inform.*, vol. 55, p. 101876, Jan. 2023, doi: 10.1016/j.aei.2023.101876.
[60] 'References — CityLearn 2.3.0 documentation'. Accessed: Jun. 17, 2025. [Online]. Available: https://www.citylearn.net/references.html

[61] D. Deltetto, ‘Data-driven coordinated building cluster energy management to enhance energy efficiency, comfort and grid stability’, laurea, Politecnico di Torino, 2020. Accessed: Jun. 17, 2025. [Online]. Available: https://webthesis.biblio.polito.it/16358/

[62] ‘Gymnasium Documentation’. Accessed: May 19, 2025. [Online]. Available: https://gymnasium.farama.org/introduction/basic_usage.html

[63] R. S. Sutton and A. G. Barto, ‘Reinforcement Learning: An Introduction’.

[64] L. Cattafesta, C. Bahr, and J. Mathew, ‘Fundamentals of Wind-Tunnel Design’, 2010. doi: 10.1002/9780470686652.eae532.

[65] M. Grieves, *Virtually Perfect: Driving Innovative and Lean Products through Product Lifecycle Management*. 2011.

[66] ‘What Is a Digital Twin?’, NVIDIA. Accessed: Jun. 17, 2025. [Online]. Available: https://www.nvidia.com/en-sg/glossary/digital-twin/

[67] F. Tao *et al.*, ‘Five-dimension digital twin model and its ten applications’, *Jisuanji Jicheng Zhizao XitongComputer Integr. Manuf. Syst. CIMS*, vol. 25, pp. 1–18, Jan. 2019, doi: 10.13196/j.cims.2019.01.001.

[68] D. Adamenko, S. Kunnen, and A. Nagarajah, ‘Digital Twin and Product Lifecycle Management: What Is the Difference?’, in *Product Lifecycle Management Enabling Smart X*, vol. 594, F. Nyffenegger, J. Ríos, L. Rivest, and A. Bouras, Eds., in IFIP Advances in Information and Communication Technology, vol. 594. , Cham: Springer International Publishing, 2020, pp. 150–162. doi: 10.1007/978-3-030-62807-9_13.

[69] L. Sun, A. Pei, X. Qi, S. Cao, R. Yang, and X. Liu, ‘Dynamic Analysis of Digital Twin System Based on Five-Dimensional Model’, *J. Phys. Conf. Ser.*, vol. 1486, p. 072038, Apr. 2020, doi: 10.1088/1742-6596/1486/7/072038.

[70] M. Bouali, M. A. Sammuneh, R. E. Meouche, F. Ababsa, B. Salavati, and F. Viguier, ‘Digital Twin Architectures for Railway Infrastructure’, in *Innovations in Smart Cities Applications Volume 7*, M. Ben Ahmed, A. A. Boudhir, R. El Meouche, and İ. R. Karaș, Eds., Cham: Springer Nature Switzerland, 2024, pp. 213–223. doi: 10.1007/978-3-031-54376-0_19.

[71] M. J. Page *et al.*, ‘Updating guidance for reporting systematic reviews: development of the PRISMA 2020 statement’, *J. Clin. Epidemiol.*, vol. 134, pp. 103–112, Jun. 2021, doi: 10.1016/j.jclinepi.2021.02.003.

[72] M. J. Page *et al.*, ‘The PRISMA 2020 statement: an updated guideline for reporting systematic reviews’, *Syst. Rev.*, vol. 10, no. 1, p. 89, Dec. 2021, doi: 10.1186/s13643-021-01626-4.

[73] A. Aghazadeh Ardebili, M. Zappatore, A. I. H. A. Ramadan, A. Longo, and A. Ficarella, ‘Digital Twins of smart energy systems: a systematic literature review on enablers, design, management and computational challenges’, *Energy Inform.*, vol. 7, no. 1, p. 94, Oct. 2024, doi: 10.1186/s42162-024-00385-5.

[74] C. Qian, Y. Guo, A. Hussaini, A. Musa, A. Sai, and W. Yu, ‘A New Layer Structure of Cyber-Physical Systems under the Era of Digital Twin’, *ACM Trans. Internet Technol.*, p. 3674974, Jun. 2024, doi: 10.1145/3674974.

[75] K. O. Adu-Kankam and L. M. Camarinha-Matos, ‘Collaborative Digital Twins: The Case of the Energy Communities’, *SN Comput. Sci.*, vol. 4, no. 5, p. 664, Aug. 2023, doi: 10.1007/s42979-023-02050-2.

[76] G. Steindl, M. Stagl, L. Kasper, W. Kastner, and R. Hofmann, ‘Generic Digital Twin Architecture for Industrial Energy Systems’, *Appl. Sci.*, vol. 10, no. 24, Art. no. 24, Jan. 2020, doi: 10.3390/app10248903.

[77] İ. F. Tepe and E. Irmak, ‘Optimizing real-time demand response in smart homes through fuzzy-based energy management and control system’, *Electr. Eng.*, Jul. 2024, doi: 10.1007/s00202-024-02613-3.

[78] E. Proedrou, ‘A Comprehensive Review of Residential Electricity Load Profile Models’, *IEEE Access*, vol. 9, pp. 12114–12133, 2021, doi: 10.1109/ACCESS.2021.3050074.

[79] ‘How much electricity do solar panels produce?’, National Energy Action (NEA). Accessed: Dec. 12, 2024. [Online]. Available: https://www.nea.org.uk/who-we-are/innovation-technical-evaluation/solarpv/how-much-electricity-solar-produce/

[80] E. Al-Ali *et al.*, ‘Solar Energy Production Forecasting Based on a Hybrid CNN-LSTM-Transformer Model’, *Mathematics*, vol. 11, p. 676, Jan. 2023, doi: 10.3390/math11030676.

[81] A. Podder, N. Roy, and H. Pota, ‘MPPT Methods for Solar PV Systems: A Critical Review Based on Tracking Nature’, *IET Renew. Power Gener.*, vol. 13, May 2019, doi: 10.1049/iet-rpg.2018.5946.

[82] O. Omoyele *et al.*, ‘Increasing the resolution of solar and wind time series for energy system modeling: A review’, *Renew. Sustain. Energy Rev.*, vol. 189, p. 113792, Jan. 2024, doi: 10.1016/j.rser.2023.113792.

[83] M. J. Mayer, ‘Effects of the meteorological data resolution and aggregation on the optimal design of photovoltaic power plants’, *Energy Convers. Manag.*, vol. 241, p. 114313, Aug. 2021, doi: 10.1016/j.enconman.2021.114313.

[84] M. Hoffmann, L. Kotzur, D. Stolten, and M. Robinius, ‘A Review on Time Series Aggregation Methods for Energy System Models’, *Energies*, vol. 13, no. 3, Art. no. 3, Jan. 2020, doi: 10.3390/en13030641.

[85] M. Hoffmann, J. Priesmann, L. Nolting, A. Praktiknjo, L. Kotzur, and D. Stolten, ‘Typical periods or typical time steps? A multi-model analysis to determine the optimal temporal aggregation for energy system models’, *Appl. Energy*, vol. 304, p. 117825, Dec. 2021, doi: 10.1016/j.apenergy.2021.117825.

[86] M. Hoffmann, L. Kotzur, and D. Stolten, ‘The Pareto-optimal temporal aggregation of energy system models’, *Appl. Energy*, vol. 315, p. 119029, Jun. 2022, doi: 10.1016/j.apenergy.2022.119029.

[87] M.-C. Popescu, V. Balas, L. Perescu-Popescu, and N. Mastorakis, ‘Multilayer perceptron and neural networks’, *WSEAS Trans. Circuits Syst.*, vol. 8, Jul. 2009.

[88] K. Roy, A. Ishmam, and K. A. Taher, ‘Demand Forecasting in Smart Grid Using Long Short-Term Memory’, Jul. 28, 2021, *arXiv*: arXiv:2107.13653. doi: 10.48550/arXiv.2107.13653.

[89] B. Yilmaz, ‘A scenario framework for electricity grid using Generative Adversarial Networks’, *Sustain. Energy Grids Netw.*, vol. 36, p. 101157, Dec. 2023, doi: 10.1016/j.segan.2023.101157.

