Optimizing Energy Balance in Multi-Energy Microgrids

Enhancing Grid Efficiency: Utilizing Deep Reinforcement Learning for Efficient Distribution of Local Renewable Energy Resources

Thesis Geomatics

Tessel Elisabeth Kaal

Delft University of Technology

TU Delft

accenture

Optimizing Energy Balance in Multi-Energy Microgrids

Enhancing Grid Efficiency: Utilizing Deep Reinforcement Learning for Efficient Distribution of Local Renewable Energy Resources

by

Tessel Elisabeth Kaal

A thesis submitted to the Delft University of Technology in partial fulfillment of the requirements for the degree of Master of Science in Geomatics

Student number:	4654005	
Project duration:	November 14, 2023 – July 3, 2024	
External Supervisor:	Veringa, S.	Accenture
Thesis committee:	Rafiee, A.	TU Delft, first supervisor
	Meijers, B.M.	TU Delft, second supervisor
	Welle Donker, F.M.	TU Delft, delegate
	Agugiaro, G.	TU Delft, co-reader

Cover:	Man plugging charger into car (Accenture)
Style:	TU Delft Report Style, with modifications by Daan Zwaneveld

An electronic version of this thesis is available at http://repository.tudelft.nl/.

TU Delft accenture

Preface

The last eight months have taken me on a deep dive into an unfamiliar territory: machine learning. I came to Azarakhsh with the request for a research project where I could develop these specific skills and, preferably, do something energy-related. What a stroke of luck it was, as Azarakhsh provided me with the incredible chance and honor to tackle the topic presented in this thesis.

It has been a struggle at times, but that struggle has led to immense learning. Sometimes I felt completely stuck, but when things "clicked", I made leaps of ten meters ahead in my skills. Despite the stress and challenges that come with an eight-month thesis project, especially one where I chose to walk an entirely new path, I wouldn't change a thing about this journey, as I immensely value the lessons I've learned. These new skills I've acquired will serve as a foundation for further development in my career.

During the course of my research, I also engaged with the Pymgrid community. Pymgrid, a library that I now know like no other library, offers in my opinion significant potential for future research into microgrids. My interaction with the creator of the library was extremely positive; they were very open and enthusiastic about my suggestions, and hopefully, that might even bring forward handy and meaningful advancements for the next student who will carry over the stick.

I want to thank Azarakhsh, my first supervisor, for her trust in me and my process, and for the ongoing appreciation that motivated me greatly. I thoroughly enjoyed our safe, easygoing, and supportive meetings. Additionally, I would like to thank Martijn, my second supervisor, for his critical eye, thought-provoking questions, and assistance in decoding problems when I was stuck. Furthermore, I want to thank Amin, who is pursuing a PhD in the field. Working together has been a pleasure; his help with the project kickoff and setting up the initial steps was very valuable. Lastly, I want to thank the entire committee for their attendance and flexibility, which was greatly appreciated!

I conducted my thesis within Accenture, under the guidance of Sanne Veringa. Not only has she provided a listening ear and perhaps even been my 'rubber duck,' but we have also become good friends. I have valued our coffee chats and football games immensely. As part of the GIS team at Accenture, I participated in their meetings and was given the opportunity to engage in some additional activities that interested me. Together with Sanne, I helped set up our team's booth at the ESRI conference. This experience taught me a great deal about internal business development and cooperation. Also, the days themselves were great, as it was incredibly interesting to learn about new possibilities in the geofield and share knowledge with all the other geo-focused attendees.

Lastly, I need to thank all my friends and family for their support and for recharging my battery with *'gezelligheid'*. Although I enjoyed the entire process and learning curve, the last few weeks were particularly tough. During this period, their encouragement and companionship were crucial in helping me push through to the end.

And now, do I really have a master's degree? WHAT! This realization still feels surreal, but I am incredibly proud of what I have accomplished and excited for the future.

Tessel Elisabeth Kaal
Delft, July 2024

Summary

The transition to a fully electrified energy grid in the Netherlands is essential for the country's energy transition strategy. However, the existing grid infrastructure struggles to support the increased demand due to electrification, leading to congestion and potential outages anticipated before 2030. Microgrids, decentralized networks where generation, heat, storage, and consumption are coordinated, offer a promising solution by providing the required flexibility for the integration of Distributed Renewable Energy Resources (DERs) and potentially reducing energy losses associated with long-distance energy exchange.

This study explores the implementation of Deep Reinforcement Learning (DRL) in a district-level microgrid in the Netherlands. The primary objective is to minimize grid connection dependency and reduce energy exchange distances within the microgrid cost-effectively, thereby enhancing grid reliability and sustainability, answering the research question: *Main Research Question*: *How can Distributed Energy Resources be deployed and managed in a cost-effective manner within an electrified microgrid to achieve a balance between energy consumption and production in order to minimize the burden on the central grid given fluctuating demand?*

A comprehensive framework was developed offering a structured approach for simulating and optimizing microgrid scenarios, providing practical solutions and frameworks for the efficient management and deployment of microgrids. The study demonstrates that DERs can be effectively deployed and managed within an electrified microgrid through a combination of simulation techniques with geographical data input, a DRL model with a Deep Q-Network (DQN) that controls the system, and dynamic mappings and visualization. While the DQN demonstrated its potential in optimizing microgrid configurations, challenges were noted due to the large action spaces required for routing optimization. To overcome these limitations, future work is proposed to combine DRL with Graph Neural Networks (GNNs). This novel method shows promise in enhancing the scalability and efficiency of the optimization process, enabling distance and routing optimization without an aggregated model.

Scenarios were created to test the influence of different components, demonstrating how various factors affect overall system performance. The results of the model were as expected, and undertakings to optimize the model further led to significant improvements in the variability of the learning curve and of the rewards.

The study concludes that the developed pipeline contributes to existing works by offering a structured approach for simulating and optimizing microgrid scenarios and testing different setups. While this research approach has proven effective, future work should enhance the realism of components and address some computational limitations encountered. Refining these methodologies by incorporating GNNs, integrating real-world data, and extending the model with additional algorithms for performance comparison will further advance the field. This effort ultimately supports the vision of a reimagined, sustainable, resilient, and efficient energy grid.

Keywords: Renewable Energy integration, Microgrid, Deep Reinforcement Learning, Dynamic Clustering, DQN, Graph Neural Network, Digital Twin

Contents

List of Figures

List of Tables

Listings

Nomenclature

Abbreviations

Abbreviation	Definition
AC	Alternative Current
ANN	Artificial Neural Network
BES	Battery Energy Storage
BESS	Battery Energy Storage System
COP	Coefficient of Performance
DC	Direct Current
DER	Distributed Energy Resources
DQN	Deep Q Network
DRL	Deep Reinforcement learning
DT	Digital Twin
EMS	Energy Management System
ESS	Energy Storage System
EU	European Union
EV	Electric Vehicle
FC	Fuel Cell
FCEV	Fuel Cell Electric Vehicle
GA	Genetic Algorithm
GE	Graph Embedding
GIS	Geographical Information System
GNN	Graph Neural Network
HS	Hydrogen Storage
HRES	Hybrid Renewable Energy Sources
IEA	International Energy Agency
IoT	Internet of Things
LAN	'Landelijk Actie plan Netcongestie'
LES	Local Energy System
MG	Microgrid
PJ	Peta Joule
POC	Proof of Concept
PSO	Particle Swarm Optimization
RED	Renewable Energy Directive
RES	Renewable Energy Sources
RL	Reinforcement Learning

Symbols

Symbol	Definition
α	Learning rate
A_{swept}	Rotor swept area of the turbine
A_{PV}	Surface area of PV panels

Continued on next page

Table 2 – continued from previous page

Symbol	Definition
A_{cross}	Cross-sectional area of the conductor
C_p	Power coefficient of wind turbines
C_{tU}	Cost of purchasing electricity from the grid at time t
E_{max}	Maximum energy level of energy storage device
E_{min}	Minimum energy level of energy storage device
$E_{storage}$	Energy level of energy storage device or state of charge
E_{solar}	Energy from PV panels
η_{ch}	Charging efficiency of energy storage device
η_{dis}	Discharging efficiency of energy storage device
η_{PV}	Efficiency of PV panels
$\eta_{turbine}$	Efficiency of wind turbine
∇	Gradient
P_{loss}	Power loss in the conductor
P_{wind}	Power output from the wind turbine
P_{geo}	Power from geothermal energy
P_{out}	Heat pump output power
P^E_{max}	The maximum charging/discharging power of energy storage device
P^E_t	Charging or discharging power of energy storage device at time t
$Q^*(s,a)$	Optimal state-action value function at state s and action a
r	Immediate reward received after taking action a in state s
R	Reward function
R_t	Price of electricity from grid at time t
γ	Discount factor used in the calculation of future rewards
ρ_{res}	Resistivity of the conductor material
ρ_{air}	Air density at turbine location
δt	Time period
s	Current state in Markov Decision Process
s'	Next state in Markov Decision Process
a	Action taken in the current state s
a'	Possible next action from the new state s'
v	Wind speed
Y	Discount factor
$\mathrm{SOC}_{\mathrm{new}}$	New state of charge of the battery
$\mathrm{SOC}_{\mathrm{old}}$	Old state of charge of the battery
P_{charge}	Power for charging the battery
$P_{\mathrm{discharge}}$	Power for discharging the battery
μ	Mean reward
σ^2	Variance of the rewards
σ	Standard deviation of the rewards

1 Introduction

With the European Union (EU) raising its renewable target, set within the Renewable Energy Directive, to 42.5 percent by 2030 and moving away from gas, the energy landscape in the Netherlands is undergoing significant changes toward sustainability and renewable energy integration. The shift is underpinned by the European Commission's REPowerEU plan introduced on 18 May 2022 in response to the energy market disruptions caused by Russia's invasion of Ukraine. Part of the broader 'Fit for 55' package, the plan aims to significantly reduce the EU's dependence on Russian fossil fuels and enhance its overall climate objectives (European Council, 2022, 2023).

The Netherlands has set ambitious goals, including revisions to critical energy legislation, such as the Energy Performance of Buildings Directive and the Energy Efficiency Directive, to adopt a more vigorous approach towards renewable energy. The necessity of this shift is highlighted by the benefits of renewable energy being low-cost and domestically produced, which reduces dependency on external suppliers (European Commission, 2023). The transition requires the electrification of energy systems, but the Dutch electricity grid is currently ill-prepared for a scale-up of this magnitude.

Progress is evident, particularly in integrating solar and wind energy. However, the inherent intermittency and variability of these renewable sources present considerable challenges to the existing grid infrastructure, necessitating the development of more flexible and efficient transmission systems to ensure consistent and reliable energy delivery (CBS Statline, 2023a). Tennet, the Dutch electricity transmission system operator, has publicly underscored the urgency of the situation by highlighting the potential for electricity shortages by 2030 if grid flexibility is not enhanced (Reuters, 2023). Therefore, innovative and flexible solutions are crucial for the Netherlands to align with the EU's renewable energy goals and successfully navigate the transition EBN, 2023; European Commission, 2023; International Energy Agency, 2023.

1.1. Current Strategies to Address Grid Challenges

The Dutch government has called the National Action program Netcongestion (LAN) into being (Netbeheer Nederland et al., 2022), in which they have identified three main approaches to combat the congestion. One of these three approaches is the creation of 'energy hubs', where businesses in proximity coordinate their energy demand and supply, thereby reducing the overall load on the grid. The LAN aims to enhance system flexibility by incorporating energy hubs, which, as described, represent "decentralized networks where generation, heat, storage, and consumption are coordinated" (Netbeheer Nederland et al., 2022). These energy hubs, scientifically known as microgrids, are an increasingly popular topic. The growing body of research highlights the potential of microgrids to advance the sustainability and flexibility of the grid, positioning them as a promising alternative (Uddin et al., 2023). By reducing the transmission distance to and from energy sources, microgrids minimize transmission losses. Local balancing of energy within these microgrids is highly feasible, optimizing their independence, flexibility, and effectiveness in energy distribution. This optimization is crucial for the advancement of resilient and efficient energy systems. The exploration and development of microgrids reflect a critical step towards reimagining energy distribution networks to meet the demands of the future in line with the government plans.

> By 2030, it is expected that, without additional measures, 40 percent of the low-voltage networks will have bottlenecks, and thus will not meet the requirements of the network operator. (LAN, 2024)

Optimizing microgrids is challenging, as it often involves multiple variables and constraints, making it difficult to find a single definitive solution. Literature highlights many different algorithms and numerous objectives. Among these, DRL is emerging as the most effective method for improving the performance of microgrids that rely on renewable energy (Hossain et al., 2023; Ifaei et al., 2023; Tajjour & Singh Chandel, 2023). DRL has demonstrated strong performance in similar research setups, effectively handling non-linearity and uncertainty, making it well-suited for the microgrid environment (Subramanya et al., 2022). Its distinct advantage over metaheuristics, particularly its ability to consistently reach optimal solutions, is supported by various studies (Klemm & Wiese, 2022; Li et al., 2023; Subramanya et al., 2022). The potential of DRL to effectively manage complexities, such as efficiency and adaptability, makes it an ideal candidate for application in advanced microgrid optimization strategies.

1.2. The Research gap

Currently, there are numerous existing implementations of DRL on microgrids; however, applications in the urban domain remain underrepresented. Most existing research on microgrid and DRL models focuses primarily on industrial or isolated settings. This study proposes an automated pipeline to test DRL models on an urbanized environment in the Netherlands, aiming to illuminate the complexities arising from the diverse user profiles in close proximity. Moreover, many studies rely on small-scale synthetic loads, whereas this research uses real input data to simulate loads that accurately mimic different user profiles in a scalable manner. The approach can produce load simulations for various buildings within minutes. The input data includes both residential and building information, creating a realistic depiction of demand per building. The data leverages Geographical Information Systems (GIS) and geoprocessing tools for information gathering and maintaining a geographical extent for visualization purposes, utilizing open data from the GIS community. In this research, loads are simulated using EnergyPlus, employing a scalable methodology to generate loads based on the characteristics of buildings and their occupants. These efforts aim to provide insights into load balancing among users with varying load profiles and how this affects peak demand. Aggregated information on demand and supply is optimized, with a fully functional model designed to offer insights into system traits and the requirements for implementing microgrids in the Netherlands. Additionally, real geographical data is used for exploratory analysis for creating scenarios that validate and showcase the system to stakeholders. Finally, this study presents a detailed process description and suggestion for conducting a novel approach to address the varying topologies and non-uniformity of energy microgrids. This innovative method involves utilizing a Graph Neural Network (GNN) with dynamic clustering. The framework outlines this approach and sets the stage for implementation in future work.

1.3. Proposed Approach and Research Questions

Recognizing the limitations of centralized power grids in meeting the nuanced energy demands of urban environments, this study proposes a new model to manage microgrids that perform independent and sustainable. The inherent intermittency and variability of renewable sources present considerable challenges to the existing grid infrastructure, necessitating the development of more flexible and efficient transmission systems to ensure consistent and reliable energy delivery (CBS Statline, 2023a). Microgrids offer numerous benefits over conventional power grids, including enhanced reliability, reduced transmission losses, environmental benefits, and increased flexibility (Ali et al., 2022; Shahzad et al., 2023). They allow for the integration of diverse and DERs, fostering a more resilient and sustainable energy infrastructure. This research proposes to implement an optimization model for a case study DRL to manage an urban microgrid. The main objective is to minimize the burden on the grid and enhance grid flexibility by increasing the energy balance while integrating Renewable DERs and meeting variable energy demand. Additionally, a location-based optimization using Hierarchical Clustering aims to

mitigate transmission losses. The results are visualized to provide value to stakeholders, confirmed through collaboration. The research question guiding this study is: **Research question:** *How can Distributed Energy Resources be deployed and managed in a cost-effective manner within an electrified microgrid to achieve a balance between energy consumption and production in order to minimize the burden on the central grid given fluctuating demand?* To effectively address the main research question, it's divided into several sub-questions. This division allows for a targeted and in-depth exploration of each component of the main issue. The sub-questions are as follows:

Q 1. *Which modeling aspects need to be considered when designing a microgrid and its energy management system, and how are these reflected in the existing literature?*

Q 2. *How can reinforcement learning be utilized to effectively place and manage all identified modeling aspects in a microgrid while maintaining grid stability and reliability?*

Q 3. *How can the location-based optimization of a microgrid be effectively visualized to ensure it creates value for all stakeholders?*

1.4. Case Study and Scenarios

The proposed study involves a detailed case study to showcase the implementation of the microgrid model in the Netherlands. The model will operate on a 15-minute basis over one year, assuming a fully electrified grid. This microgrid is connected to the main grid, reflecting a realistic energy network configuration, focusing on urban areas with mixed functions such as industrial or residential. The model is tested through various scenarios to provide insights into the required innovations for transitioning towards a microgrid-based energy system.

1.5. Contributions Research

This study aims to contribute to the knowledge of energy system modeling necessary for the national transition to a reimagined energy distribution network. By integrating various renewable energy sources into microgrids, the study seeks to minimize the burden on the main grid. Additionally, location-based optimization through hierarchical clustering is showcased in order to set the stage for further advancements in this framework to expand the complexity of the state spaces, and to minimize transmission losses further. Lastly, the results are visualized for stakeholders finalizing the pipeline to gain insights into the implementation of microgrids in more case studies. The research also aims provides new insights into the practical application of microgrids, specifically for the Netherlands, contributing to the transition towards an energy grid that aligns with the European Union's goals. By bridging this gap, the study offers innovative solutions crucial for the Netherlands to navigate the energy transition successfully. In summary, this investigation not only advances the field of energy system modeling with a completely scalable pipeline for the assessment of urban areas and microgrids, but it also introduces a novel combination of methods to minimize transmission losses and enhance energy efficiency. All in all, aiming for a contribution to the current landscape that is oriented towards achieving climate neutrality.

1.6. Reading Guide

The subsequent chapters provide a comprehensive exploration of the topic, showcasing an intermediate framework, and consequential modeling steps that aim to answer the sub-research questions and with that the main research question in the results and conclusion, see Figure 1.1. Chapter 2 delves into related work to contextualize the problem statement and it aims to function as the catalog for the theoretical framework, which is presented in the subsequent chapter, Chapter 3. Chapter 4 presents the research steps that are taken to answer the other three research questions, and finally answer the main research question. The implementation of the optimization model is provided in a separate chapter, Chapter 5, in order to clearly provide the implementation steps. The results for the optimization, and for the visualization are presented in the subsequent chapters respectively Chapter 6, and Chapter 7. Lastly, the research is discussed, and thereafter the conclusion and recommendations for further research are presented in Chapter 8.

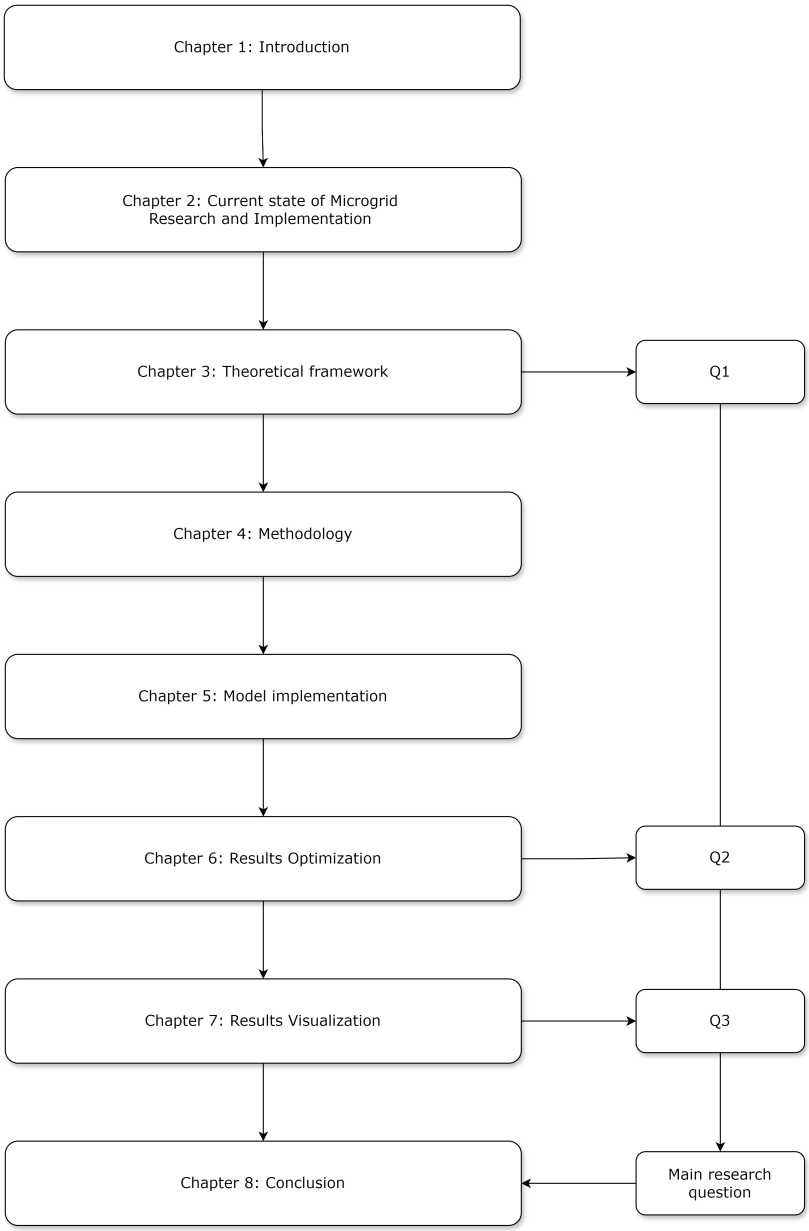


Figure 1.1: Chapters and presentation research questions

2

Current State of Microgrid Research and Implementation

This chapter provides essential background information to support an understanding of the proposed work. It begins with an examination of the current state of microgrids as presented in the literature, offering both historical and contemporary perspectives. The literature review focuses on the specific challenges and issues associated with adopting microgrids, aiming to identify difficulties and concerns relevant to their implementation. Subsequently, the chapter delves into the various components that microgrids can encompass. Finally, it presents and analyzes relevant algorithms and case studies from existing literature. This analysis serves as a foundation for the theoretical framework, which is developed and presented in Chapter 3.

2.1. Background Information

The electrification of the energy landscape is substantially increasing electricity demand, leading to grid congestion and the heightened risk of outages among other potential consequences. This growing demand places an escalating burden on the existing grid infrastructure, intensifying the need for immediate and effective solutions. Microgrids represent a promising approach to reduce this strain. By facilitating localized generation and consumption of electricity, microgrids can significantly reduce the need for extensive energy transmission, thereby minimizing energy losses and enhancing the overall resilience and efficiency of the energy system. The strategic implementation of microgrids can help alleviate some of the burdens currently faced by the national grid, enhancing overall resilience and efficiency of the energy system. The need for such innovations is particularly visible in the Capacity Map provided by Netbeheer Nederland, which illustrates the regions where grid capacity is already stretched to its limits, see Figure 2.1. This map offers a comprehensive overview of the available capacity within regional electricity grids for connections exceeding 3x80A. The map reveals that in red-marked areas, no excess energy can be supplied to the grid, highlighting the limitations of the current infrastructure and the urgent need for expansion and modernization. As well as leading to a slower shift towards more sustainable forms of energy, these congestion problems lead to market barriers (Netbeheer Nederland et al., 2022). Companies are not able to feed energy into or take energy from the grid, meaning their economic activities are significantly hindered. This situation underscores the importance of strategic investments in grid enhancement and expansion, alongside the development of innovative technologies for energy storage and distribution. The aggregate peak demand in the Netherlands underscores a critical challenge for the electricity grid.

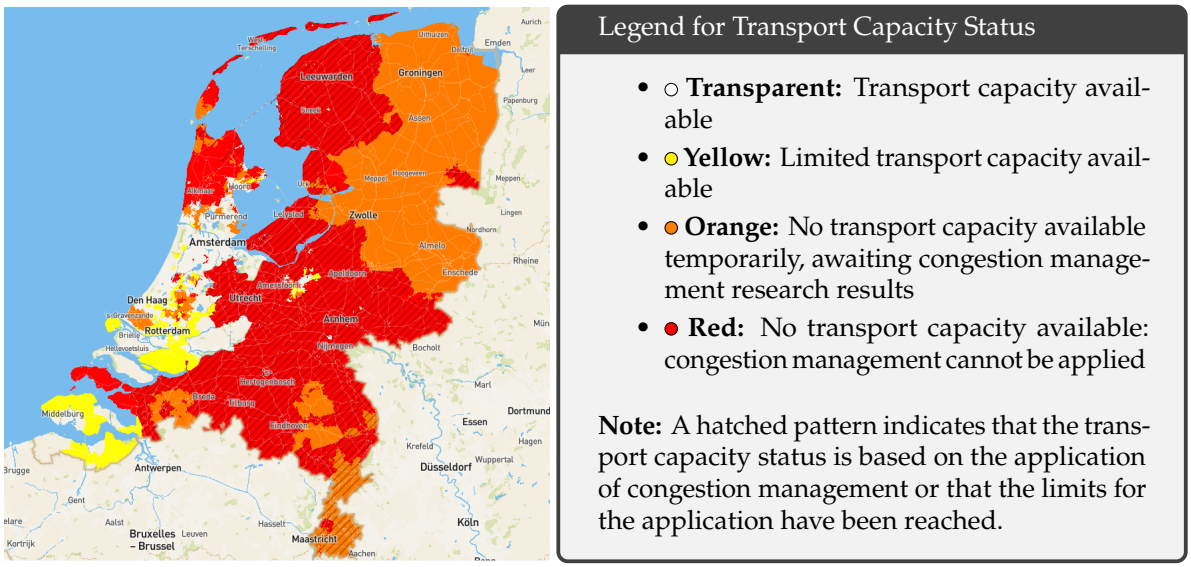


Figure 2.1: Capacity map illustrating regional grid limitations (Netbeheer Nederland, 2024)

This map, depicted in Figure 2.1, highlights the red-marked areas where no excess energy can be supplied to the grid. Such capacity constraints underscore the urgent need for implementing microgrids, which can operate independently of the main grid and provide localized support, thus mitigating congestion and enhancing the grid's ability to handle peak loads. Furthermore, by reducing the distance electricity has to travel, microgrids not only decrease transmission losses but also bolster the grid's reliability and sustainability.

The following sections will further explore the benefits and challenges of microgrid technology, drawing on specific case studies and theoretical discussions to illustrate how these decentralized systems can transform the energy landscape in the Netherlands and beyond.

2.1.1. Benefits Microgrid and Renewable Energy

Microgrids present a promising option as an alternative to the conventional central grid that relies predominantly on fossil energy and is a one-way system (Uddin et al., 2023). Microgrids not only facilitate the localized distribution of renewable energy but also bring about a multitude of benefits, including "increased reliability, reduced energy costs, improved energy security, environmental benefits, and increased flexibility," as noted by Shahzad et al. (2023). Increasing flexibility of the power grid (by its renewable energy sources-integration) in particular is one of the top priorities of the International Energy Agency (2023).

The shift towards renewable energy frequently requires the adoption of distributed generation technology (Gao et al., 2021), distinct from large-scale, centralized power generation systems. This approach, aligned with the geographic distribution of renewable resources, optimizes the development and utilization of these energies (Tajjour & Singh Chandel, 2023). Distributed generation not only supports voltage profiles through reactive power and decentralizes energy supply, but it also integrates heat loads in co-generation and provides ancillary services and demand response (Zia et al., 2018). It effectively reduces line losses and mitigates outages in transmission and distribution systems (Basu et al., 2011; Zia et al., 2018). In their comprehensive review, Gao et al. (2021) describe the large-scale adoption of renewable energy through distributed generation as an inevitable evolution for power grids. Further, Shahzad et al. (2023) underscore microgrids' role in addressing energy poverty and promoting sustainable economic growth. Lastly, Uddin et al. (2023) points out that there is no doubt that microgrids will lead to a greener and more flexible power system. However, there are still a number of issues and challenges ahead, described further in Subsection 2.1.2.

2.1.2. Challenges Microgrid and Renewable Energy

The management of microgrids presents a series of challenges that can be categorized into technical, economic, market, or regulatory issues. A state-of-the-art review by Uddin et al. (2023) highlights the difficulties in meeting demand given the unpredictable daily and seasonal variations in renewable

energy generation and sudden load changes. These variations introduce significant challenges for system frequency and voltage stability, making the management of isolated microgrids demanding (Li et al., 2023; Uddin et al., 2023). Furthermore, they encounter various issues, including more technical complexities, high initial costs, regulatory barriers, difficulties with interconnection, and ongoing maintenance and operational demands (Shahzad et al., 2023). One of the inherent drawbacks of renewable energy sources used in microgrids is their volatility, uncertainty, and instability, which can negatively impact microgrid performance by the short-term voltage changes (Gao et al., 2021; Uddin et al., 2023).

Effective load forecasting is crucial for reliable microgrid operations. However, this often requires extensive datasets that are not readily available, particularly for newly distributed sources. Traditional energy management systems, used for centralized systems, are expensive, complex, and sometimes limited in functionality. While the energy storage systems (ESSs) and hybrid technologies are employed to maintain frequency, stability, and continuous power supply, these solutions can be cost-prohibitive and necessitate complex control mechanisms (Tajjour & Singh Chandel, 2023). The integration of technologies such as ESSs, electric vehicles (EVs), and Internet of Things devices into microgrids adds to this complexity.

Alongside the importance of load forecasting, long-term power forecasting is also critical for the strategic planning and development of microgrids. This involves predicting future energy needs and production capabilities over an extended period, which is essential for ensuring the scalability and sustainability of microgrid operations. Long-term forecasting helps in making informed decisions regarding capacity expansion, infrastructure investment, and integration of new technologies. However, long-term forecasting comes with its own set of challenges, including the need to balance economic, technical, and environmental considerations, particularly in the context of evolving energy policies and market conditions (Pang et al., 2024). The dynamic nature of renewable energy sources and the growing trend towards decentralized energy generation further complicate long-term planning in microgrids.

Lastly, Uddin et al. (2023) explains that energy losses still occur, attributed to several factors including transmission losses. Further insights into transmission and distribution losses are provided in a report by CEER (2020). The report details the distribution losses for the Netherlands, as shown in Figure 2.2.

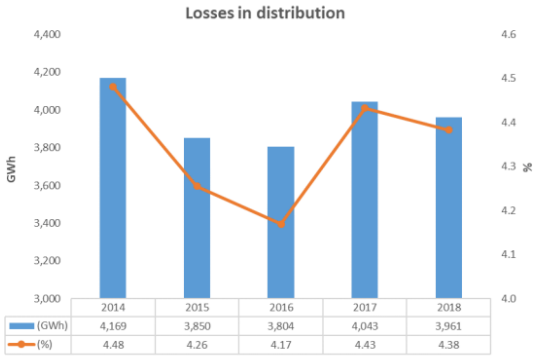


Figure 2.2: Distribution losses in the Netherlands (CEER, 2020)

The technical attributes of the distribution network significantly influence the losses. Distribution losses are typically higher in extended networks because the line length is directly correlated to power loss. This correlation can be quantitatively expressed by the following equation, which details how power loss is a function of line length, resistivity of the conductor material, and the current flowing through the line, see Equation 2.1.

$$P_{loss} = I^2 \left(\rho_{res} \frac{L}{A_{cross}} \right) = I^2 \times R \tag{2.1}$$

Where:

- ρ_{res} represents the resistivity of the conductor material.
- L represents the length of the conductor.

- A_{cross} represents the cross-sectional area of the conductor.
- P_{loss} represents the power loss.
- I represents the current flowing through the line.
- R represents the resistance of the line, given by the equation $\rho_{res} \frac{L}{A_{cross}}$.

Addressing these losses involves strategic planning and engineering solutions aimed at reducing line lengths and improving conductor quality, thereby minimizing resistive losses and enhancing overall system efficiency. This is crucial for maintaining both economic viability and operational effectiveness in microgrid deployments. The variety and complexity of microgrids necessitate advanced optimization techniques to balance energy demand and production effectively. These optimizations are important to enhance our ability to ensure stable microgrid operations, and to reduce the dependency on the central grid. This need for optimization underscores the requirement and importance of the development of sophisticated analytical approaches, including mathematical and computational models, to achieve a harmonious balance between energy consumption and production.

2.1.3. Approaches and Used Models

Addressing the challenges associated with microgrid management, the literature proposes various methods and strategies aimed at optimizing the balance between energy demand and production. Strategies like peak shaving and load shifting are designed to even out the demand curve and reduce strain on the grid during high usage times. Advanced load forecasting uses sophisticated algorithms to anticipate energy consumption patterns, facilitating proactive energy distribution adjustments (Tajjour & Singh Chandel, 2023). Hybrid Energy Systems combine various energy sources to enhance the reliability of power supply, while DER management optimizes the deployment of localized energy generation units.

Effective control approaches are crucial for achieving this balance. Uddin et al. (2023) highlights centralized, decentralized, and distributed control structures. Centralized control focuses on a single point of command, decentralized control distributes authority across multiple points, and distributed control emphasizes a networked approach, enhancing flexibility and resilience in the face of disruptions. More complex systems differentiate between two or three control levels (Gao et al., 2021; Trivedi & Khadem, 2022; Yamashita et al., 2020), which are beneficial for managing different operational aspects of a microgrid, ensuring both stability and economic efficiency.