[90] M. Meiser, B. Duppe, I. Zinnikus, and A. Anisimov, ‘VA-Creator—A Virtual Appliance Creator based on adaptive Neural Networks to generate synthetic power consumption patterns’, *Energy AI*, vol. 18, p. 100427, Dec. 2024, doi: 10.1016/j.egyai.2024.100427.

[91] S. Wilhelm, D. Jakob, J. Kasbauer, and D. Ahrens, ‘GeLaP: German Labeled Dataset for Power Consumption’, in *Proceedings of Sixth International Congress on*

Information and Communication Technology, X.-S. Yang, S. Sherratt, N. Dey, and A. Joshi, Eds., Singapore: Springer, 2022, pp. 21–33. doi: 10.1007/978-981-16-2377-6_5.

[92] ‘REFIT Smart Home dataset’. Loughborough University, Jun. 20, 2017. doi: 10.17028/rd.lboro.2070091.v1.

[93] C. Shin, E. Lee, J. Han, J. Yim, W. Rhee, and H. Lee, ‘The ENERTALK dataset, 15 Hz electricity consumption data from 22 houses in Korea’, *Sci. Data*, vol. 6, no. 1, p. 193, Oct. 2019, doi: 10.1038/s41597-019-0212-5.

[94] D. Villanueva, D. San-Facundo, E. Miguez-García, and A. Fernández-Otero, ‘Modeling and Simulation of Household Appliances Power Consumption’, *Appl. Sci.*, vol. 12, no. 7, Art. no. 7, Jan. 2022, doi: 10.3390/app12073689.

[95] B. Berg *et al.*, ‘Occupant-driven end use load models for demand response and flexibility service participation of residential grid-interactive buildings’, *J. Build. Eng.*, vol. 96, p. 110406, Nov. 2024, doi: 10.1016/j.jobe.2024.110406.

[96] S. S. Ghazimirsaeid, M. S. Jonban, M. W. Mudiyanselage, M. Marzband, J. L. R. Martinez, and A. Abusorrah, ‘Multi-agent-based energy management of multiple grid-connected green buildings’, *J. Build. Eng.*, vol. 74, p. 106866, Sep. 2023, doi: 10.1016/j.jobe.2023.106866.

[97] J. Jiang, K. Su, and Z. Lu, ‘Fully Decentralized Cooperative Multi-Agent Reinforcement Learning: A Survey’, Jan. 10, 2024, *arXiv*: arXiv:2401.04934. doi: 10.48550/arXiv.2401.04934.

[98] P. Wilk, N. Wang, and J. Li, ‘Multi-Agent Reinforcement Learning for Smart Community Energy Management’, *Energies*, vol. 17, no. 20, Art. no. 20, Jan. 2024, doi: 10.3390/en17205211.

[99] I. Jendoubi and F. Bouffard, ‘Multi-agent hierarchical reinforcement learning for energy management’, *Appl. Energy*, vol. 332, p. 120500, Feb. 2023, doi: 10.1016/j.apenergy.2022.120500.

[100] ‘Work Breakdown Structure (WBS) - Basic Principles | PMI’. Accessed: Dec. 22, 2024. [Online]. Available: https://www.pmi.org/learning/library/work-breakdown-structure-basic-principles-4883

[101] ‘The C4 Model for Software Architecture’, InfoQ. Accessed: May 28, 2025. [Online]. Available: https://www.infoq.com/articles/C4-architecture-model/

[102] T. Fonseca, L. Ferreira, B. Cabral, R. Severino, and I. Praça, ‘EnergAIze: Multi Agent Deep Deterministic Policy Gradient for Vehicle-to-Grid Energy Management’, in *2024 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, Sep. 2024, pp. 232–237. doi: 10.1109/SmartGridComm60555.2024.10738095.

[103] T. Fonseca *et al.*, ‘EVLearn: extending the cityLearn framework with electric vehicle simulation’, *Energy Inform.*, vol. 8, no. 1, p. 16, Feb. 2025, doi: 10.1186/s42162-024-00445-w.

[104] ‘11.8: Heat Engines and Refrigerators’, Physics LibreTexts. Accessed: Jun. 18, 2025. [Online]. Available: https://phys.libretexts.org/Bookshelves/Thermodynamics_and_Statistical_Mechanics/Heat_and_Thermodynamics_(Tatum)/11%3A_Heat_Engines/11.08%3A_Heat_Engines_and_Refrigerators

[105] M. A. Zehir and M. Bagriyanik, ‘Demand Side Management by controlling refrigerators and its effects on consumers’, *Energy Convers. Manag.*, vol. 64, pp. 238–244, Dec. 2012, doi: 10.1016/j.enconman.2012.05.012.

[106] S. Mohseni, R. Khalid, and A. C. Brent, ‘Stochastic, resilience-oriented optimal sizing of off-grid microgrids considering EV-charging demand response: An efficiency comparison of state-of-the-art metaheuristics’, *Appl. Energy*, vol. 341, p. 121007, Jul. 2023, doi: 10.1016/j.apenergy.2023.121007.

[107] ‘(PDF) Application of Machine Learning Tools for Long-Term Diagnostic Feature Data Segmentation’, *ResearchGate*, May 2025, doi: 10.3390/app12136766.

[108] S. Levine, A. Kumar, G. Tucker, and J. Fu, ‘Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems’, Nov. 01, 2020, *arXiv*: arXiv:2005.01643. doi: 10.48550/arXiv.2005.01643.

[109] ‘ISO - ISO 8601 — Date and time format’, ISO. Accessed: Jun. 18, 2025. [Online]. Available: https://www.iso.org/iso-8601-date-and-time-format.html

[110] ‘Cost Function — CityLearn 2.3.0 documentation’. Accessed: Jun. 08, 2025. [Online]. Available: https://www.citylearn.net/overview/cost_function.html

[111] Y.-C. Chen, ‘A tutorial on kernel density estimation and recent advances’, *Biostat. Epidemiol.*, vol. 1, no. 1, pp. 161–187, Jan. 2017, doi: 10.1080/24709360.2017.1396742.

[112] M. Grieves and J. Vickers, ‘Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems’, in *Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches*, F.-J. Kahlen, S. Flumerfelt, and A. Alves, Eds., Cham: Springer International Publishing, 2017, pp. 85–113. doi: 10.1007/978-3-319-38756-7_4.

[113] A. Daneels and W. Salter, ‘What is SCADA?’, 1999, Accessed: Jun. 26, 2025. [Online]. Available: https://cds.cern.ch/record/532624

[114] A. Raza, L. Jingzhao, Y. Ghadi, M. Adnan, and M. Ali, ‘Smart home energy management systems: Research challenges and survey’, *Alex. Eng. J.*, vol. 92, pp. 117–170, Apr. 2024, doi: 10.1016/j.aej.2024.02.033.

[115] M. Uddin, H. Mo, D. Dong, S. Elsawah, J. Zhu, and J. M. Guerrero, ‘Microgrids: A review, outstanding issues and future trends’, *Energy Strategy Rev.*, vol. 49, p. 101127, Sep. 2023, doi: 10.1016/j.esr.2023.101127.

[116] B. Nordman, L. Parker, A. K. Prakash, and M. A. Piette, ‘Transforming Demand Response using OpenADR 3.0’, 2024.

[117] ‘2030.5-2023 - IEEE Standard for Smart Energy Profile Application Protocol’. Accessed: Jun. 26, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/10785536

[118] ‘FIWARE - Open APIs for Open Minds’. Accessed: Jun. 26, 2025. [Online]. Available: https://www.fiware.org/

[119] ‘Low-code programming for event-driven applications : Node-RED’. Accessed: Jun. 26, 2025. [Online]. Available: https://nodered.org/

[120] ‘ScienceDirect.com | Science, health and medical journals, full text articles and books.’ Accessed: Dec. 13, 2024. [Online]. Available: https://www.sciencedirect.com/

[121] ‘MDPI | MDPI Journal List’. Accessed: Dec. 13, 2024. [Online]. Available: https://www.mdpi.com/about/journals

[122] ‘PRISMA 2020 flow diagram’, PRISMA statement. Accessed: Dec. 13, 2024. [Online]. Available: https://www.prisma-statement.org/prisma-2020-flow-diagram

[123] ‘AIcrowd | NeurIPS 2022: CityLearn Challenge | Challenges’, AIcrowd | NeurIPS 2022: CityLearn Challenge | Challenges. Accessed: Jun. 01, 2025. [Online]. Available: https://www.aicrowd.com/challenges/neurips-2022-citylearn-challenge

Appendix A – PRISMA Methodology used on Research Question 1

This systematic review explores what's the current state of the art on Digital Twins used on the domain of Renewable Energy Communities.