Mathematical optimization of these energy systems involves formulating problems using mathematical models to represent the microgrid system's behavior and constraints. Techniques such as linear programming, mixed-integer linear programming, and nonlinear programming are used to find optimal solutions for energy distribution, storage management, and demand-response strategies. These methods provide precise and scalable solutions that can adapt to the dynamic nature of microgrids, ensuring efficient and reliable operation while meeting multiple objectives.

Comprehensive optimization approaches also include heuristics, metaheuristics, and machine learning algorithms. Heuristic algorithms provide practical solutions by quickly navigating through complex problems, though they may not always guarantee the optimal outcome. metaheuristics, such as genetic algorithms and particle swarm optimization, offer a structured approach to search space exploration, balancing between exploring new areas and refining known good solutions. Machine learning enhances microgrid optimization by predicting patterns and making informed decisions based on data-driven insights. For example, reinforcement learning develops adaptive control strategies to manage the variability of renewable sources and demand fluctuations.

All these optimization techniques aim to achieve a balance across multiple objectives: enhancing technical performance, minimizing environmental impact, ensuring economic viability, and meeting social expectations (Ifaei et al., 2023). Examples include optimizing voltage regulation, the share of renewables, greenhouse gas emissions, or data prediction (Hossain et al., 2023; Ifaei et al., 2023; Klemm & Wiese, 2022). The case studies in the literature often focus on one or a few main objectives due to the complexity and specificity of each optimization goal. This approach allows researchers to isolate the impact and feasibility of particular strategies without the confounding effects of multiple simultaneous objectives.

Performance indicators used in these studies include optimization of voltage stability, voltage deviation, and unmet load. Klemm and Wiese (2022) warns of the deterioration when using only one sub-indicator, recommending multi-criteria approaches. Constraints in these optimizations can be

related to planning, management, physics, policy, validity, feasibility, reliability, balance, design, or network. Examples include supply-demand, spatial configurations, or loss of load expected (Ifaei et al., 2023).

In summary, the literature suggests a range of strategies and control methods for constructing and managing microgrids. This thesis explores minimizing the burden on the main energy grid, focusing on connected microgrids. This further exploration aims to achieve a harmonious balance between energy demand and production, enhancing overall grid resilience and efficiency.

2.2. Implemented Microgrids and Case Studies

Microgrids are very diverse, dependent on their configurations and the component in the system. Parisio et al. (2014) defined microgrids as "subsystems of the distribution grid, which comprises generation capacities, storage devices, and controllable loads, operating as a single controllable system either connected or isolated from the utility grid." Following an extensive review of the literature, Uddin et al. (2023) further refined the concept of a "microgrid" as a small-scale power subsystem comprising a limited number of DERs, including both renewable and conventional energy sources. Uddin et al. (2023) categorize microgrids into groups based on applications, infrastructure, and end-users requirements. Multiple case studies of microgrids in similar contexts have been conducted, demonstrating the composition of a conceptual model often consisting of multiple sources, loads, and management entities. A notable example is the conceptual model by Uddin et al. (2023).

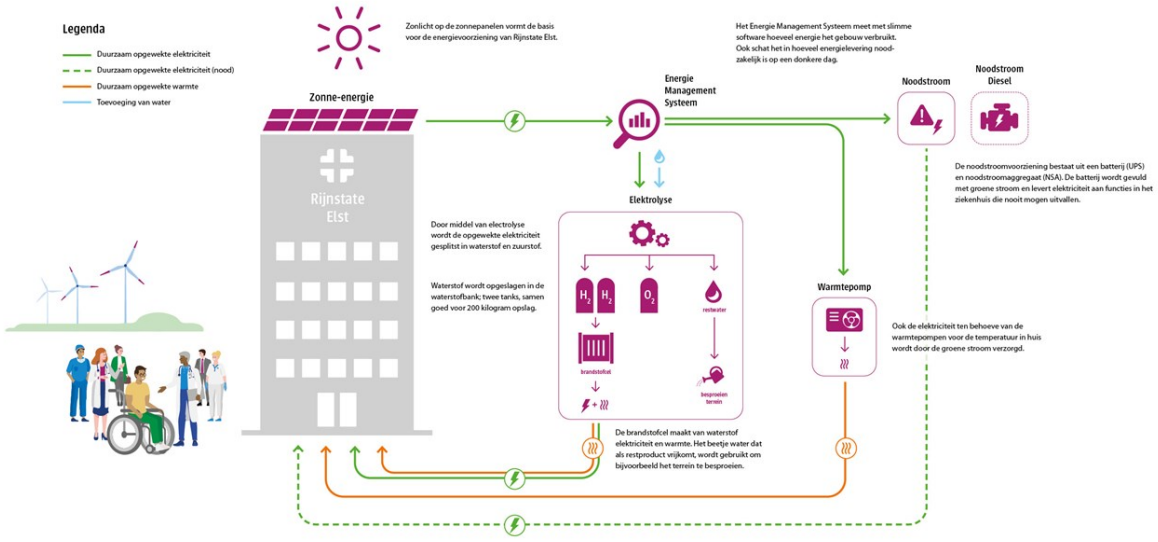


Figure 2.3: Schematic Energy Management System Rijnstate (Rijnstate, 2023)

There are a couple of microgrids effective in the Netherlands. An example of an implemented microgrid is the hospital Rijnstate (Rijnstate, 2023). The hospital functions largely independently from the central grid, providing enhanced resilience crucial for a healthcare facility that requires a continuous and reliable energy supply.

Another implemented microgrid in the Netherlands is in Nederweert, primarily serving industrial purposes (Stultiens, 2023). Additionally, an exploratory research project in Aardehuizen features a microgrid consisting of 23 earth ship-type houses, achieving almost 90% self-sufficiency (Spectral, 2018). The project focuses on increasing knowledge and evaluating several techno-economic performance indicators, such as energy production versus consumption, self-sufficiency, capital investment, and payback period.

The diversity in microgrid setups stems from varying needs and socio-environmental factors. Industrial areas, hospitals, and residential communities each have distinct energy demands and spatial configurations for DERs.

2.3. Geographical Analysis

The problem of microgrids exhibits a significant amount of contextual variety, largely due to the differences in resources available at various locations and the varying demand profiles. The geographical context significantly influences the design, implementation, and management of microgrids, making it essential to consider these elements for effective microgrid operation.

In residential neighborhoods, energy demand typically peaks during morning and evening hours when residents are most active, leading to lower and more predictable demand compared to industrial areas. These residential areas can leverage renewable resources such as solar power due to the availability of roof space for solar panels. Additionally, there is a high emphasis on reliability and quality of power supply to avoid disruptions in daily activities, and the implementation of demand response programs and energy-efficient appliances can be more effective.

On the other hand, industrial areas have higher and more consistent energy demands throughout the day, with significant variations depending on the type of industry and shift patterns. These areas might require a mix of energy sources, including renewable and non-renewable, to meet their substantial energy needs, and the integration of Combined Heat and Power (CHP) systems can be advantageous. High reliability and quality of power are critical to prevent production losses and maintain operational continuity, and industrial energy management systems, process optimization, and the use of energy-efficient machinery are essential strategies.

The geographical component is analyzed by means of exploratory geographical research and processes. It is an important component in the planning and optimization of microgrids. It contributes to the approaches mentioned in the previous sections in several ways. For instance, geographical analysis helps in assessing the availability of renewable energy resources such as solar, wind, and hydro power, which is crucial for designing Hybrid Energy Systems that combine various sources to ensure a reliable power supply.

Determining the optimal locations for DERs is essential for maximizing their efficiency and effectiveness, and geographical analysis helps in identifying sites that may minimize transmission losses and enhance the overall resilience of the microgrid.

The geographical context influences the choice of control strategies (centralized, decentralized, or distributed) by highlighting the logistical and infrastructural constraints of an area, making decentralized control more suitable for areas with dispersed energy resources.

Geographical data can also enhance optimization algorithms by providing insights into local conditions and constraints, which can be used to fine-tune heuristic, meta heuristic, and machine learning algorithms for better performance in specific contexts.

2.4. Configurations & Components

The set up of a microgrid is made up out of the components and by the configuration. The configuration entails the network setup and the components are the elements that are integrated in the system. These two aspects are further explained below.

2.4.1. Configurations

In microgrid configurations, the architecture can vary among alternative current (AC), direct current (DC), or a hybrid AC/DC system, tailored to the specific operational needs of the microgrid (Ali et al., 2022). DC-coupled energy storage is generally more suited for low-power applications like EVs and residential solar installations, optimizing battery use, stability, compactness, and quietness despite its complexity and expense (ACP, n.d.). On the other hand, AC-coupled energy storage fits better with medium to high-power uses such as industrial energy and grid storage, offering flexible design and affordability, yet with some trade-offs like inverter losses and capacity constraints. When linked to solar power sources, ESSs benefit from DC coupling, leveraging the DC from solar panels for enhanced efficiency across residential to utility scales. Each coupling approach presents unique advantages and challenges, shaping their adoption in varying energy storage scenarios (Ali et al., 2022).

For a hybrid grid a bidirectional converter is essential in to convert DC from the battery/fuel cell or their combination into AC (Singh et al., 2019). See Figure 2.4 for a hybrid configuration.

As mentioned before, these microgrids are designed to operate either independently or in conjunction with the main power grid, a duality that enhances the overall security and reliability of the power supply (Gong et al., 2020). The distinct operational features of each microgrid configuration are critical

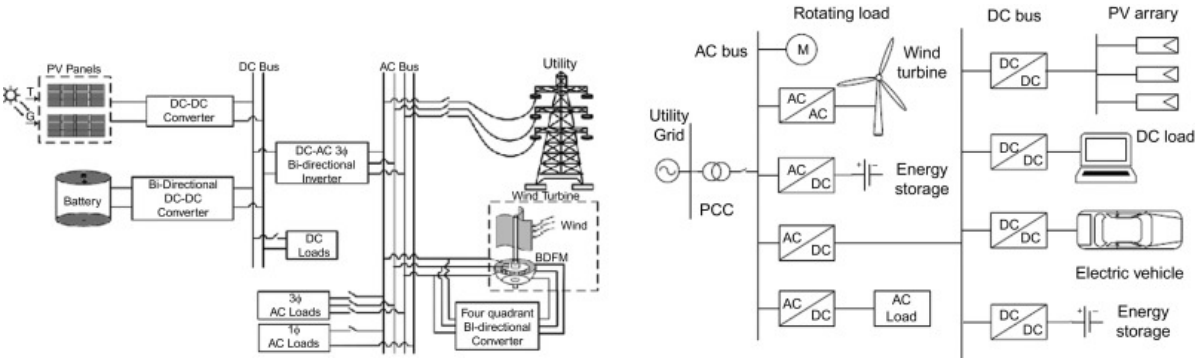


Figure 2.4: AC/DC hybrid microgrid (Ali et al., 2022)

to the strategies employed for energy management and distribution, playing a significant role in the grid's efficiency and stability (Sinha & Kanwar, 2024).

2.4.2. Components

The components in a microgrid can include entities such as (distributed) energy sources, ESSs, loads, and the utility grid (Nakabi & Toivanen, 2021). Depending of the objective of the grid and the specific requirements for the energy demand of the location a combination of components is to be put together that is able to meet the objectives. In Figure 2.5 a conceptualization is depicted of a microgrid capable of both isolated and grid-connected operations, including components such as energy sources, generators, energy storage, loads, and the utility grid. Meanwhile, Figure 2.6 presents a schematic for a test microgrid setup, notably disconnected from the central grid, thereby illustrating the adaptability and possible independence of microgrid configurations.

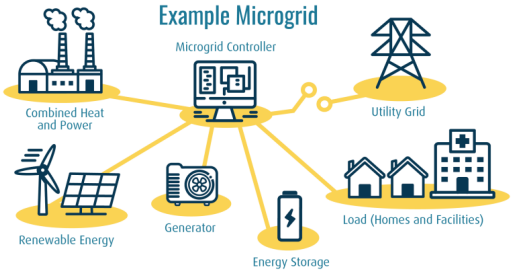


Figure 2.5: Conceptualization connected microgrid (NASEO, 2023)

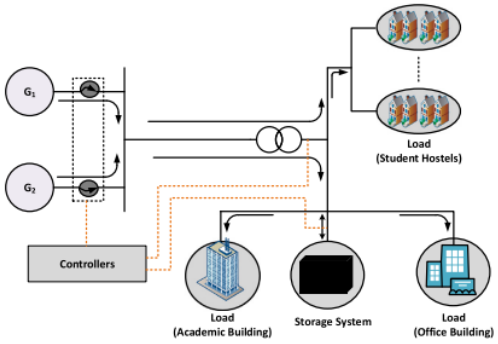


Figure 2.6: Conceptualization isolated microgrid (Uddin et al., 2023)

In an impact analysis by Stephan Brandligt (2024) the impact of electricity installations on the main grid is researched and presented, see Table 2.1 for a selection of the researched installations that are relevant for the scope of this research. The different installation types are compared on their peak generation moments, in order to analyze the impact of the current Dutch grid congestion that is expected to grow, see Chapter 1. It can be seen the current grid is largely unprepared for the high and medium demand installations, indicating a need for significant upgrades or changes in consumer usage patterns.

Solar panels and heat pumps, which are indicated to have a medium impact on the grid, can already be effectively integrated into microgrids. As they include ESSs that can store excess energy during low demand and distribute it during peak times, lessening the burden on the central grid. In anticipation of a 'future-proof' grid that can support a wide range of new installations, it is essential to adopt renewable energy sources and develop systems that can manage simultaneous high demands. Preparing for the future grid involves accommodating renewable energy sources and shifting towards systems that offer high simultaneity without overwhelming the grid, thus ensuring reliability and efficiency. The components are further explored in more detail in Section 2.5.

Table 2.1: Impact of various household installations on the electricity grid (Stephan Brandligt, 2024)

Installation	Peak Power	Simultaneity	Impact on Grid	Current Grid Ready	Future-proof Grid Ready
Electric Vehicle	Medium/High (11 kW)	Medium/High	Medium/High	No	Yes
Water Heat Pump (ground/surface/very low temp network)	Medium/Low (2-5 kW)	High	Medium	No	Yes
Air Heat Pump	Medium/Low (3-6 kW)	High	Medium	No	Yes
Hybrid Heat Pump	Low (1-3 kW)	High	Medium	Partially	Yes
Solar Panels	Medium (4-8 kW)	High	Medium	No	Yes
District Heating on residual heat or geothermal	None (0 kW)	High	None	Yes	Yes

2.4.3. The Role of Geographical Information (Systems) in Microgrid Design

Geographical Information can support the efficient composition and configuration of microgrid components. GIS tools can analyze spatial data to determine the optimal locations for DERs, such as solar panels and wind turbines, based on factors like solar irradiance, wind speed, and land use patterns. By mapping out these resources, the use of GIS can help in identifying the best sites for energy generation to maximize efficiency and output.

Moreover, GISs can assist in planning the layout of ESSs and other infrastructure within the microgrid. By evaluating spatial data on load centers, transportation networks, and existing utility grids, GISs can facilitate optimization of the placement of these components to ensure effective energy distribution and minimal transmission losses. Spatial optimization is particularly important in urban areas where space is limited and energy demand is high.

Furthermore, GISs can help enable accurate load simulations by making use of the open-source geographical data on buildings. Data includes information on building size, type, occupancy, and energy consumption patterns. By integrating these datasets, geographical information can help model energy demand profiles for different types of buildings within the microgrid.

In summary, GIS can provide support and tools for microgrid design, implementation, and operation by providing spatial insights and optimizing the placement and management of components.

2.5. Distributed Energy Resources

In the previous subsection common components of microgrids are briefly discussed that can take part in the setting up of a microgrid. In this section a more in depth analysis of components that are suitable for urban microgrids is provided by means of literature. First, the renewable energy sources are explored. Second, the role of EVs in a microgrid is discusses, and thereafter the different sorts of batteries are presented. Lastly, the role of the loads and the access to utility grid is considered.

2.5.1. Urban microgrids and common renewable energy sources

The most prevalent renewable energy sources globally in the power sector are wind, solar, hydro power, bio energy, geothermal, and marine/ocean energy (Al-Shetwi, 2022).

Urban microgrids are influenced by factors such as safety and spatial constraints. Consequently, microgrids in urban areas typically utilize smaller wind turbines or solar PV systems (Nakabi & Toivanen, 2021). These renewable sources are often supplemented by generators or the main grid.

Solar panels are commonly installed on rooftops, while solar parks are located in more open areas. SolarPower Europe reports that "when it comes to solar power per capita, Europe's long-time solar leader, Germany, does not hold the first position. For the second year in a row, the Netherlands ranks

first" (SolarPower Europe, n.d.). Additionally, "in 2022, the biggest market segment in the Netherlands was the residential rooftop market, with a share of 46% (approx. 1.8 GW) of the total market" (SolarPower Europe, n.d.). The total solar generation over the last 12 months, with its seasonal variance, is presented in Figure 2.7.

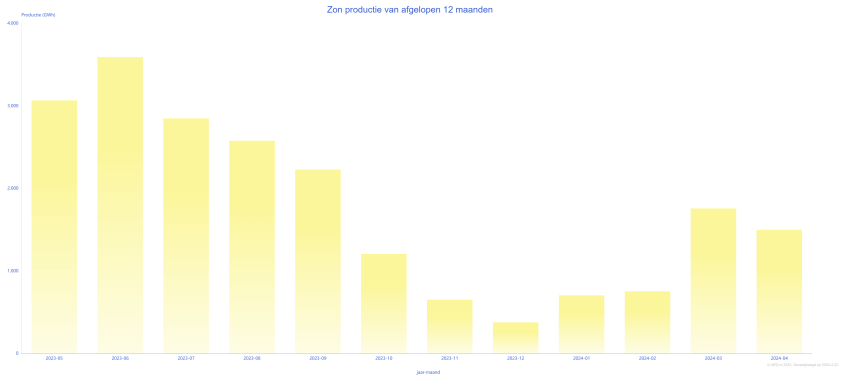
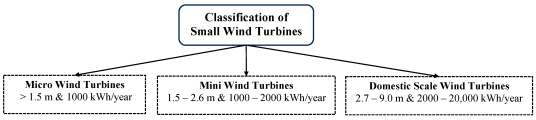
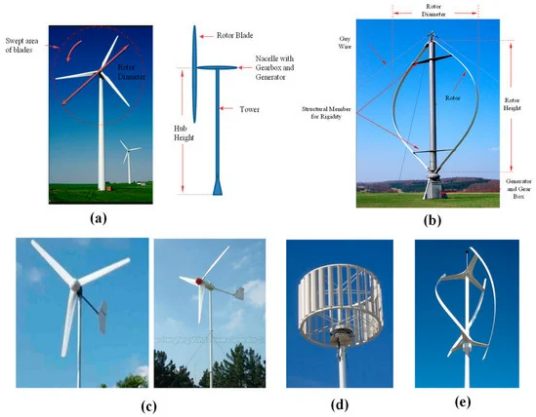


Figure 2.7: Generated solar energy over the last 12 months (Nationaal Energie Dashboard, 2024)

For the wind energy generation, the profile is less patterned. The generated energy varies throughout the day. The wind in the Netherlands can pick up just as quickly as it can die down (EBN, 2023). Domestic scale wind turbines can vary from 2.7 to 9 meter in their diameters, see Figure 2.8. Alam and Jin (2023) explains how "small wind turbines up to 10 m tall and with a diameter under 7 m are exempt from some regulations in several countries". Furthermore, the Netherlands only has guidelines and laws for noise restrictions. According to the Nl and Blezer (n.d.), the spatial guideline is to have a minimum space of three times the diameter between two motors, meaning that there should be buffer around the center of a turbine of three times its size.



(a) Small Wind Turbines



(b) (a) Horizontal Axis Wind Turbine, (b) Vertical Axis Wind Turbine, (c) Domestic HAWTs, (d) Domestic VAWT, (e) Domestic VAWT

Figure 2.8: Comparative illustrations of small wind turbines taken from Alam and Jin (2023)

In order to manage the fluctuation of these weather dependent sources, good estimations are of importance. One method for forecasting renewable energy generation involves leveraging historical data and input parameters, utilizing past observations, such as solar irradiance and wind conditions, to predict future energy outputs. According to Israr and Yang (2021), integrating historical data with current variables like solar irradiation intensity, the orientation of solar panels, and wind characteristics is essential for accurately predicting renewable energy generation.

An emerging element in sustainable energy systems is green hydrogen, produced by water electrolysis using power from renewable sources. This technology is poised to facilitate emission reductions in sectors that are traditionally challenging to decarbonize. While the prospects for green hydrogen are

promising, its current production costs are higher than fossil fuel alternatives, which remains a barrier to widespread adoption (Hodges et al., 2022).

Strategic placement of hydrogen electrolyzers could address renewable energy curtailment issues caused by grid stability and network constraints. These devices can absorb surplus energy, thereby acting as local loads and preventing waste of generated renewable power (Lasantha Meegahapola, 2023).

Despite being in a developmental phase, hydrogen's role in the energy landscape is expanding. The Energy Technology Perspectives Clean Energy Technology Guide provides a detailed overview of renewable energy sources, tracking their development from initial ideas to fully operational systems (IEA, 2023). Within this overview, hydropower and induction cooking for example are marked as mature technologies, while nuclear fusion is an aspirational concept described as a potential source of base-load electricity with negligible CO2 emissions and a nearly inexhaustible fuel supply (IEA, 2023).

When analyzing the renewable energy sources and their impact when incorporated into microgrids, the ETP Clean Energy Technology Guide can be very useful. It may informed estimations of what is possible now and what might be feasible in the future. From the 550 included sources 121 are for buildings and urban environments, and 182 are for energy transformation. Other sectors that are taken in are industry, transport, and CO_2 management.

Considering the urban context and factors like scalability and compatibility, options like hydrogen blending in existing natural gas networks, floating solar PV, hydrogen boilers, and hybrid heat pumps seem more immediately feasible than other options in the guide. These sources are therefore further explored. The entire selection process and the other concepts are presented in Appendix A.

2.5.2. Hydrogen Integration

Hydrogen technology is set to be integrated into the energy systems. A significant development in this area is the integration of hydrogen into existing natural gas networks. Gasunie (2024) outlines the high investment costs associated with this transition but predicts its realization in the coming years. A proposed hydrogen utility network spanning 1200 kilometers is in the works, with 1000 kilometers using the existing natural gas infrastructures.

Complementing this, Power-to-Gas technology plays a crucial role by using surplus renewable energy to produce hydrogen through water electrolysis. This method not only allows for the effective storage of renewable energy but also ensures a steady supply of hydrogen for various applications, including those in CHP systems, heat pumps and boilers (Yu et al., 2023).

In these CHP systems, hydrogen can be utilized in modified internal combustion engines to simultaneously generate electricity and heat with minimal emissions. Moreover, the proximity of hydrogen production facilities to CHP systems can significantly reduce transportation and storage costs, enhancing overall efficiency and cost-effectiveness (Gasunie, 2024). Hydrogen-specific technologies such as hydrogen boilers and hybrid heat pumps also gain relevance with these developments. These systems are capable of using hydrogen blended into natural gas pipelines to provide efficient and low-emission heating solutions.

This shift toward hydrogen integration makes technologies like hydrogen boilers and hybrid heat pumps increasingly relevant. Hydrogen boilers will be able to utilize the blended hydrogen in natural gas pipelines, providing heating without high carbon emissions.

Finally, hydrogen fuel cells in CHP systems offer a highly efficient means of converting hydrogen into electricity, operating at lower temperatures than combustion engines and surpassing the performance of systems based on fossil fuels in terms of efficiency and emissions (Yu et al., 2023).

Fuel cells in general are recognized for their high efficiency and minimal emissions and are seen as potential substitutes for traditional engines and turbines, despite facing higher production costs at the moment. Blending hydrogen with other fuels to create hybrid fuels is an interim approach that bridges the transition from fossil fuels to hydrogen (Yu et al., 2023).

2.5.3. Batteries and Battery Energy Storage Systems

Over the past decade, batteries have been at the forefront of energy storage technology. Extensive research has been conducted on their various types, applications, characteristics, and how they can be optimized and programmed, especially in the context of microgrids (Zarate-Perez et al., 2022). An overview of battery characteristic ranges is given below in Table 2.2.

A Battery Energy Storage System (BESS) serves as a fundamental voltage device that helps stabilize power fluctuations and maintain balance in the energy system (Rostami et al., 2020). They make it

Table 2.2: Summarized technical characteristics of storage technologies in microgrids (Zarate-Perez et al., 2022)

Characteristic	Range or Value	High Performance Examples	Common Uses
Specific Energy (Wh/kg)	10–250	Lithium-ion, Sodium Sulfur	Portable devices, electric vehicles
Specific Power (W/kg)	75–300	Nickel metal hybrid, Lead acid	High power applications
Efficiency (%)	65–95	Lithium-ion	Energy storage systems
Service Life (years)	5–20	Sodium Sulfur, Nickel-cadmium	Long-term energy solutions
Self-discharge Rate (%)	0–High	Redox flow (low), Nickel metal hybrid (high)	Energy storage when not in use

possible to deal with the intermittency of renewable energy sources by storing the excess energy (Israr & Yang, 2021). When developing such a system there are a variety of factors to be considered, such as reliability, battery technology, power quality, and frequency variations along with environmental impacts (Zarate-Perez et al., 2022). Operational constraints like the State of Charge (SOC), battery life, and degradation rate also play critical roles in optimizing BESS functionality. Additionally, battery aging, influenced by factors such as temperature, voltage, and the cyclical nature of charging and discharging, must be accounted for in the total cost and efficiency calculations of the system (Rostami et al., 2020; Zarate-Perez et al., 2022).

2.5.4. Electric Vehicles: Sink and Source

The integration of EVs with renewable energy sources offers substantial opportunities to enhance the adoption of both technologies. Charging EVs with renewable energy not only supports the transition to green transportation but also reduces the demand from the main grid. EVs can serve a dual role; they not only act as consumers of clean energy but also as dynamic ESSs through vehicle-to-grid (V2G) services (Hannan et al., 2022; Richardson, 2013). This capability enhances grid flexibility and offers economic benefits both to the grid operators and EV owners by allowing EVs to charge during times of renewable energy surplus and supply energy back to the grid during peak demand periods (Ali et al., 2022).

Smart Charging for Peak Reduction

The landscape of EVs is varied, encompassing fully electric, hybrid, and fuel-cell vehicles (Singh et al., 2019). Each type presents unique demands and benefits concerning grid integration and renewable energy utilization. Uncontrolled EV charging, often dictated by daily mobility patterns, can lead to significant peaks in electricity demand. These peaks are influenced by routine activities, such as home charging overnight or workplace charging during the day (Shepero & Munkhammar, 2018).

Power system support algorithms are designed to optimize charging schedules based on grid demands and renewable energy availability, thus facilitating a more responsive and resilient grid infrastructure (Wang et al., 2014). For example, Fachrizal et al. (2020) has illustrated daily EV charging patterns alongside typical renewable energy production, as shown in Figure 2.9, highlighting the interplay between renewable energy availability and EV charging demand.

Smart charging is defined by Richardson (2013) as "The idea behind smart charging is to charge the vehicle when it is most beneficial, which could be when electricity is at its lowest price, demand is lowest, when there is excess capacity, or based on some other metric". Smart charging strategies, which intelligently adjust the charging times and rates of EVs based on grid conditions and renewable availability, have been highlighted in numerous studies for their potential to reduce peak loads (Fachrizal et al., 2020). Furthermore, a study by Hannan et al. (2022) shows that a smart charging algorithm for plug in vehicles is able to provide islanded microgrids with enough power to survive critical loads without any direct power connection between the microgrid. These strategies are crucial in a landscape where vehicles are parked and potentially connectable to the grid 95% of the time, ready to either draw energy or supply it back depending on real-time needs and conditions (Mwasilu et al., 2014).

Smart charging can thus significantly enhance the efficiency and sustainability of urban energy systems, and employing renewable energy to charge EVs presents a great potential for increased penetration of EVs as well as renewable energy sources (AbuElrub et al., 2020).

The use of renewable energy to charge EVs not only result in green transport but also help to reduce the energy demand on the main grid and prevent or reduce costly grid upgrades. EV present a very critical and effective use of renewables where the charging energy can be extracted from such sustainable sources, or it can be used as sink serving as storage unit presenting V2G services improving grid's flexibility, known as V2G (Hannan et al., 2022). The concept of V2G could encourage EV owners to

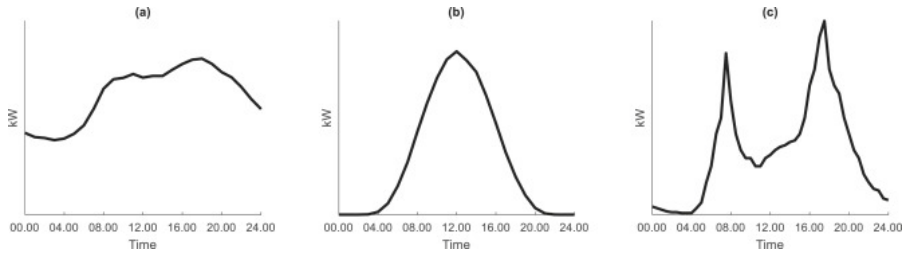


Figure 2.9: Electricity load profile in Sweden on a typical day juxtaposed with PV power production and city-scale EV charging load shape taken from Fachrizal et al. (2020)

use their EVs' batteries for charging at times when there is surplus of renewable energy and use them as storage to supply power to the grid at times of peak demand. In this way, EVs provide economic benefits to the grid as well as to the EV owner (Ali et al., 2022).

Challenges Presented by Electric Vehicle Integration

The nonlinear behavior and dynamic characteristics of EV charging can introduce several challenges to the electrical grid (Ali et al., 2022). Factors such as the charging current, the initial state of charge of the battery, the penetration level of EVs, and their distribution within the power system significantly influence the grid's stability and operational efficiency (Ali et al., 2022; Richardson, 2013). These dynamics necessitate the development and implementation of specialized charging algorithms that can mitigate negative effects and support grid operations.

Charging points

EVs utilize various types of charging points, which can broadly be classified into three categories based on typical usage locations: Work, Home, and Other (Shepero & Munkhammar, 2018).

Charging power demands are presented in an overview by Arias et al. (2017), see Figure 2.10.

Optimizing charging points of EVs is much discussed in literature, as it influences the performance of the smart charging. It comes forward in Arias et al. (2017), Fachrizal et al. (2020), and Shepero and Munkhammar (2018).

2.5.5. Loads

The load management in microgrids is thus crucial due to the variability in demand, characterized by peak loads and other temporal fluctuations. There are primarily two types of loads: critical loads, which are essential and must always be met, and controllable loads, which can be adjusted or deferred during supply constraints, such as standby devices and daytime lighting (Parisio et al., 2014). Balancing these loads effectively within the grid can be achieved through sharing energy between entities with different load profiles, which eliminates the need for additional storage or generation, thereby easing the pressure on the grid.

To optimize load sharing within a microgrid, clustering techniques can be utilized to group buildings or parcels based on their energy consumption profiles (Rostami et al., 2020). This approach is frequently used in the formation of multiple microgrids but can also be applied within a single microgrid to dynamically create optimal energy balance at each time step. In terms of specific clustering methods, hierarchical clustering comes forward as the best method for clustering microgrids.

2.5.6. Utility Grid

The utility grid, as earlier mentioned, can function as back up for when the demand is not met by the DERs in the microgrid. However, in order to release burden of the grid, the microgrid can also buy energy of the grid when it has an overload. In this way it receives cheap energy and also relieves the grid. The demand of the main grid is also in line with the aggregate peak hours. Therefore, the costs are higher on those peak hours. Also, taking off energy off the main grid during overload might even be free of costs.