Research sources

The search was conducted primarily using the most influential journals and databases on sustainable energy. Amongst such journals and databases was the Association for Computing Machinery (ACM) Digital Library [222] and the Springer Nature [223].

Search Terms

To identify research relevant to this topic, the focus was set on "Scalable Digital Twin Frameworks for Renewable Energy Communities." Keywords and their associated synonyms were thoughtfully selected from authoritative sources to construct a precise and comprehensive search query. Table 12 highlights the core keywords used in this systematic review process:

Table 12 - Domain and Keywords defined for the scope

Domain	Keywords
Digital Twin	"Digital Twin" OR "Digital Twins"
Renewable Energy Communities	"renewable energy communities" OR "energy grids"
Household Appliances	"Electric Vehicle" OR "flexible energy assets" OR "home appliances" OR "smart appliances" OR "dishwasher" OR "dish washer" OR "washing machine" OR "clothes dryer" OR "fridge"

The previous process resulted in the following query: ("Digital Twin" OR "Digital Twins") AND ("renewable energy communities" OR "energy grids") AND ("Electric Vehicle" OR "flexible energy assets" OR "home appliances" OR "smart appliances" OR "dishwasher" OR "dish washer" OR "washing machine" OR "clothes dryer" OR "fridge")

To refine the search results, inclusion and exclusion criteria were defined. These criteria ensured that only relevant and high-quality studies were considered. Table 13 and Table 14 summarise the criteria:

Table 13 - Criteria for Inclusion of Papers used during PRISMA

Inclusion Criteria Number	Description
IC1	Papers published between 2019 and 2024 (no more than five years old).
IC2	Papers available in PDF format.
IC3	Papers written in English.

Table 14 - Criteria for Exclusion of Papers used during PRISMA

Exclusion Criteria Number	**Description**
EC1	Papers published before 2019.
EC2	Papers not available in PDF format.
EC3	Papers not written in English.
EC4	Papers not addressing the research topic.

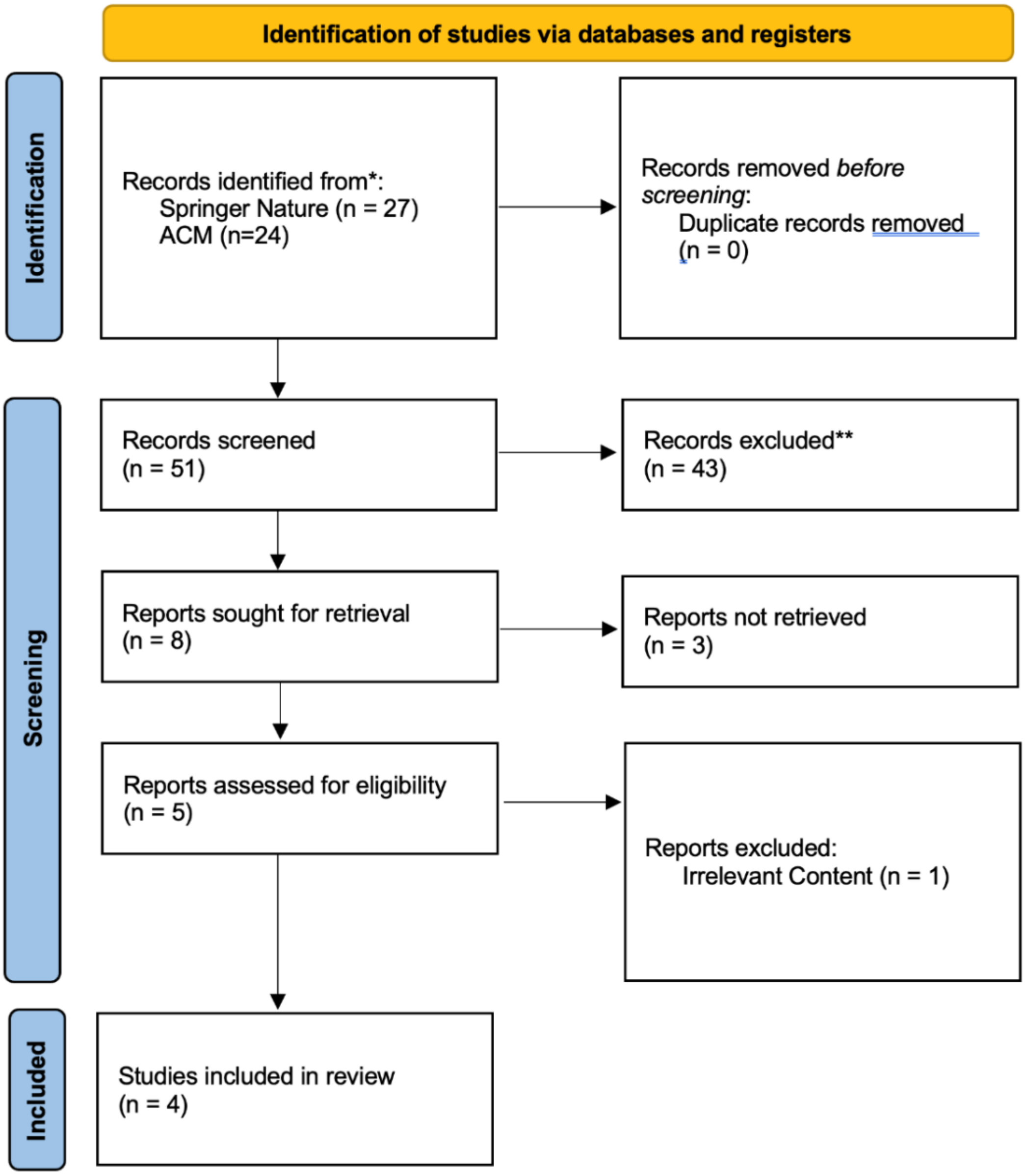


Figure 30 - Prisma 2020 Flow Diagram for Research Question 1 [91]

As seen in, the ACM Digital Library search was conducted using the above query and criteria. The initial search yielded 24 papers. Spring Nature came up with 27 papers. No duplicates were encountered. Titles and abstracts of the identified papers were screened, resulting in the exclusion of forty articles due to irrelevance. The remaining five papers underwent a full-text review, during which three were deemed unrelated to the research objectives. Ultimately, two papers were included in the systematic review.

Appendix B – PRISMA Methodology used on Research Question 3

This systematic review explores the integration of diverse energy assets, such as ESS and household appliances (e.g., refrigerators, dishwashers), within a scalable DT framework to simulate real-world energy consumption and behaviours.

Science Direct [120] and MDPI [121] served as the primary data source for collecting relevant research articles, due to them being aggregators of articles and papers that are open access and peer reviewed.

To identify studies pertinent to this topic, the scope of "Scalable Digital Twin for Energy Asset Modelling" was established. A range of keywords and their synonyms were derived from credible references to build the search query. Table 15 outlines the primary keywords used in the review process:

Table 15 - Domain and Keywords defined for the scope

Domain	Keywords
Synthetic Power Consumption	"synthetic power consumption"
Household Appliance Energy Modelling	"Modelling", "Household Appliances Power Consumption", "dishwasher", "washing machine", "dryer"
Demand Response and Smart Systems	"load models", "demand response", "residential grid-interactive buildings"

The previous process resulted in the following query: ("synthetic power consumption" AND "Neural Network") OR ("Modelling" AND "Household Appliances Power Consumption" AND ("dishwasher" OR "washing machine")) OR ("load models" AND "demand response" AND "residential grid-interactive buildings")

To refine the search results, inclusion and exclusion criteria were defined. These criteria ensured that only relevant and high-quality studies were considered. Table 16 and Table 17 summarise the criteria:

Table 16 - Criteria for Inclusion of papers used during PRISMA

Inclusion Criteria Number	Description
IC1	Papers published between 2020 and 2024 (no more than five years old).
IC2	Papers available in PDF format.
IC3	Papers written in English.