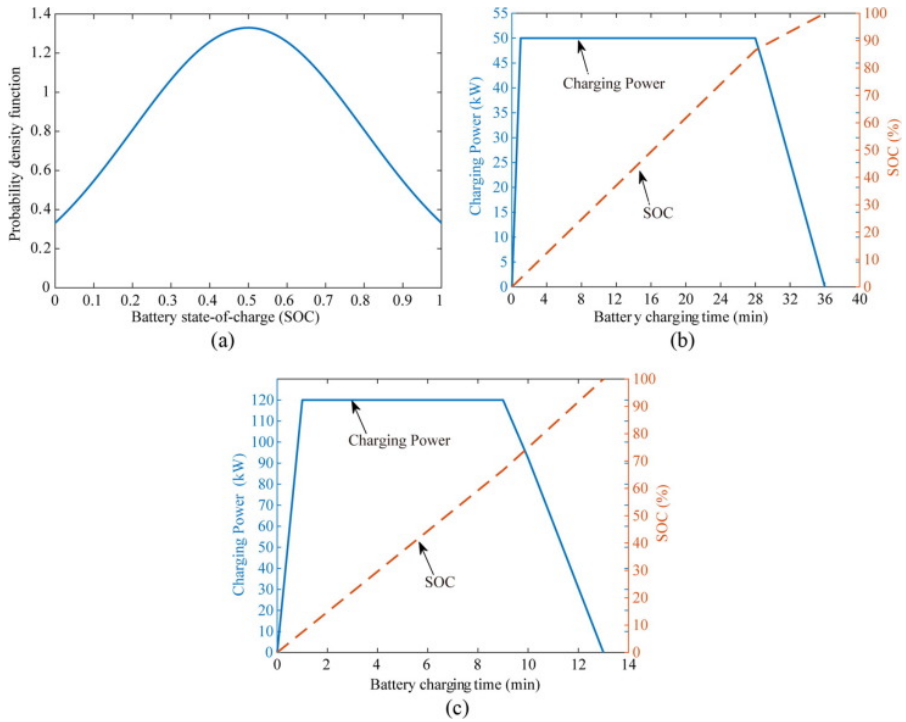


Figure 2.10: Battery EV characteristics: Battery characteristics. (a) Gaussian distribution of the remaining available battery SOC. (b) Battery-charging profile of a lithium-ion polymer battery with 50-kW charging power. (c) Battery-charging profile of a lithium-ion polymer battery with 120-kW charging power taken from Arias et al. (2017)

2.6. Used Optimization Methods

Optimization refers to the process of deriving the most favorable outcomes from available resources (Ifaei et al., 2023). A wide range of algorithms is used in the literature to approach the optimizing of microgrids with integrated renewable energy sources. This reflects the wide range of considerations including system size, grid connectivity, technology integration, cost factors, and automation levels.

In a significant state-of-the-art review by Tajjour and Singh Chandel (2023), an analysis of 170 papers on Energy Management Systems (EMS) for solar microgrids spanning from 2011 to 2023, including 20 review papers, offers an insightful overview of the currently used optimization methods. This review highlights how microgrid management systems vary, influenced by several factors such as system size, grid connectivity, technology, costs, and the level of automation. An overview of all approaches in this review is given in Figure 2.11. Tajjour and Singh Chandel (2023) suggest that the Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and RL, are well-suited for addressing challenges in decentralizing electricity grids. In another review into heuristic algorithms for the matter by Ifaei et al. (2023), also the GA and PSO come forward as highly popular. These algorithms are particularly noted for their ability to handle the complexities and variabilities associated with renewable energy systems, making them suitable choices for efficient and effective microgrid management and optimization.

Hossain et al. (2023) offer a categorization of optimization approaches for Volt-Var Optimization, distinguishing between classical optimization, population-based search techniques, and learning-driven intelligent methods. They note that while numerical optimization methods like mixed-integer linear programming are common, they can be computationally intensive and less efficient for real-world applications due to increased complexity from detailed controller design and large search spaces. Heuristic methods like GA and PSO are again said to be effective for complex scenarios. However, Hossain et al. (2023) explains they may become less efficient with an increase in decision variables.

Recent advancements in microgrid optimization emphasize the superiority of RL and DRL over traditional heuristic algorithms. Tajjour and Singh Chandel (2023) note that RL's key advantage lies in its robust decision-making and stable convergence capabilities, even when faced with numerous variables. What sets RL apart is its consistent ability to achieve optimal solutions, an attribute well-supported by a variety of studies (Klemm & Wiese, 2022; Li et al., 2023; Subramanya et al., 2022).

These advanced learning methods represent a significant move towards model-free or data-driven management approaches (Chen et al., 2022). Nakabi and Toivanen (2021) highlight that learning directly from a microgrid's operational data allows these systems to fine-tune control strategies without the need for an explicit system model. This advantage is twofold: it enhances EMS scalability and reduces maintenance costs, marking a shift towards more autonomous and intelligent energy management systems.

The adaptability of RL and DRL is especially crucial in the dynamic environment of microgrid operations (Chen et al., 2022). Israr and Yang (2021) writes the following about it: "In this context, the adoption of RL and DRL strategies presents a viable path forward. These learning methods can adapt to the dynamics of the system iteratively, solving the control problem effectively without requiring pre-existing knowledge of the system."

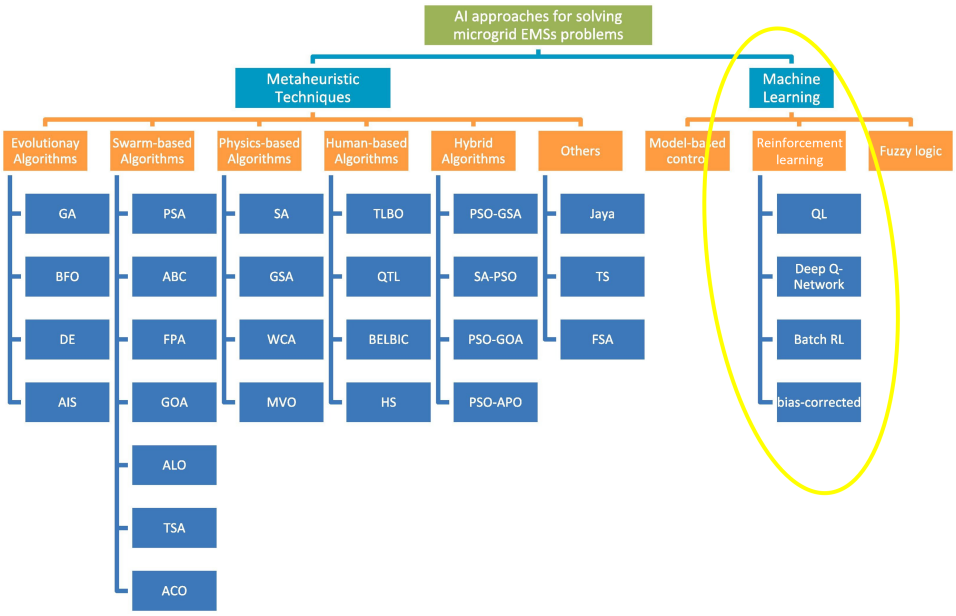


Figure 2.11: Adjusted scheme for AI approaches for solving microgrid EMSs problems (Tajjour & Singh Chandel, 2023)

2.7. Reinforcement Learning

Figure 2.12 illustrates a flowchart of reinforcement learning (RL), which is an iterative process where an agent continuously learns from its interactions with the environment. The objective is to develop a policy that maps states to actions to maximize rewards. The application of this framework to microgrids is discussed below.

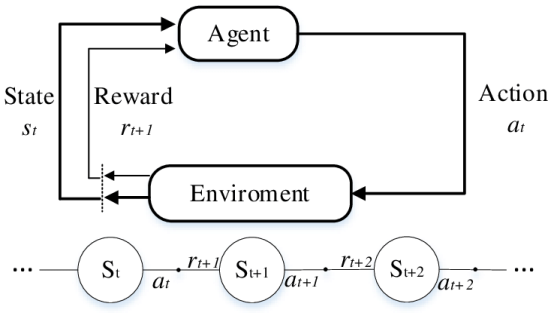


Figure 2.12: Flowchart Reinforcement Learning

The RL approach encompasses a range of control strategies where an agent acquires the best policies for control through direct interaction with its surroundings (Nakabi & Toivanen, 2021). Nakabi and Toivanen (2021) lists several works that have shown the effective implementation of reinforcement

learning-based Energy Management Systems across various microgrid configurations. The agent evaluates the effectiveness of actions based on previous interactions with the environment. Lastly, Van et al. (2023) present a juxtaposition of deep learning implementations. The implementations can show near-optimal performance, without knowing the future demands (Wu et al., 2020).

2.7.1. Markov Decision Process

Markov Decision Processes (MDP) provide the mathematical backbone for RL, offering sequential decision-making under uncertainty. MDPs account for the immediate outcomes of current decisions and potential future actions, articulated through a dynamic state and a value function in the Bellman equation (Sutton & Barto, 2018), shown in Equation 2.2.

$$Q^*(s, a) = \mathbb{E}[r + \gamma \max_{a'} Q^*(s', a')] \tag{2.2}$$

Where:

- $Q^*(s, a)$ represents the optimal state-action value function, which gives the maximum expected return starting from state s, taking action a, and thereafter following the optimal policy.
- $\mathbb{E}$ denotes the expected value, indicating that the returns are averaged over all possibilities, weighted by their probability of occurrence.
- r is the immediate reward received after taking action a in state s.
- γ is the discount factor, which represents the difference in importance between future rewards and immediate rewards. It is a factor between 0 and 1.
- $\max_{a'}$ denotes the maximum value over all possible actions a' from the new state s'.
- $Q^*(s', a')$ represents the optimal state-action value function for the next state s' and any action a'.
- s' is the new state after action a is taken in state s.

Reinforcement Learning algorithms use MDPs to learn optimal policies that specify the best action to take in each state to maximize the cumulative reward over time (Kallenberg, 2016). The entire RL framework includes elements such as decision moments, states, actions, transition probabilities, and rewards (Kallenberg, 2016). Each action leads to a reward and transitions the agent to a subsequent state.

A paper by Chen et al. (2023) presents the elements that typically make up an RL model for microgrids.

- **State Representations for Microgrids:** Possible states include current energy demand, available renewable energy, energy storage levels, and grid connectivity.
- **Actions in Microgrid Optimization:** Actions may involve managing power storage, selling energy back to the main grid, distributing energy among loads, and adapting to fluctuating energy conditions.
- **Rewards for Microgrid Management:** Rewards should align with grid management goals, such as minimizing peak-to-average ratios, maximizing renewable usage, ensuring supply-demand equilibrium, and maintaining grid stability.

Currently, there are many variations of RL-implementations, as shown in Figure 2.13. The methods are mainly divided into value-based methods and policy-based methods (Nakabi & Toivanen, 2021). For the value-based methods, a neural network is used that learns the Q-function of each state. For the policy-based methods a probability distribution is used.

2.7.2. Reinforcement Learning Approach for Microgrids

Nakabi and Toivanen (2021) examine the efficacy of various DRL algorithms in optimizing the energy management of a microgrid. They conclude that deep Q-learning algorithms generally offer better stability compared to policy-based methods. This includes implementations such as SARSA, and Double DQN, along with new methods like an enhanced A3C++ algorithm.

Q-Learning is a model-free approach for solving MDPs, based on the notion of a Q-function. The Q-function of a policyπ measures the expected return or discounted sum of rewards obtained from state by taking action first and following policy thereafter. The optimal Q-function can be defined as

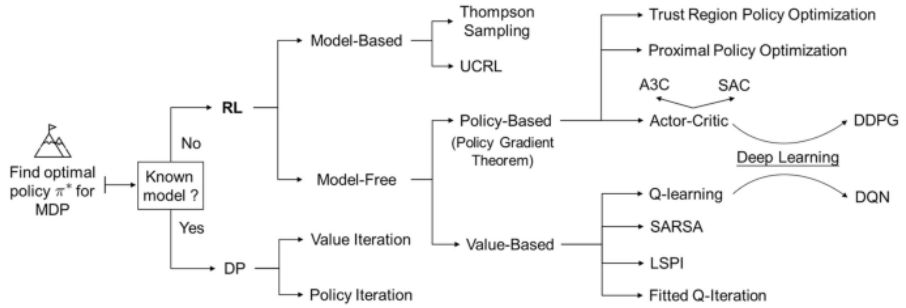


Figure 2.13: An overview of Reinforcement Learning algorithms and required policies by Chen et al. (2022)

the maximum return that can be obtained starting from observation, taking action and following the optimal policy thereafter. The optimal Q-function obeys the Bellman optimality equation, presented in Equation 2.2.

DQN and Neural Networks

When tackling practical problems with large or continuous state and action spaces, and complex system dynamics, traditional value-based RL methods struggle due to the impracticality of computing or storing extensive Q-value tables for all possible state-action pairs. To overcome this limitation, function approximation techniques are employed. These methods utilize parameterized function classes, such as linear or polynomial functions, to estimate the Q-function (Chen et al., 2022). DQN extends Q-learning to continuous state spaces using neural networks as function approximators, though it still requires discrete action spaces.

DQN's ability to deal with discrete action spaces makes it suitable for decision-making scenarios where actions are predefined, such as selecting power flows or switching between energy sources. Chen et al. (2022) demonstrate that DQN significantly improves performance. For instance, a DQN-based EV charging station can reduce charging costs by 77.3% compared to no implementation. However, due to the maximization operations, a maximization bias can occur (Wu et al., 2020). A variation of Q-learning, Double DQN, reduces this bias by using two action-value estimates, $Q1$ and $Q2$ (Wu et al., 2020).

In summary, DQN enhances the management of energy resources within microgrids, facilitating implementations that lead to improved self-sufficiency and reduced reliance on external power supplies.

Minimizing Grid Burden and Maximizing Independence

Minimizing the grid burden involves reducing the reliance on the main grid. This can be achieved by optimizing the energy balance within the microgrid, thereby decreasing the energy drawn from the external grid. By strategically managing the energy flows and utilizing local renewable sources, the dependency on external power supplies can be minimized.

Minimizing Distances for Energy Balancing

Introducing distance considerations into the RL paradigm complicates the optimization process. The action space, representing all possible actions the RL agent can take, increases exponentially when considering shortest paths. This complexity arises from the combinatorial nature of calculating the shortest distances between various nodes in the system, where the action space can potentially grow factorially with each additional node (Almasan et al., 2022).

All elements in microgrids exchange energy with each other. However, larger microgrids introduce higher distribution losses. Reducing these distances can enhance overall performance. As discussed in Subsection 2.5.5, clustering can facilitate optimal load sharing and inform the RL algorithm to factor in distances between loads and generation sources. By integrating clustering results, microgrids can be strategically sized to reduce line lengths and minimize technical losses. These dynamic clusters would thus change topology over time.

A challenge arises when combining this concept with a DQN-trained RL agent, as they are not designed to handle changing topologies. Almasan et al. (2022) highlight that current advancements in DRL for networking are limited to network configurations encountered during training and struggle with unfamiliar structures. This limitation stems from the reliance on conventional neural network

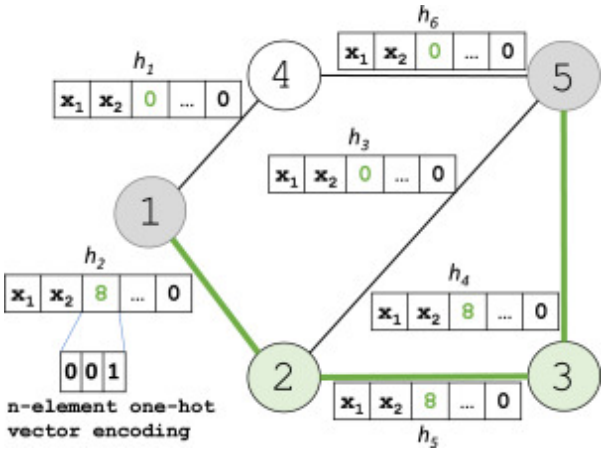
architectures, such as densely connected layers, which do not effectively capture the complexities of graph-based data.

Almasan et al. (2022) propose combining DRL with Graph Neural Networks (GNNs) for routing optimization in internet networks. This combination allows for route allocation and training an agent for unknown topologies (Almasan et al., 2022). The architecture they provide can learn and generalize over unknown topologies due to optimization over an additional dimension. However, this approach requires significant computational power, and the input network of nodes and action space is limited. In their research, the agent is trained in scenarios with a single topology of 14 nodes (Nsfnet) and then analyzed in larger topologies up to 100 nodes. Performance dropped by 15% in these larger topologies. See Figure 2.14 for the algorithm for the DRL-GNN agent and the action space representation.

Algorithm 2 DRL Agent operation

```
s, src, dst, bw ← env.init_env()
reward ← 0, k ← 4, agt.mem ← { }, Done ← False
while not Done do
    k_q_values ← { }
    k_shortest_paths ← compute_k_paths(k, src, dst)
    for i in 0, ..., k do
        p' ← get_path(i, k_shortest_paths)
        s' ← env.alloc_demand(s, p', src, dst, dem)
        k_q_values[i] ← compute_q_value(s', p')
    q_value ← epsilon_greedy(k_q_values, ϵ)
    a ← get_action(q_value, k_shortest_paths, s)
    r, Done, s', src', dst', bw' ← env.step(s, a)
    agt.rmb(s, src, dst, bw, a, r, s', src', dst', bw')
    reward ← reward + r
    If training_steps % M == 0: agt.replay()
    src ← src'; dst ← dst'; bw ← bw', s ← s'
```

(a) Algorithm Agent (Almasan et al., 2022)



(b) Action Space representation (Almasan et al., 2022)

Figure 2.14: Detailed views of (a) a DRL-GNN Algorithm Agent and (b) an Action Space representation.

Hierarchical Clustering

Hierarchical clustering algorithms utilize a heuristic approach based on a distance matrix. The algorithm's inherent simplicity and computational efficiency are significant benefits, especially when dealing with extensive datasets (Cheong et al., 2017). The versatility of hierarchical clustering is a key advantage, particularly in scenarios involving sparse and heterogeneous data typical of remote or isolated locations. Its ability to adapt to various datasets allows for the formation of dense, isolated clusters, which are optimal for microgrid configurations in these areas (Cheong et al., 2017).

Furthermore, hierarchical clustering employs similarity measures or distance matrices that facilitate the visualization of data relationships through dendrograms. These visual representations help in understanding the connectivity and structure within the data, presenting a clear hierarchical linkage between elements (Ediriweera & Widanagama Arachchige, 2022).

This clustering can be used to organize the microgrids strategically to optimize energy distribution. The resulting clusters can act as isolated units that manage their net load independently, which is particularly crucial for balancing the energy within the main microgrid system. Each cluster produces a specific net load that corresponds to the unbalanced demand within that cluster, connecting back to the main grid in various configurations. These configurations can range from fully interconnected systems where every cluster is connected to every other, to arrangements such as a circular or linear chain, depending on the specific energy and infrastructure needs (Ediriweera & Widanagama Arachchige, 2022).

This clustering approach not only optimizes the physical distances between energy production and consumption points but also minimizes transmission losses, thereby enhancing the overall efficiency of the microgrid system.

Objectives and Rewards
The reward for the reinforcement learning model is very important. If the reward is not set up correctly, the model might not learn the correct behavior and therefore the model will not perform like desired. An example of the creation of a reward function is provided in (Nakabi & Toivanen, 2021). Here the reward function aims to maximize the economic profit from operations: "The reward is calculated as the gross margin from operations i.e. the revenues made by selling electricity to the microgrid and to the external grid, minus the costs related to power generation, purchases, and transmission from the external grid." (Nakabi & Toivanen, 2021).

Uncertainty
The earlier mentioned uncertainties inherent in the energy supply and demand sides are critical factors in the development of robust and efficient reinforcement learning models. To address these challenges, probabilistic models and forecasts are essential tools. Additionally, as suggested by Kim et al. (2022), adjusting the discount rate—employing a higher or progressively increasing discount factor for variables that introduce more uncertainty—can enhance model performance. Their implementation of Proximal Policy Optimization demonstrates that, depending on the environment, "a slightly lower discount factor could yield a higher performance than a higher discount factor."

In power systems, particularly when considering renewable energy resources and load demands, the unpredictability of weather-dependent elements like wind turbines and photovoltaic systems necessitates sophisticated modeling techniques. Rostami et al. (2020) emphasize the importance of incorporating probabilistic analysis to manage these uncertainties effectively. Gaussian models, among other probability distribution functions, are frequently employed due to their capability to handle continuous data and provide a good approximation of natural variability. With these probabilistic models, it is possible to better predict and control the fluctuating outputs from WTs and PVs, thereby enhancing the reliability and efficiency of power systems operations.

2.7.3. Performance of Reinforcement Learning Models
Assessing the performance of RL algorithms and models, can be difficult as the answer is not defined and the situations described and created are partially simulated. One measure on how to do this is by the cumulative reward, which quantifies the total reward accumulated by the model over time, indicating its proficiency in executing the task it was trained for (Sutton & Barto, 2018). However, understanding a model's convergence rate—the speed at which it reaches a near-optimal policy—is equally important, as it reflects the model's learning efficiency (Sutton & Barto, 2018). Beyond these, assessing how well a model generalizes to new data is essential for practical applications. This involves testing the model on tasks or scenarios that were not part of its training environment to gauge its adaptability and robustness (Sutton & Barto, 2018). Together, these metrics provide a comprehensive framework for assessing the performance of RL models.

DRL models are highly sensitive to their hyperparameters, and tuning these parameters can significantly affect their performance. While manual tuning is possible, it is often very time-consuming and impractical for complex models. Automated tools like Optuna provide a systematic way to optimize hyperparameters efficiently. Optuna is an open-source hyperparameter optimization framework that automates the search process using advanced algorithms (Akiba et al., 2019). By leveraging such tools, a wide range of hyperparameter values can be systematically explored to identify the optimal settings, thereby improving the model's performance without the extensive manual effort typically required.

As for the training process, there are different approaches that might affect the performance of the model. To capture seasonal variations and other temporal dynamics effectively, it may be beneficial to adopt a cyclic training and evaluation approach. This strategy allows the model to adapt to changes in the environment's dynamics and maintain its relevance across different conditions. It also facilitates the identification of any deviations in performance that could indicate a need for model recalibration or additional training data. Such a structured and iterative evaluation process not only enhances the model's adaptability but also provides continuous feedback loops that are vital for incremental learning and improvement (Sutton & Barto, 2018).

3

Theoretical Framework

In this chapter the theoretical framework resulting from the literature research is presented. The objective of this chapter is to lay down the foundation for the further research by defining which components are part of the bigger system and can be taken in for the optimization of the energy management system of the specific microgrid environment.

3.1. Defining the Energy System Model

The theoretical framework of an energy system model is important to understand its role in energy planning and utilization. The definition and extend of the total management system and the model is determined by its components and the interplay between them. These different types of energy models can guide decision-making by illustrating potential scenarios or clarifying the optimal use of endogenous variables (Farzaneh, 2019). The different types of models that are utilized are mainly distinguished based on two contrasting approaches: the top-down versus the bottom-up, also known as aggregated versus disaggregated, respectively. The latter, as explained by Farzaneh (2019), adopts a process-oriented perspective, taking into account technical details that make it possible for such a model to function supportive in techno-economical rationing (Herbst et al., 2012).

The research questions in this study necessitate a bottom-up methodology, which is inherently data-intensive. This methodology is typified as an "optimization model" that employs a dynamic programming approach mathematically. As discussed in the Subsection 2.1.3, addressing the challenges associated with microgrid management requires sophisticated optimization techniques.

The characteristics defining the energy system model are cataloged according to the parameters used by Farzaneh (2019), as presented in Table 3.1. This approach, however, does not take into account transaction costs that are otherwise encompassed within a top-down approach. Consequently, employing a bottom-up strategy, as executed in this instance, is not ideally suited for exceedingly long-term forecasts, particularly for components with a lifespan shorter than about 20 years (Herbst et al., 2012). The rapid evolution of technological advancements underscores the relevance of this consideration. As this is the case for various elements within the framework, such as fuel cells and electrolyzers, a time frame extending from medium to long-term is considered most appropriate, indicating 20-30 years which leads to the year 2045-2050. This estimation aligns well with the national plans that often focus on implementations by 2050, such as the LAN and the European green deal, see Chapter 1.

The objective of the modeled microgrid in this thesis is to generate insights for microgrids in a gas-less context, thereby contributing to the literature by demonstrating the management of a microgrid driven solely by renewable energy sources and minimizing the burden on the main grid. This approach aligns with a future perspective constructed by Netbeheer Nederland (2023), within which various scenarios are envisioned. The scenarios, recommended by the report to focus on flexibility alongside two other themes—demand and supply development, and the reduction of greenhouse gases (Netbeheer Nederland, 2023; Netbeheer Nederland et al., 2022)—show a wide variety in options, all within expectations. The energy system model reflects a comprehensive approach to understanding and addressing future energy challenges by connecting it to these scenarios. The alignment not only

enhances the energy system model's relevance and applicability to potential future states but also facilitates a nuanced exploration of strategies for managing energy demand and expanding renewable energy integration.

The scenarios make up different topologies to test the setup and analyze the performance of the trained agent. Key elements such as diverse renewable energy sources and EVs are included to model a microgrid configuration for the year 2050, in line with the future scenarios that aim to be fully climate neutral by that time but differ in their approach. Within the boundaries of these scenarios, electrical power, geothermal power, and hydrogen power are identified as potential sources of energy.

Table 3.1: Classification energy system model, based on classification structure (Farzaneh, 2019)

Consideration	Classification
Purpose	Integrated
Modeling paradigm	Bottom-up
Methodological approach	Optimization
Mathematical approach	Dynamic programming
Spatial perspective	Local
Time horizon	Medium to long

The definition of the elements in the to be implemented system are given below. The system consists of the environment, the possible states, the actions, and a reward function.

The system to be implemented comprises the following key elements: The environment, which includes the RL environment encompassing the grid with all DERs, loads, generators, and connections to the main grid. The states represent the current status of the grid, detailing outputs of each DER and generator, load demands, the extent of power exchange with the main grid, battery storage levels, and relevant environmental factors like weather conditions affecting renewable outputs. Actions available to the agent include adjusting outputs of controllable DERs, managing ESSs, and deciding on power exchanges with the main grid. The reward function, which encodes the system's objectives includes penalties for using grid power and incentives for using renewable energy, designed to encourage self-sufficiency and minimize operational costs. Furthermore, the reward function must prefer local load balancing in order to minimize travel loss.

3.2. Components Microgrid

In this section, the research framework for further study is outlined, presenting a conceptualization that includes all potential system components and the relationships between them. This framework delineates all the variables critical for the model architecture and optimization. The comprehensive overview details all possible elements, although not every element will feature in the scenarios. The scenarios are constructed using a subset of these elements, specifically chosen to align with governmental plans and expectations.

To illustrate the relationships and dependencies within the energy grid system, a detailed conceptual flow chart is developed. This chart serves as a foundational guide, specifying all relevant variables, performance indicators, and entities as defined in the supporting literature, see Chapter 2. Each component included in the chart is elaborated on in the subsequent sections, ensuring a clear understanding of their roles and interactions within the model.

3.2.1. Energy Loads

Energy loads make up the energy demand. Different residential objects can have very different load profiles. For a building with multiple residential objects, the load shall be the sum over time. The energy loads are the total energy demand, including the part that would potentially currently be sufficed with gas-powered energy. Loads can be balanced between buildings that vary in user profile, thereby minimizing the need to access the 'main roads' of the energy grid.

3.2.2. The Utility grid

The utility grid functions as a fall back option, for when the microgrid is not able to meet the energy demand. It is either connected or disconnected to the utility grid in a binary manner. Due to the variable

and unpredictable characteristics of DERs, it's not always feasible to achieve a balance between supply and demand within the microgrid solely through these resources. In instances of energy shortfall, the main power grid can immediately provide electricity to the microgrid, or alternatively, it can absorb any excess energy when there is a surplus.

The import and export costs of energy from and to the main grid are measured by their real costs, which align with the peak hours of energy consumption. Therefore, fallback to the main grid is strongly penalized.

3.2.3. Distributed Energy Resources

Given the focus of this research on urban environments, the DERs considered in constructing the scenario approach are limited to those powered by solar, geothermal, or bio energy.

Photo Voltaic power

The solar panels in the microgrid absorb Photo Voltaic (PV) power. The generated output is modeled by means of existing weather data for the Netherlands for solar panels in Amstelveen (BRON). With the new input of the Dutch Energy Bank and the associated API, a connection with the solar radiation and therefore the expected solar generation can be made.

Geothermal power

Geothermal power can contribute directly to the supply of thermal energy, which constitutes a significant portion of the overall energy demand. This power can be conveyed using water through repurposed old gas pipelines, see the further explanation below in the explanation about Hydrogen use. The geothermic potential must be analyzed in order to know whether the case study area has access to geothermal power.

Hydrogen and the Utility Grid

Hydrogen is a key energy carrier in various scenarios, despite its current high costs and the uncertainties surrounding its widespread adoption. It holds significant potential for reducing emissions when used as a fuel or raw material in industries. To support this, Hynetwork Services, a subsidiary of Gasunie, is tasked with developing a national hydrogen network. Scheduled for completion by 2030, this network will supply major industrial regions and connect with German and European systems. Utilizing repurposed natural gas pipelines, it aims to seamlessly integrate into the existing infrastructure and facilitate Europe's transition to sustainable energy. In scenarios where hydrogen has been successfully adopted, this network will support extensive distribution and accessibility.

(Hydrogen-Based) Combined Heat and Power

Integrated energy systems can provide a more efficient supply than individual systems by utilizing resources such as cogeneration (Ko & Kim, 2019). A CHP system can operate using both hydrogen and electrical energy, as well as harnessing the heat generated from electrolysis. The CHP has different parameters for different scenarios.

(Hydrogen-based) Heat pump

Heat pumps are increasingly being adopted as a reliable technology. While their efficiency can still be influenced by weather conditions, geothermal heat pumps provide consistent performance regardless of the climate. In the framework both weather dependent, and geothermal pumps are taken in.

Electrolyzer

An electrolyzer can convert a surplus of renewable energy into hydrogen. This process, known as electrolysis, involves splitting water into hydrogen and oxygen using electricity. When renewable energy sources like solar or wind produce more electricity than the system demands, an electrolyzer can utilize this excess to generate hydrogen. Incorporating an electrolyzer enhances the flexibility and resilience of the energy system. It allows for energy storage in chemical form, which can be particularly valuable during periods of low renewable generation or high demand. In the framework, the electrolyzer is implemented as a direct conversion device with a predetermined efficiency to accurately reflect real-world costs.

Fuel Cell
A fuel cell can convert the chemical energy from a fuel such as hydrogen into electricity, heat and byproducts through a chemical reaction with oxygen or another oxidizing agent (Herbst et al., 2012). Herbst et al. (2012) have shown that fuel cells provide cost-competitive, highly reliable, efficient, clean, quiet, contained, modular, scalable and community-friendly energies. They can be divided into four primary market applications: primary power sources, backup power systems, CHP systems, and fuel cell-powered vehicles (Herbst et al., 2012). In the framework the fuel cell is taken in for the fuel cell vehicles, and for CHP systems.

Electric Vehicles
EVs can, apart from functioning as vehicle, function as batteries in the microgrid. By smart management of the EV, they can be used as ESS or a power plant when not used (Bartels et al., 2022). This is called V2G and G2V (AbuElrub et al., 2020). In the modeled system it is assumed that EVs follow a trip pattern where they leave and return to the home base at specific times. During the evening and night, and in the weekend they function therefore as such a ESS. It is assumed that they return to the microgrid with a random SOC.

Fuel Cell Electric Vehicle
Fuel cell electric vehicles (FCEVs) transform the chemical energy from hydrogen into electricity to drive their motors. They carry an onboard fuel source, such as hydrogen or natural gas, and may rely solely on the fuel cell or be part of a hybrid system incorporating a battery, similar to hybrid EVs or plug-in hybrid EVs. As part of the hydrogen driven scenarios, FCVs could play a significant role in transportation. If the hydrogen they use is produced via water electrolysis using renewable energy or derived from biomass, then FCVs are powered by renewable energy sources (Richardson, 2013).

Electric Vehicle battery
EV batteries charge at a fixed pace, which is predetermined based on the battery's parameters: the capacity and the power output of the charging infrastructure. This pace can be optimized to match the grid's demand patterns or to take advantage of lower electricity rates during off-peak hours.

Batteries
Traditional batteries, such as lead-acid and lithium-ion, are integral components in energy storage for grid applications. In the framework the parameters of the batteries that are taken in can differ and determine the efficiency and capacity. The different batteries that can potentially be utilized and characteristics are presented in Table 2.2.

3.3. Conceptualization

Below the conceptualization of the framework is presented, see Figure 3.1. All the components, are encapsulated with the interplay between them. The green lines represent electrical energy flow, the orange lines represent thermal energy flow, and the blue lines represent hydrogen flow. The EMS also entails the model configuration setting that might differ, like the forecast horizon and accuracy.