Table 17 - Criteria for Exclusion of Papers used during PRISMA

Exclusion Criteria Number	Description
EC1	Papers published before 2019.
EC2	Papers not available in PDF format.
EC3	Papers not written in English.
EC4	Papers not addressing the research topic.

The search was conducted on both platforms using the specified query and criteria. The initial results included six papers from ScienceDirect and eleven from MDPI, with no duplicates identified. Titles and abstracts of the retrieved papers were screened, leading to the exclusion of seven articles due to irrelevance.

The remaining ten papers underwent a detailed review. Of these, six were excluded because they did not align with the research objectives. For instance, some focused-on anomaly detection rather than evaluating sensor data methodologies, which is the primary aim of this study. Others explored broader topics, such as energy consumption patterns or household modelling, without addressing appliance-level analysis or real-world energy use cases involving sensor data.

One paper was excluded as it examined behavioural energy usage patterns specific to China. While this could have been categorised as irrelevant due to its lack of alignment with the focus on modelling energy assets, the added regional specificity further reduced its applicability to the broader context of this dissertation. Ultimately, two papers met the inclusion criteria and were selected for the systematic review. The flow diagram summarizing this process is presented in Figure 31.

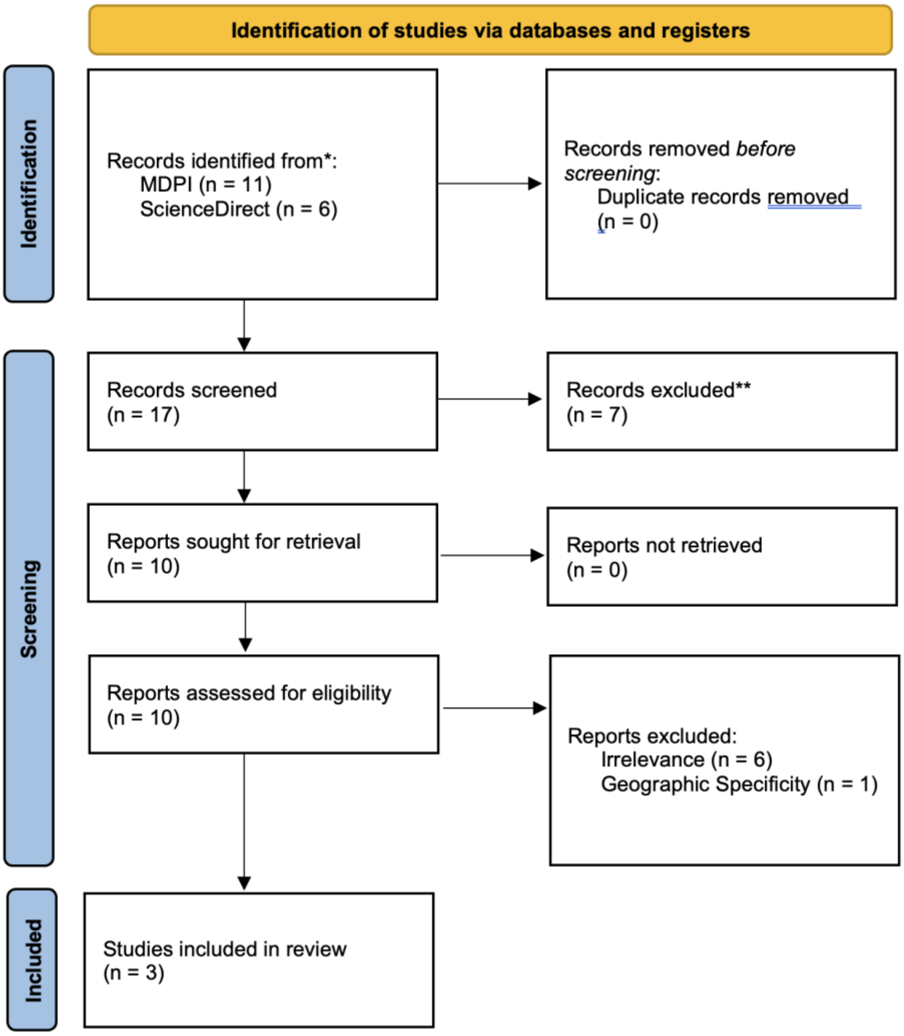


Figure 31 - Prisma 2020 Flow Diagram for Research Question 3 [122]

Appendix C – Self and Peer Reviewed Skills Evaluation

Table 18 - Self and Peer Assessment of Competencies

COMPETENCIES	Importance	Self-evaluation	Peer evaluation
Problem analysis and resolution	5	4	4
Lifelong learning	5	4	4
Teamwork and collaboration	5	4	4
Motivation for excellence	3	3	3
Adaptability and flexibility	4	4	4
Ethics and social responsibility	3	3	3
Proficiency in foreign languages	4	4	5
Technical skills in the specific area of expertise	5	4	4
Decision-making	4	3	4
Time management	5	3	3
Presenting	5	2	2
Ability to take risks	2	3	3
Oral communication	5	3	4
Written communication	3	4	4
Resilience	4	4	3
Active listening	4	3	3
Interpersonal relationships	2	3	4
Leadership	4	2	3
Creativity and innovation	3	2	3
Adaptation and flexibility	4	4	4
Critical thinking	4	3	3
Planning and organisation	5	4	4
Emotional management	4	4	4
Negotiation	4	2	3
Empathy	4	3	3
Assertiveness	3	3	4

ACTION PLAN

Most Developed Competencies

Proficiency in foreign languages Problem analysis and resolution Teamwork and collaboration

Competencies to Develop

Presenting Negotiation

For each competency to develop, define at least two actions:

Competency	Development actions
Presenting	**Attend a course:** Complete the "Complete Presentation Skills Masterclass for Every Occasion" course on Udemy by March 2025. **Practice with a group:** One week before the dissertation’s presentation, in July, rehearse in front of a group of friends to gather feedback and refine the presentation.
Negotiation	**Practice active listening techniques in meetings:** In each meeting, rephrase what the other party said before responding. Apply this technique in 80% of meetings and document the outcomes of at least 10 interactions per quarter. **Develop proposals with "win-win" scenarios:** In weekly project meetings, list at least two possible solutions that benefit both parties in conflict resolution.

Appendix D – Microsoft Project expanded insights

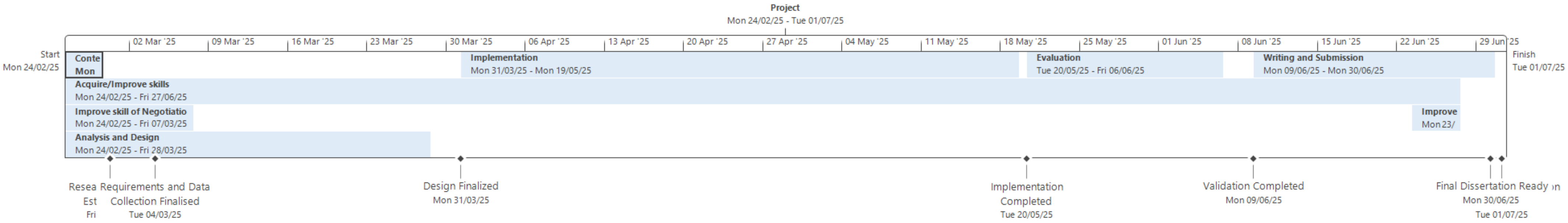
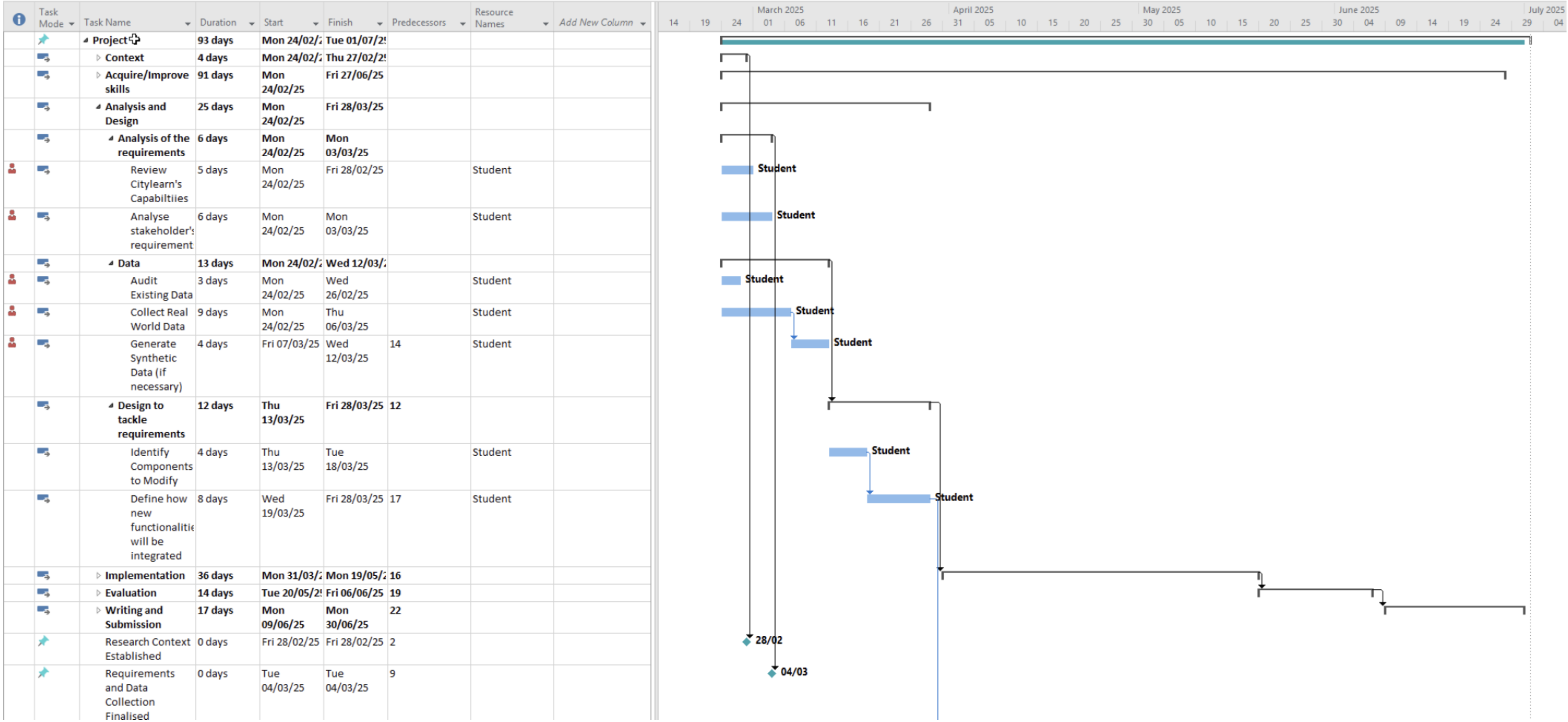


Figure 32 - Extended Project's Timeline

In Figure 32, the timeline does not provide sufficient detail to represent the entire scope of information. However, in Figure 33, "Context" is scheduled from 24/02/2025 to 27/02/2025. Simultaneously, the task "Improve Presentation Skills" runs from 23/06/2025 to 27/06/2025.



	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names	Add New Column
		Project	**93 days**	**Mon 24/02/2**	**Tue 01/07/2**			
		Context	**4 days**	**Mon 24/02/2**	**Thu 27/02/2**			
		Acquire/Improve skills	**91 days**	**Mon 24/02/25**	**Fri 27/06/25**			
		Analysis and Design	**25 days**	**Mon 24/02/25**	**Fri 28/03/25**			
		Analysis of the requirements	**6 days**	**Mon 24/02/25**	**Mon 03/03/25**			
		Review Citylearn's Capabiltiies	5 days	Mon 24/02/25	Fri 28/02/25		Student	
		Analyse stakeholder's requirement	6 days	Mon 24/02/25	Mon 03/03/25		Student	
		Data	**13 days**	**Mon 24/02/2**	**Wed 12/03/2**			
		Audit Existing Data	3 days	Mon 24/02/25	Wed 26/02/25		Student	
		Collect Real World Data	9 days	Mon 24/02/25	Thu 06/03/25		Student	
		Generate Synthetic Data (if necessary)	4 days	Fri 07/03/25	Wed 12/03/25	14	Student	
		Design to tackle requirements	**12 days**	**Thu 13/03/25**	**Fri 28/03/25**	**12**		
		Identify Components to Modify	4 days	Thu 13/03/25	Tue 18/03/25		Student	
		Define how new functionalitie will be integrated	8 days	Wed 19/03/25	Fri 28/03/25	17	Student	
		Implementation	**36 days**	**Mon 31/03/2**	**Mon 19/05/2**	**16**		
		Evaluation	**14 days**	**Tue 20/05/2**	**Fri 06/06/25**	**19**		
		Writing and Submission	**17 days**	**Mon 09/06/25**	**Mon 30/06/25**	**22**		
		Research Context Established	0 days	Fri 28/02/25	Fri 28/02/25	2		
		Requirements and Data Collection Finalised	0 days	Tue 04/03/25	Tue 04/03/25	9		

Figure 33 - View of Microsoft Project's layout

Appendix E – Risk Register

Project Renewable Energy Communities Digital Twin Risk Register

Positive Risk Response Options	Exploit	Share	Enhance	Accept
Negative Risk Response Options	Avoid	Transfer	Mitigate	Accept
Alternate Response Options	Contingency			

Risk ID	Description	Cause	Effect	Risk Owner	Probability (1-5)	Impact (1-5)	PI Score	Expected Result, No Action	Risk Response Type	Response description
	Description of the risk	Cause of the risk	Effect on the project	Name of person who monitors the risk	Group sourced rough estimate of how likely this is to occur	Rough estimate of how significant the impact of this risk	Probability multiplied by Impact	What will happen if the risk becomes an issue and no action is taken	Decision made by group on how to respond to this risk (see above in blue)	How do you know it is time to put the response into play
1	Computational limitations	Insufficient hardware	Inability to simulate realistic scenarios	Rui Pina	3	5	15	Reduced simulation efficiency	Mitigate	Optimise algorithm performance. Use INESCTEC HPC computing resources.
2	Ethics compliance failure	Mishandling of private data	Legal issues and reputational damage	Tiago Fonseca	1	5	5	Breach of ethical guidelines and potential lawsuits	Avoid	Conduct regular internal ethical audits in order to detect misbehaviours sooner
3	Data received not according with the requirements	Lack of data	Not able to analyse the data	Rui Pina	3	5	15	Not able to analyse the data	Mitigate	investigate available datasets
4	Lack of stakeholder alignment	Disagreement on project scope	Delayed decisions and scope creep	Tiago Fonseca	1	4	4	Delayed delivery	Transfer	Establish clear agreements and regularly align expectations.
5	Knowledge gaps in the project team	Insufficient expertise	Delays in specific project phases	Rui Pina	3	4	12	Errors in deliverables or missed deadlines	Mitigate	Upskilling of the members on the core needs of the project
6	Inadquate quality of datasets for effective training	Quality of data received	less model performance. bias in predictions	Rui Pina	4	5	20	Leading to inaccurate or unreliable outputs	Mitigate	Curate or enhance existing datasets. Synthetic data generation
7	Disruptions caused by unforeseen health concerns or personal crises.	Illness or other personal issues	Decreased work and even ceasing work	Rui Pina	1	5	5	No throughput is done and the project risks failing	Mitigate	Ensure consistent communication with the supervisor committee
8	Failures in software or risks of technical obsolescence	Lack of updates	Delayed project timelines, increased costs	Rui Pina	3	3	9	Critical project components may become inoperable	Avoid	Use open-source tools with active communities.
9	Budget constraints or funding cuts	Economic downturn	Project scope reduction	Tiago Fonseca	1	5	5	Project may face delays, inability to meet objectives	Mitigate	Plan a modular research approach Prioritize core components

Figure 34 - Expanded view from the Risk Register of the project

Appendix F – Benchmarking CSV vs JSON in exporting mechanism

During a simulation using the citylearn_challenge_2022_phase_all_plus_evs dataset [123], run over 4379 steps, two different data export mechanisms were evaluated: CSV and JSON. The primary metric under observation was the cumulative simulation time, measured against the number of simulation steps. The purpose was to assess not only how each export method scaled over time but also to determine their practical suitability in large-scale simulations.