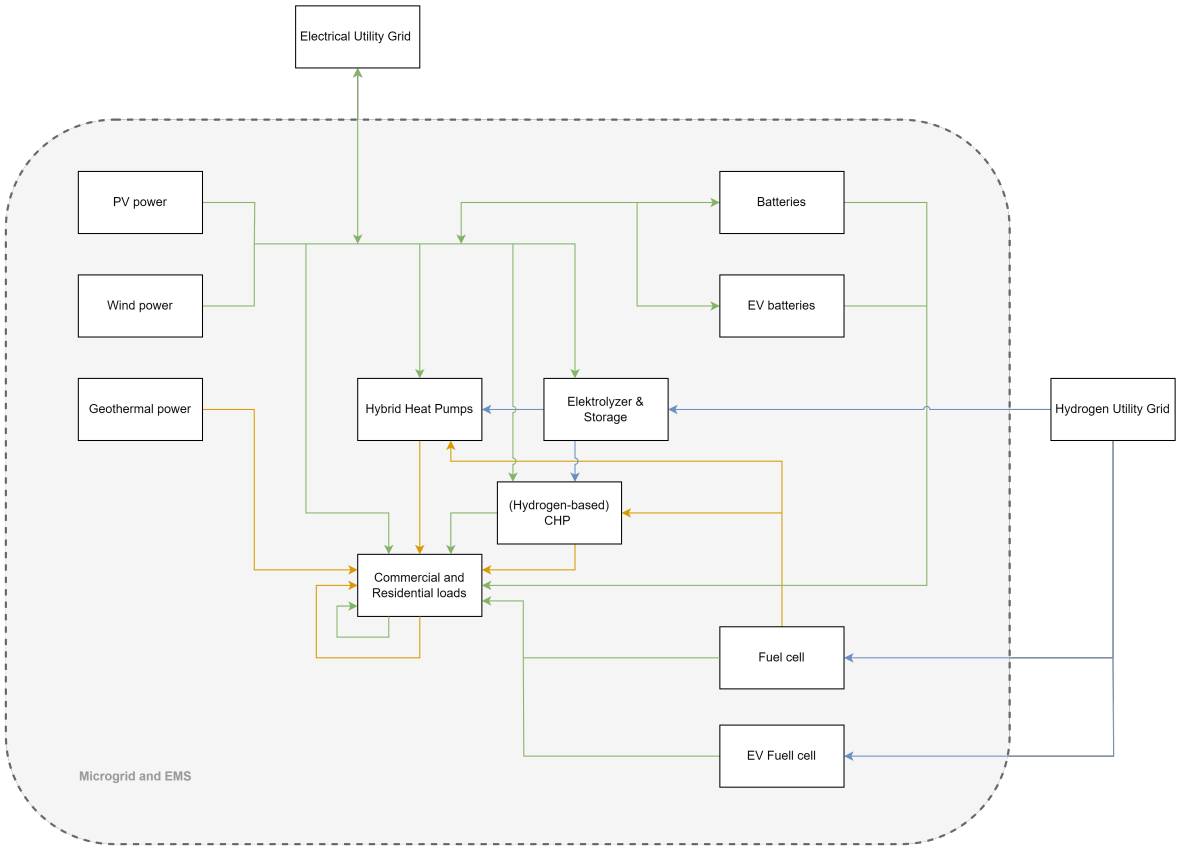


Figure 3.1: System conceptualization | (Green, Electricity), (Orange, Heat), (Blue, Hydrogen)

The conceptualization is in line with the dutch Energy system in transition: Sustainable, supply driven with bidirectional energy flow, decentralized, electrified, and digital and data-driven.

the n

3.4. Scenario Approach

Scenarios are now designed with the objective of maximizing rewards, as presented in Section 3.4. This approach involves crafting a base case scenario and then methodically adjusting environmental variables to ascertain their impact on performance. Key factors under consideration include demand, supply, and storage capabilities. The structure of the scenarios is presented in Table 3.2.

Each scenario aims to identify key levers that significantly influence the energy system's efficiency and effectiveness, thus guiding strategic enhancements.

3.5. Reward Shaping Indicators and Objectives

There are two types of indicators or objectives, namely focused on optimizing financial objectives or enhancing energy efficiency (Subramanya et al., 2022). The objective of the model is to foremostly enhance energy efficiency. The main objective is to minimize grid connection and to minimize travel distances. However, besides those objectives performance indicators also entail financial parameters such as the the investment costs but also the degradation costs. All the KPIs are presented in Table 3.3.

3.6. GNN and Dynamic Clustering: Process Description

From literature research, it is evident that incorporating distances into the action space significantly increases computational demands. This implies the necessity of (too) high computation. These challenges and its implications for system performance are detailed in Chapter 2. In order to be able to take in distances as an objective, another approach is required. This section outlines a novel method that

Table 3.2: Summary of energy system scenarios

Scenario	Description
Base case	This scenario serves as the control setup, where the current energy system configuration is used without any modifications. The focus here is to establish a benchmark for performance comparison.
Demand side decrease	In this scenario, variations in demand patterns are introduced to evaluate how changes in consumer behavior affect system performance.
Increase in Renewable Energy supply	This scenario explores the impact of varying renewable energy supply sources. It considers the effects of utilizing a single type of renewable energy versus a combination of multiple sources, thus providing insights into the benefits of diversifying energy supply.
Increase in (EV) battery storage possibilities	This scenario examines the role of ESSs in enhancing grid resilience and efficiency. It assesses how different storage capacities and technologies can optimize overall system performance.

Table 3.3: The objectives and KPIs of the model

Objective Category	Specific Objective	Key Performance Indicators (KPIs)
Energy Efficiency	Minimize grid connection	Reduction in grid energy consumption in percentage
	(Future work) Minimize travel distances	Reduction in operational distances in meters
	Maximize energy self-sufficiency	Percentage increase in self-generated energy usage
Financial Objectives	Minimize operational costs	Operating cost reduction per kWh

combines GNNs with dynamic clustering to optimize microgrid operations. This approach addresses the computational challenges associated with large action spaces and varying topologies in microgrid management.

While this thesis employs a DQN algorithm, which lacks generalizability across varying topologies, the proposed method introduces DRL+GNN+clustering to create manageable clusters within the microgrid and one agent that can traverse these sub-microgrids. This method allows for the inclusion of distance minimization in cluster formation without expanding the action space excessively.

GNNs are specialized deep learning models that adapt to graphs of varying sizes and structures. This adaptability enables GNN-equipped DRL agents to effectively learn from and apply knowledge to a wide range of network topologies. The proposed method aims to overcome the limitations of the DQN approach by leveraging GNNs to generalize over different topologies. In a paper by Almasan et al. (2022) a routing optimization for internet is performed with a DRL+GNN trained agent, indicating good performance for topologies up to a 100 nodes. The combined DRL and GNN agent is tested by means of testing with a routing optimization problem in optical networks, across 180 synthetic and 232 real-world network configurations that it had not encountered during its training phase. The findings indicate that the DRL + GNN agent surpasses existing leading solutions in these diverse topologies.

Combining the DRL+GNN agent with cluster techniques, can limit the necessary computation and

complexity and increasing the possibility of optimizing over multiple dimensions and topologies. With this novel approach it is possible to take in distance, and therefore to balance on a lower level.

The clustering can already impose strict spatial constraints necessary for efficient microgrid generation, while also limiting the size of the sub-microgrids. The microgrid is divided into several sub-microgrids that exchange energy, aiming to minimize distances and balance loads within the network. Constraints on the maximum number of loads per sub-microgrid ensure system efficiency and stability.

An important consideration in clustering is whether dynamic clustering per time step significantly increases performance or if a longer-term approach, for example annual clusterin, would suffice to determine optimal clusters while still achieving distance minimization.

All in all, integrating GNNs with dynamic clustering can train an agent to better handle the complexity and variability of microgrid environments.

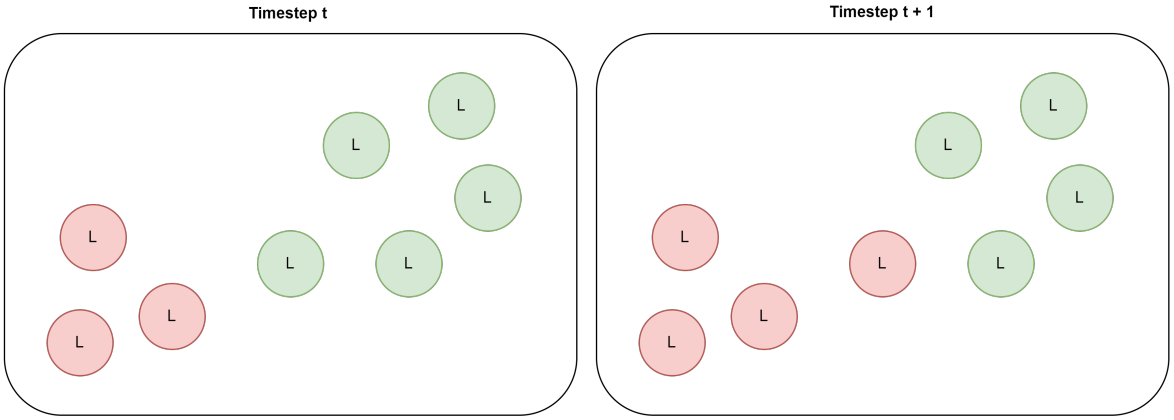


Figure 3.2: Dynamic clustering of sub microgrids

3.7. Case study: Modeling decisions

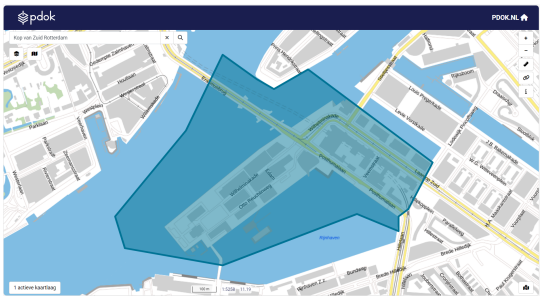
A case study for the model is chosen and designed by means of exploratory research. The case study must provide a clear insight into the key elements and it must enable research into all the objectives. The case study forms the fixed environment in which the flexible scenario, detailed in the Chapter 3, is placed.

This research adopts a local spatial perspective when selecting a specific area that meets a series of criteria essential for the case study. The chosen area must be distinct, support mixed-use functions, and allow for diverse user demographics, ensuring that all buildings within it can engage in spatial energy sharing. Additionally, there must be adequate space for the introduction of new energy solutions, situated within an urban context.

To identify areas that fulfill these requirements, a detailed exploratory analysis was conducted using tools such as the PDOK viewer and the 3DBAG viewer. This exploration led to the selection of the "Kop van Zuid", a neighborhood in Rotterdam, for an in-depth study (PDOK, n.d.; tudelft3d & 3DGI, 2023).

The selected area is characterized by a diverse array of building functions, including commercial, business, and residential uses. Kop van Zuid notable for its extensive panel space and large buildings, which facilitate potential energy interactions due to the large space between structures which leaves space for the placement of new structures. Moreover, the presence of surrounding water bodies distinctly demarcates these areas, providing a naturally intuitive and suitable environment for implementing a microgrid system, as visible in detailed in Figure 3.3a.

(a) Neighborhood "Kop van Zuid" in the 3DBAG viewer (tudelft3d & 3DGI, 2023)



(b) Boundaries of the case study (PDOK, n.d.)

Figure 3.3: Overview of the case study area "Kop van Zuid"

Information on the demographics, the amounts of buildings and more energy and housing-related metrics are gathered. This entails information on the amount of residential objects within the building, and the functions of those objects, see Table 3.4 for a subset for three buildings. The case study is made up out of large buildings with high loads. This makes that a lot of energy is required to suffice these needs.

Table 3.4: Buildings in case study and function (ESRI Nederland, 2024a; Nationaal georegister, 2023)

Attribute	*Building 1*	*Building 2*	*Building 3*
Identification number	...203529	...359215	...432617
Amount of residential objects	2	2	40
Residential		0	0
Office	0	0	39
Meeting Facility	1	0	0
Educational Facility	0	1	0
Retail Store	0	0	0
Sports Facility	0	0	0
Lodging Facility	0	0	0
Healthcare Facility	0	0	0
Industrial Facility	1	0	0
Cell	0	0	0
Other	0	1	1
Industrial Business Building	0	0	0
Green Business Building	0	0	0
Energy Label	C	A	A2+

The scope of the case study is limited to incorporating demand, supply in terms of weather-related energy, geothermal energy, and (EV) battery storage. Hydrogen is excluded for the case study due to its efficiency challenges through gas pipes, which require separate investigation. Future research should estimate hydrogen's potential more accurately and explore its integration into microgrid systems. This focused approach ensures that the study addresses the most pressing aspects of microgrid implementation, while acknowledging the need for future studies to explore additional energy sources such as hydrogen.

Cost numbers, particularly those related to cycles, have been assumed in this study. These numbers are sourced from examples of other PymGrid implementations. The exact determination and precise modeling of these cost figures are beyond the scope of this thesis. However, it is recommended that future work should focus on refining these numbers to enhance the accuracy and reliability of the model.

4

Methodology

This chapter describes the methodology that are used to answer the research questions posed in Chapter 1. The first research question is already handled by means of the previous literature research, and intermediate results are presented in Chapter 3. The focus now shifts to answering the remaining two research questions, *Q2* and *Q3*.

The methodology outlines the steps to be taken in the research, serving as a road map for the study. It systematically details the approach, allowing other researchers to replicate the study in future research. The case studies function as a test environment. A scenario approach is used to test the implementation of the optimization for the microgrid energy management system, and to showcase the performance of the model under various circumstances. The methodology below is used for both case studies to build a working model that aims to reach the objectives set in the introduction.

4.1. Objectives and Reward

The optimization of the energy system aims to minimize the burden on the grid and reduce the costs in the system. This involves minimizing grid connections and balancing loads. These objectives form the main objective function, where weights are determined based on potential scenarios. For example, a decision is made regarding the charging or discharging of a battery and whether that is preferable.

- Burden on the Grid: This is measured by minimizing grid connections and reducing the demand on the main grid.
- Operation costs: This is measured by the costs that are spend for DER usage per timestep, and energy import and export that is required to adhere to the constraints of meeting the varying demand.

To minimize the burden on the grid, the following cost function is used:

$$\min \left(C_{grid} \right) \tag{4.1}$$

To minimize the operation costs, the following cost function is used:

$$\min \left(C_{mg} \right) \tag{4.2}$$

The overall objective function, which serves as the reward shaping function for the DRL algorithm, is formulated as:

$$\min \left(C_{grid} + C_{oc} \right) \tag{4.3}$$

The objective function, Equation 4.3, is the reward shaping function. The DRL algorithm, and its reward, steers the agent to minimize the costs.

The primary constraint in the model is that the load demand must always be met, ensuring no outages occur. If local supply is insufficient, energy must be imported from the main grid. It is assumed that this import capability is always available, ensuring reliability in the energy supply.

Demand must be met

$$\sum_{i \in \text{Loads}} D_i \leq \sum_{j \in \text{Energy Resources}} G_j \tag{4.4}$$

Where:

- D_i represents the demand at the i-th location.
- G_j represents the generation at the j-th generator.
- The summation over i encompasses all the demand points.
- The summation over j covers all the generation points.

This ensures the balance of supply and demand.

In summary, by focusing on minimizing the burden on the grid and reducing costs, the objective function aims to optimize the energy system's efficiency and reliability. The reward shaping function directs the DRL algorithm to achieve these goals, while the constraints ensure the continuous and reliable supply of energy.

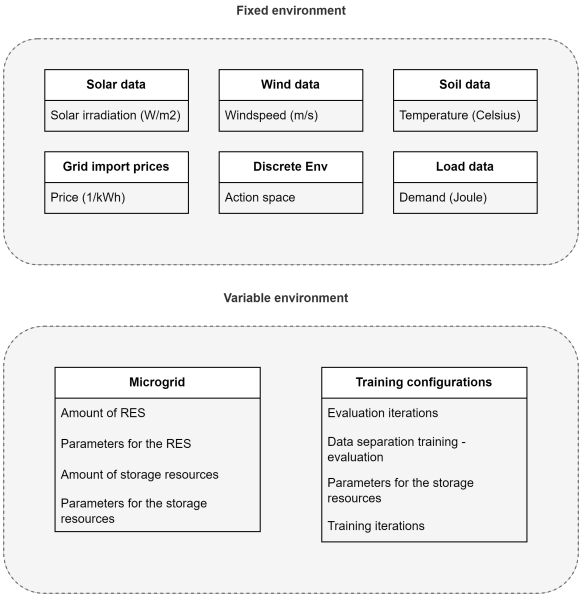


Figure 4.1: Fixed and variable environment

4.2. Model Architecture: Fixed Environment

From the conceptual flow chart, presented in Chapter 3, the model's architecture is developed, focusing on the integration of the determined key elements. Data must be prepared and simulated, and elements and relationships must be formalized. For each case study the static environment is created, and thereafter the changing environment per scenario is constructed. See Figure 4.1 for the overview of the model environment.

For the reason that energy data is only completely available after a complete year, 2022 is the first year to offer a complete overview in terms of data. This year is therefore taken for every dataset. Data has to be found and converted in order to perform the simulation in the consequential step. The data that is needed in order to simulate the loads for the fixed environment entails building information, demographic information, and energy-related metrics.

For the built environment data is gathered from several data sources, namely PDOK (PDOK, n.d., 2024), TNO (TNO, n.d.), and ESRI (ESRI Nederland, 2022, 2024a, 2024b). Data is gathered for the categorization of the buildings. In order to make the simulations energy related data for each building is required. More extensive insights on the complete data used is provided in Appendix B and Appendix C. The flow chart in Appendix C shows the process that is conducted in order to acquire the desired output from the input data.