In Figure 35, the results clearly indicate that the CSV export mechanism significantly outperforms the JSON export in terms of efficiency. In the case of the CSV export, the cumulative simulation time remains relatively low and increases in a smooth, gradual fashion. By the 4000th simulation step, the total time remained under 500 seconds (8 minutes and 20 seconds). This performance suggests that CSV is highly efficient for scenarios where data is being written frequently, as in stepwise simulations. The lightweight nature of CSV, its flat data structure, and the simplicity of its I/O operations contribute to this high level of efficiency.

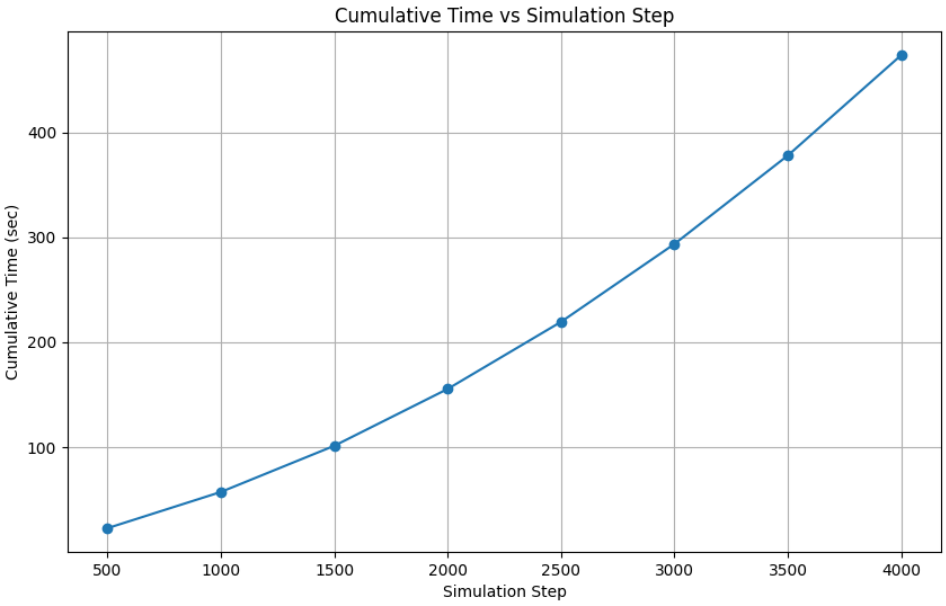


Figure 35 - Cumulative Simulation Time vs Steps (CSV Export Method)

In contrast, in Figure 32, the JSON export mechanism exhibited a much steeper increase in cumulative simulation time. By the same 4000th step, the cumulative time had risen to nearly 2900 seconds (48 minutes and 20 seconds), almost six times higher than that of CSV. This steep increase can be attributed to the additional overhead involved in JSON serialization. JSON is inherently more verbose and is designed to handle complex, nested data structures, which makes it more computationally expensive to generate and write to disk at each simulation step. Over the course of thousands of steps, this overhead accumulates significantly.

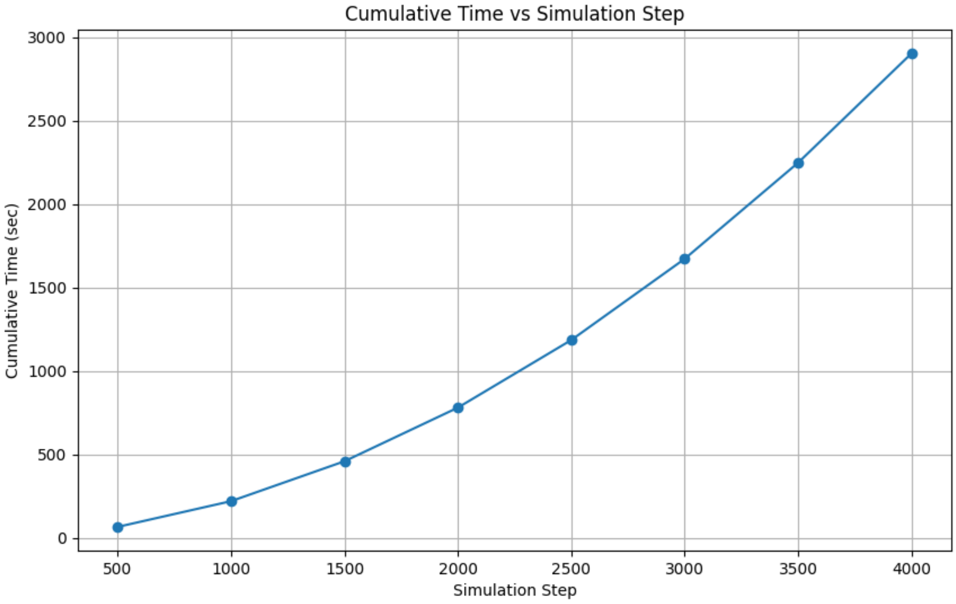


Figure 36 - Cumulative Simulation Time vs Steps (JSON Export Method)

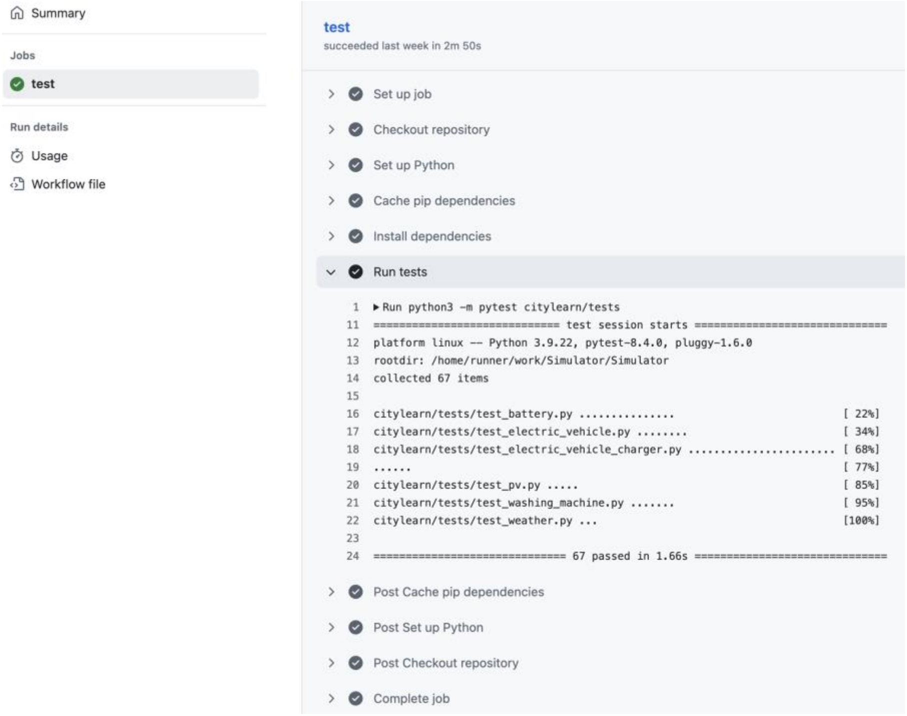
This discrepancy in performance has important implications for simulation scalability. CSV scales gracefully with simulation length, making it ideal for long or frequent simulations where performance and runtime are critical. On the other hand, while JSON is less performant, it remains useful in contexts where the preservation of hierarchical data structures or compatibility with web services and APIs is required. Its readability and flexibility are valuable, but they come at a substantial performance cost.

In conclusion, for the citylearn_challenge_2022_phase_all_plus_evs simulation, the CSV export mechanism is clearly more suitable when speed and scalability are priorities. JSON may still have a role in specific use cases, but when exporting data at every simulation step, its inefficiencies become a significant burden. Choosing the appropriate export format should therefore be guided by the simulation's specific requirements, balancing performance needs with data structure and interoperability consideration.

Appendix G – Continuous Integration Workflow using GitHub Actions

```yaml
name: CI
on:
  push:
jobs:
  test:
    runs-on: ubuntu-24.04
    steps:
      - name: Checkout repository
        uses: actions/checkout@v4
      - name: Set up Python
        uses: actions/setup-python@v4
        with:
          python-version: '3.9'
      - name: Cache pip dependencies
        uses: actions/cache@v3
        with:
          path: ~/.cache/pip
          key: ${{ runner.os }}-pip-${{ hashFiles('**/requirements.txt') }}
          restore-keys: |
            ${{ runner.os }}-pip-
      - name: Install dependencies
        run: |
          python -m pip install --upgrade pip
          pip install -r requirements.txt
          pip install pytest
      - name: Run tests
        run: python3 -m pytest citylearn/tests
```

Listing 40 - GitHub Actions Yaml Template



succeeded last week in 2m 50s

```
▶Run python3 -m pytest citylearn/tests
============================= test session starts ==============================
platform linux -- Python 3.9.22, pytest-8.4.0, pluggy-1.6.0
rootdir: /home/runner/work/Simulator/Simulator
collected 67 items

citylearn/tests/test_battery.py ...............                          [ 22%]
citylearn/tests/test_electric_vehicle.py ........                        [ 34%]
citylearn/tests/test_electric_vehicle_charger.py ....................... [ 68%]
......                                                                   [ 77%]
citylearn/tests/test_pv.py .....                                         [ 85%]
citylearn/tests/test_washing_machine.py .......                          [ 95%]
citylearn/tests/test_weather.py ...                                      [100%]

============================== 67 passed in 1.66s ==============================
```

Figure 37 - GitHub Actions Pipeline running

Appendix H – Washing Machine Data Preprocessing from Smart-PDM to CityLearn

This script, presented in Listing 41, generates a synthetic washing machine load profile for use in the CityLearn simulation environment. It processes real appliance-level data obtained from the Smart PDM dataset, specifically voltage (V) and current (A) measurements recorded during a washing machine cycle on July 7, 2022. The script simulates 8,760 timesteps by randomly assigning load windows throughout a typical week and injecting realistic power consumption patterns based on the measured values. The final output is a structured CSV file containing time-aligned washing machine usage events, formatted for compatibility with CityLearn's input requirements.

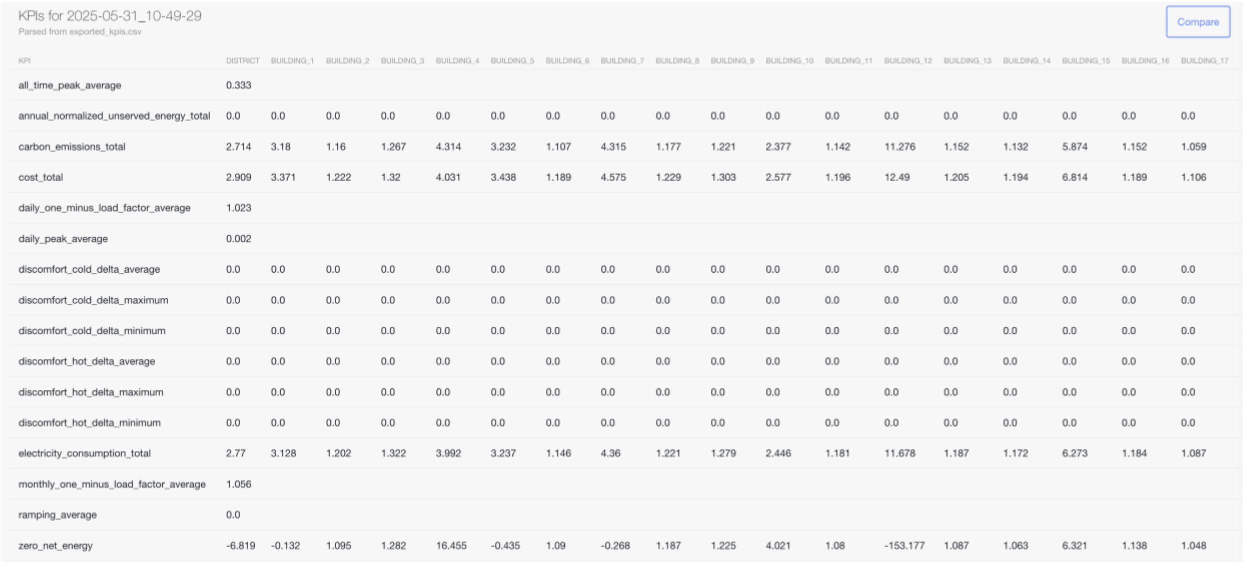
```
import pandas as pd
import random
import json

# Load real data
slow_df = pd.read_csv("/Users/rui.pina/Documents/Thesis/rebase/SMART-PDM-
Dataset/2-washing_machines/2022-07-07_10.25.00_2022-07-07_11.33.00/slow.csv")
voltage_values = slow_df['V'].values / 100
current_values = slow_df['A'].values
num_real_points = len(voltage_values)

# Parameters
target_rows = 8760
rows = []
timestep = 0
current_day_type = 7
current_hour = 24

# Track active window state
active_window = None

# Track if each day has already had a window
day_has_window = {day: False for day in range(1, 8)}

while len(rows) < target_rows:
    current_day = current_day_type
    current_hr = current_hour

    if active_window and timestep <= active_window['wm_end_time_step']:
        rows.append({
            'day_type': current_day,
            'hour': current_hr,
            'wm_start_time_step': active_window['wm_start_time_step'],
            'wm_end_time_step': active_window['wm_end_time_step'],
            'load_profile': json.dumps(active_window["load_profile"])
        })

    elif active_window and timestep > active_window['wm_end_time_step']:
        active_window = None

    if active_window is None:
```

```
        if random.random() < 0.7 and not day_has_window[current_day]:
            window_length = random.randint(2, 5)
            wm_start_time_step = timestep
            wm_end_time_step = wm_start_time_step + window_length - 1

            if wm_end_time_step >= target_rows:
                break

            num_load_points = random.randint(1, window_length - 1)
            sample_indices = random.sample(range(num_real_points),
num_load_points)
            load_profile = [
                round((voltage_values[i] * current_values[i]) / 1000, 2)
                for i in sample_indices
            ]

            active_window = {
                'wm_start_time_step': wm_start_time_step,
                'wm_end_time_step': wm_end_time_step,
                'load_profile': load_profile
            }

            day_has_window[current_day] = True

            rows.append({
                'day_type': current_day,
                'hour': current_hr,
                'wm_start_time_step': wm_start_time_step,
                'wm_end_time_step': wm_end_time_step,
                'load_profile': json.dumps(load_profile)
            })

        elif active_window is None:
            rows.append({
                'day_type': current_day,
                'hour': current_hr,
                'wm_start_time_step': -1,
                'wm_end_time_step': -1,
                'load_profile': -1
            })

    timestep += 1
    current_hour += 1
    if current_hour > 24:
        current_hour = 1
        current_day_type += 1
        if current_day_type > 7:
            current_day_type = 1
        day_has_window[current_day_type] = False

# Save to CSV
df = pd.DataFrame(rows)
df.to_csv("synthetic_load_profile.csv", index=False)
print("Saved as 'synthetic_load_profile.csv'")
```

Listing 41 - Synthetic Yearly Hourly Washing Machine Load Profile Generator Script using Real Data

Appendix I – Full KPI Tables from the Use Case using OPEVA's Frontend

This appendix presents the full tables generated through the frontend application developed by a colleague as part of a separate OPEVA initiative [26]. The data shown here was obtained using the export mechanism functionality, which enabled access to the underlying simulation results for documentation and analysis. These comprehensive results are included in their entirety here, as they were abbreviated in the main body of the dissertation for the sake of brevity. Figure 38 presents a table of the KPIs resulting from the Baseline Scenario. Figure 39 shows the KPIs from the Enhanced Scenario, while Figure 40 provides a comparison of the two, displaying the difference obtained by subtracting the Enhanced values from the Baseline values.

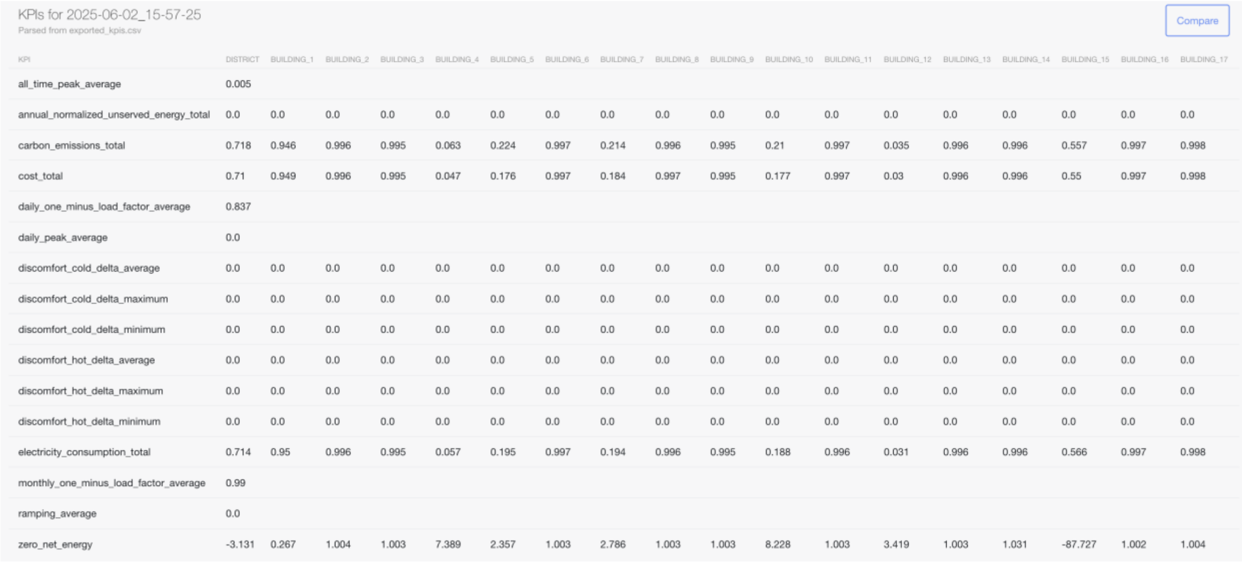


KPIs for 2025-05-31_10-49-29

Parsed from exported_kpis.csv

KPI	DISTRICT	BUILDING_1	BUILDING_2	BUILDING_3	BUILDING_4	BUILDING_5	BUILDING_6	BUILDING_7	BUILDING_8	BUILDING_9	BUILDING_10	BUILDING_11	BUILDING_12	BUILDING_13	BUILDING_14	BUILDING_15	BUILDING_16	BUILDING_17
all_time_peak_average	0.333																	
annual_normalized_unserved_energy_total	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
carbon_emissions_total	2.714	3.18	1.16	1.267	4.314	3.232	1.107	4.315	1.177	1.221	2.377	1.142	11.276	1.152	1.132	5.874	1.152	1.059
cost_total	2.909	3.371	1.222	1.32	4.031	3.438	1.189	4.575	1.229	1.303	2.577	1.196	12.49	1.205	1.194	6.814	1.189	1.106
daily_one_minus_load_factor_average	1.023																	
daily_peak_average	0.002																	
discomfort_cold_delta_average	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
discomfort_cold_delta_maximum	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
discomfort_cold_delta_minimum	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
discomfort_hot_delta_average	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
discomfort_hot_delta_maximum	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
discomfort_hot_delta_minimum	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
electricity_consumption_total	2.77	3.128	1.202	1.322	3.992	3.237	1.146	4.36	1.221	1.279	2.446	1.181	11.678	1.187	1.172	6.273	1.184	1.087
monthly_one_minus_load_factor_average	1.056																	
ramping_average	0.0																	
zero_net_energy	-6.819	-0.132	1.095	1.282	16.455	-0.435	1.09	-0.268	1.187	1.225	4.021	1.08	-153.177	1.087	1.063	6.321	1.138	1.048

Figure 38 - Baseline KPI table



KPIs for 2025-06-02_15-57-25
Parsed from exported_kpis.csv

KPI	DISTRICT	BUILDING_1	BUILDING_2	BUILDING_3	BUILDING_4	BUILDING_5	BUILDING_6	BUILDING_7	BUILDING_8	BUILDING_9	BUILDING_10	BUILDING_11	BUILDING_12	BUILDING_13	BUILDING_14	BUILDING_15	BUILDING_16	BUILDING_17
all_time_peak_average	0.005																	
annual_normalized_unserved_energy_total	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
carbon_emissions_total	0.718	0.946	0.996	0.995	0.063	0.224	0.997	0.214	0.996	0.995	0.21	0.997	0.035	0.996	0.996	0.557	0.997	0.998
cost_total	0.71	0.949	0.996	0.995	0.047	0.176	0.997	0.184	0.997	0.995	0.177	0.997	0.03	0.996	0.996	0.55	0.997	0.998
daily_one_minus_load_factor_average	0.837																	
daily_peak_average	0.0																	
discomfort_cold_delta_average	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
discomfort_cold_delta_maximum	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
discomfort_cold_delta_minimum	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
discomfort_hot_delta_average	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
discomfort_hot_delta_maximum	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
discomfort_hot_delta_minimum	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
electricity_consumption_total	0.714	0.95	0.996	0.995	0.057	0.195	0.997	0.194	0.996	0.995	0.188	0.996	0.031	0.996	0.996	0.566	0.997	0.998
monthly_one_minus_load_factor_average	0.99																	
ramping_average	0.0																	
zero_net_energy	-3.131	0.267	1.004	1.003	7.389	2.357	1.003	2.786	1.003	1.003	8.228	1.003	3.419	1.003	1.031	-87.727	1.002	1.004

Figure 39 - Enhance Simulation KPI table

Comparison between 2025-06-02_15-57-25 and 2025-05-31_10-49-29

KPI	DISTRICT	BUILDING_1	BUILDING_2	BUILDING_3	BUILDING_4	BUILDING_5	BUILDING_6	BUILDING_7	BUILDING_8	BUILDING_9	BUILDING_10	BUILDING_11	BUILDING_12	BUILDING_13	BUILDING_14	BUILDING_15	BUILDING_16	BUILDING_17
all_time_peak_average	0.328																	
annual_normalized_unserved_energy_total	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
carbon_emissions_total	1.996	2.234	0.164	0.272	4.251	3.008	0.110	4.101	0.181	0.226	2.167	0.145	11.241	0.156	0.136	5.317	0.155	0.061
cost_total	2.199	2.422	0.226	0.325	3.984	3.262	0.192	4.391	0.232	0.308	2.400	0.199	12.460	0.209	0.198	6.264	0.192	0.108
daily_one_minus_load_factor_average	0.186																	
daily_peak_average	0.002																	
discomfort_cold_delta_average	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
discomfort_cold_delta_maximum	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
discomfort_cold_delta_minimum	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
discomfort_hot_delta_average	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
discomfort_hot_delta_maximum	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
discomfort_hot_delta_minimum	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
electricity_consumption_total	2.056	2.178	0.206	0.327	3.935	3.042	0.149	4.166	0.225	0.284	2.258	0.185	11.647	0.191	0.176	5.707	0.187	0.089
monthly_one_minus_load_factor_average	0.066																	
ramping_average	0.000																	
zero_net_energy	-3.688	-0.399	0.091	0.279	9.066	-2.792	0.087	-3.054	0.184	0.222	-4.207	0.077	-156.596	0.084	0.032	94.048	0.136	0.044

Figure 40 - Exported KPIs comparison between Baseline and Controlled Simulations