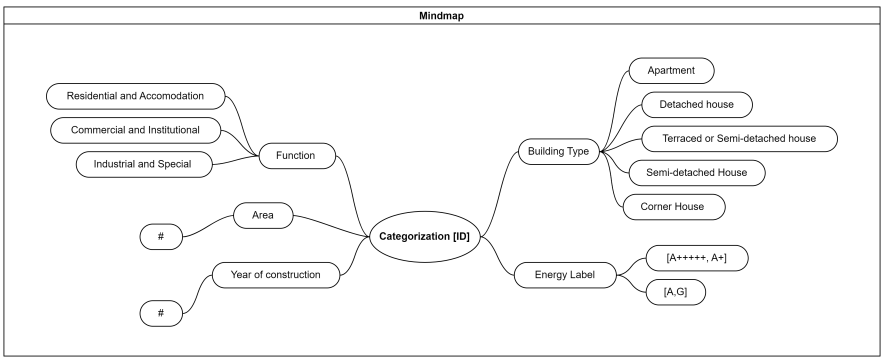


Figure 4.2: Simulation categorization for buildings

4.2.1. Categorization Buildings

Categories are created that delineate clear distinct groups in terms of user profiles, see Appendix B. This lays the foundation for a segmented approach to simulation and analysis. The classifications are instrumental in extrapolating energy consumption profiles across the dataset, while the simulation itself is selectively applied to a subset of the data to establish baseline consumption patterns. This approach allows for the application of the methodology to various scales of practice, regardless of size.

The categorization mind map, as depicted in Figure 4.2, showcases the organizational structure of the dataset into primary groups based on elements such as Function and Building type. These classification differences are made because of the relation to a potentially different energy profile. The Building Type, for example, significantly influences the energy profile of a dwelling. Terraced houses often experience less energy loss—or even gain energy—due to the insulation provided by shared walls with neighboring houses. In contrast, detached houses, which are exposed on all sides, typically suffer from greater energy loss.

The stratification by function, building type, area, year of construction, and energy label allows for a modeling of consumption profiles. This is consequentially extrapolated to the further dataset. This method provides a scalable and efficient means to project energy consumption trends and patterns that can be generalized to larger datasets.

4.2.2. Data Simulation Loads

Data is simulated for several buildings in the case studies. This is done by means of creating categories and simulating a user profile for one building of each of the categories. These profiles can thereafter be extrapolated to the rest of the case study. In this way, always only the amount of categories are simulated. This makes it possible to create reliable simulations for larger datasets without high computation time to simulate every building.

EnergyPlus is utilized for the further simulation. EnergyPlus is a robust building energy simulation program, that models various aspects of energy consumption and water usage within buildings. EnergyPlus's console-based interface can be utilized to read input and generate output in text format. Its accompanying utilities, including IDF-Editor for creating input files, EP-Launch for managing input and output files, and EP-Compare for graphical comparison of simulation results, are utilized. The insights into the energy performance of buildings are aimed to be gained, and used for further extrapolation.

See Appendix B for a complete overview of the simulation.

4.2.3. Data Retrieval: Non-Simulated Exogenous Variables

There are multiple data sources to be retrieved besides the energy consumption. Variables to be retrieved are the weather conditions for solar, wind and geothermal power, see Figure 4.1.

Wind and Solar data

The weather circumstances for the year 2022 are collectible from the data platform of the Waterstaat (2020). The data is accessible via the developer portal. Through the API the information is collected through API calls, using a generated API key. The datasets that are downloaded entail solar, wind, and cloud data. All the sets have an interval of 10 minutes for the entire year 2022, and they provide

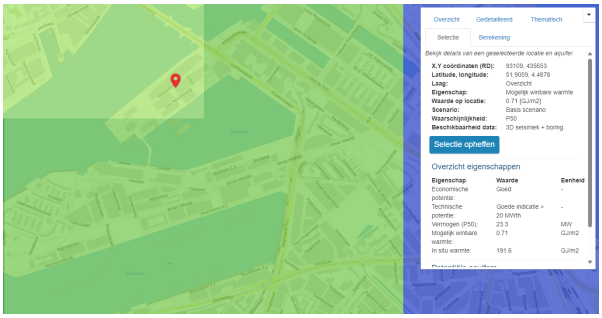


Figure 4.3: Geothermic potential

insight in the potential power generation of the renewables in the system. The datasets contain multiple measurement locations. From these locations 'Rotterdam Geulhaven' and 'Rotterdam Locatie 24t' are in closest proximity to the case studies. See Table 4.1 and Table 4.2 for summarized overviews.

Table 4.1: Wind data collection from Royal Netherlands Meteorological Institute (2024b)

Description	Details
Location	Rotterdam Geulhaven
Coordinates	51.891944, 4.3125
Data Collection Frequency	Every ten minutes
Variable: FF_SENSOR_10	Avg. wind speed from sensors 1 and 2 at sensor height (m/s).

Table 4.2: Solar data collection from Royal Netherlands Meteorological Institute (2024a)

Description	Details
Location	Rotterdam Locatie 24
Coordinates	51.960556, 4.446944
Data Collection Frequency	Every ten minutes
Variable: Q_GLOB_10	Avg. radiation (W per m^2)

Geothermal Potential

The geothermal potential is acquired from TNO (n.d.). The geothermal potential for Kop van Zuid is estimated to be 0.71 GJ/m^2 for shallow depths. This value represents the amount of geothermal energy available per square meter of land surface, which can be harnessed for heating purposes. As illustrated in Figure 4.3, this potential provides a foundation for calculating the possible geothermal heat that can be extracted and utilized.

Grid Import and Export Prices

The market prices for the main grid are derived from a historical database facilitated by Bakker (n.d.), which offers a detailed record of import prices for energy per kilowatt-hour (kWh). To align with the study, the original hourly price data is interpolated to conform to a 15-minute interval. The market prices are acquired from (Bakker, n.d.). They offer a historical database with the import price for energy per kwh. This price is interpolated to adhere to the required 15 minute interval.

4.3. Model Architecture: Variable Environment

All the data in the previous section is part of the fixed environment. The weather is not something that can be influenced by the system, as are the energy market prices, and the geothermal-potential. The variable part of the environment is dependent per scenario per case study. It is made up out of the components and placing in the system, and the training and evaluation setup.

4.3.1. Data Preparation for Variable Environment

Input data for the model is generated according to the scenario setup using QGIS. DERs, including the grid, are generated with the correct parameters and coordinates as a point file. The following table lists the parameters, their corresponding variables, and units:

Parameter	Variable	Unit
Type	Type	-
Efficiency	η, η_{solar}, $\eta_{turbine}$, E_{geo}	%
Max Capacity	C_{max}	kWh
Min Capacity	C_{min}	kWh
Max Charge	P_{charge}	kWh
Max Discharge	$P_{discharge}$	kWh
Battery Cost Cycle	$Cost_{cycle}$	Euro
Max Import	Max_import	kWh
Max Export	Max_export	kWh
CO2 Price	CO2_price	Euro/kWh
Radius	$R_{turbine}$	Meter
Power Coefficient	C_p	-

Table 4.3: Parameters, their corresponding variables, and units.

Several of the DERs are residential, such as PV panels, which are modeled based on roof surface area and availability. For each scenario, it is assumed that the solar potential is utilized to its maximum capacity.

4.3.2. Formalization of Relationships

All elements within the system, along with their relationships and interactions, are precisely formalized. This process involves transforming these relationships into quantifiable and implementable equations suitable for integration within the algorithm.

PV Panels

The energy generation from PV panels can be estimated using solar irradiance over the area with a given efficiency, as described by Equation 4.5.

$$E_{solar} = (A_{PV} \times I \times \eta_{PV} \times \delta t) \times a_t \tag{4.5}$$

Where:

- A_{PV} represents the surface area of the PV panels in square meters (m^2).
- I represents the solar irradiance, the solar power received per unit area in watts per square meter (W/m^2).
- η_{PV} represents the conversion efficiency, the efficiency at which the solar panels convert solar energy into electrical energy.
- δt represents the time period for the calculation, namely 15 minutes.
- a_t represents the availability of the panels per time step.

The availability of suitable surfaces for PV panel installation is set at 0.6, based on a preliminary estimation made using the information provided by Zonatlas (Zonatlas, n.d.). This parameter was derived after an exploratory search on the available geospatial data, see Figure 4.4. Zonatlas offers comprehensive data on roof surface availability, annual solar irradiation, and overall building suitability for PV panel installation, making it a valuable resource for initial assessments.

Buildings are categorized based on their suitability as follows (Zonatlas, n.d.):

- Suitable Buildings: Buildings with a nominal power capacity of at least 1.25 kWp, marked in yellow on Zonatlas.
- Highly Suitable Buildings: Buildings with high solar irradiance, marked in green.

- Unsuitable Buildings: Buildings on which less than 1.25 kWp can be installed or where solar irradiance is very low, marked in red.

While the current 0.6 parameter allows for preliminary calculations and planning, access to more precise and timely availability data via the Zonatlas API would significantly enhance the accuracy of this estimation. The API would provide dynamic, real-time data, allowing for more refined and location-specific assessments of roof suitability and solar potential.

The exploratory search on the available geospatial data already provided a strong foundation for understanding the potential for PV panel installations. However, integrating real-time data from the Zonatlas API would enable a more nuanced and accurate analysis, adjusting for factors such as temporal changes in solar irradiance and updates in building data.

(a) Availability of roofs **(b)** Irradiation of roofs **(c)** Suitability of roofs

Figure 4.4: Case study analysis of roofs for PV panel installation

Wind Turbines

For the urban area, domestic wind turbines can be used, see Figure 2.8. The wind power is calculated with Equation 4.6.

$$P_{wind} = 0.5 \times \rho_{air} \times A_{swept} \times v^3 \times C_p \tag{4.6}$$

Where:

- P_{wind} represents the power output from the wind turbine in watts.
- ρ_{air} represents the air density, typically in kg/m^3.
- A_{swept} represents the rotor swept area of the turbine, which is the area covered by the turbine blades in m^2.
- v represents the wind speed in meters per second (m/s).
- C_p represents the power coefficient, a dimensionless number representing the efficiency of the turbine. This coefficient is influenced by the turbine design and varies with wind speed and blade pitch.

The power coefficient is set to a value of 0.45, reflecting the efficiency with which the wind turbine converts wind energy into electrical energy. Additionally, an air density of 1.225 kg/m^3 is used, representing the standard density of air at sea level under normal conditions.

Geothermic Power & Heat Pumps

The shallow geothermal potential for the Kop van Zuid area is significant, with an estimated potential of 0.71 GJ/m^2, as illustrated in Figure 4.3. According to TNO (n.d.), the region also has the necessary permissions and an existing city heat network, making it suitable for geothermal energy exploitation.

The power generation from geothermal energy is calculated using the following formula, which converts energy to power over a specified area and time period, as shown in Equation 4.7.

$$P = \frac{E}{t} \times A \tag{4.7}$$

Where:

- P represents the power in watts.
- E represents the energy in joules, where 1 GJ = 10^9 J.

- t represents the time in seconds over which the energy is distributed, namely 15 minutes.
- A is the area over which the geothermal energy is harvested or impacts in m^2.

The Coefficient of Performance (COP) is a measure of the efficiency of a heat pump. It is defined as the ratio of useful heating or cooling provided to the work required, see Equation 4.8.

$$COP = \frac{Q_{out}}{W_{in}} \tag{4.8}$$

Where:

- Q_{out} represents the amount of heat delivered for heating in joules.
- W_{in} represents the work input or energy consumed by the heat pump to transfer the heat, in joules.

Battery Charging and Discharging
The charging and discharging for batteries per time step is taken as charge per time step in kWh. The state of charge is updated in defined by Equation 4.9, and Equation 4.10.

$$\text{SOC}_{\text{new}} = \text{SOC}_{\text{old}} + P_{\text{charge}} \tag{4.9}$$

Where:

- SOC_{new} represents the new state of charge, expressed as a percentage, reflecting the current charge level after accounting for the most recent energy transaction (charging or discharging).
- SOC_{old} represents the previous state of charge, also expressed as a percentage, indicating the battery's charge level at the start of the current time step.
- P_{charge} represents the change in energy, measured in kilowatt-hours (kWh). This variable can be positive (indicating charging) or negative (indicating discharging), depending on the energy flow.

$$\text{SOC}_{\text{new}} = \text{SOC}_{\text{old}} + P_{\text{discharge}} \tag{4.10}$$

Where:

- SOC_{new} represents the new state of charge, expressed as a percentage, reflecting the current charge level after accounting for the most recent energy transaction (charging or discharging).
- SOC_{old} represents the previous state of charge, also expressed as a percentage, indicating the battery's charge level at the start of the current time step.
- $P_{\text{discharge}}$ represents the change in energy, measured in kilowatt-hours (kWh). This variable can be positive (indicating charging) or negative (indicating discharging), depending on the energy flow.

For EV batteries, the key difference lies in their availability. These batteries are accessible when plugged in, and upon re-entering the system, their State of Charge (SOC) is randomized

4.4. Implementation Microgrid Environment and DQN Algorithm

The implementation details of the microgrid environment and the DQN algorithm are extensively discussed in Chapter 5. This section presents a generic implementation structure to provide a clear understanding of the methodology from start to finish.

4.4.1. Algorithm and Implementation

RL is identified as the most promising approach for optimizing microgrids that rely solely on renewable energy, as highlighted in Figure 2.11 and supported by Hossain et al. (2023), Ifaei et al. (2023), and Tajjour and Singh Chandel (2023). DRL's effectiveness in managing the complexities of microgrids, particularly in terms of efficiency and adaptability, makes it a suitable candidate for in-depth exploration in microgrid optimization strategies.

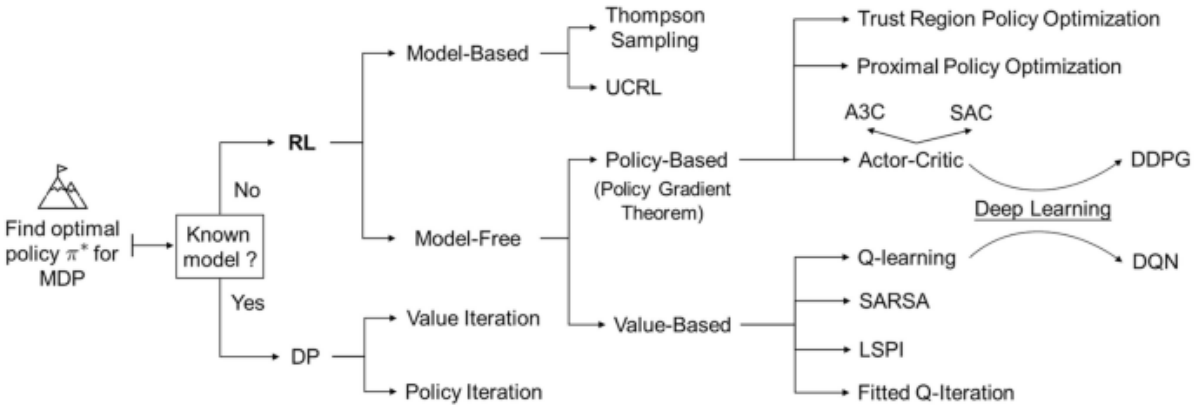


Figure 4.5: Reinforcement learning scheme (Chen et al., 2023)

The agent in this system is governed by a DQN algorithm, which is discussed extensively in Subsection 2.7.2. The DQN improves the stability by adding experience replay and a target network. DQNs are well-suited for environments with discrete action spaces. In this project, time is discretized into 15-minute segments, with each segment representing a distinct state. Actions within these states are limited to a finite set, making the model discrete and suitable for the system.

In response to the research question, and to minimize main grid burden and travel losses, the application of a DRL algorithm combined with a GNN was considered to manage dynamically formed sub-microgrids, which differ in topology. This approach is more described and detailed in the Chapter 5. This thesis focuses on establishing a clear and comprehensive framework using only DQN.

The dynamic clustering is also demonstrated in order to showcase the potential of forming multiple sub-microgrids and how through that interactions can be managed more effectively.

The DQN agent trained in this research aims to optimize the system by aiming to minimize grid connections and costs, addressing the research questions.

4.4.2. Reward Function

The reward function is crucial for training the agent to learn the desired behavior effectively. As discussed in Section 3.5, the reward aims to minimize grid connections, enhance microgrid independence, and reduce operational costs. Consequently, the reward, which aggregates all system costs in euros, is negative. It encompasses the marginal costs for the DERs, import costs, and export costs. The export costs are considered positive for the grid.

To achieve these objectives, the standard microgrid environment is extended by incorporating a custom reward shaping mechanism. This environment utilizes a reward shaping component designed to impose penalties for grid usage, thus encouraging the agent to optimize for self-sufficiency and cost efficiency. The reward shaping mechanism dynamically adjusts the reward based on the grid usage detected in each step. By introducing significant penalties for grid consumption, the agent is steered towards strategies that prioritize local energy production and minimize reliance on external grid resources.

This methodology ensures that the agent's learning process is aligned with the overarching goals of reducing operational costs and enhancing the autonomy of the microgrid system.

4.5. Training and Testing Model

The set up for the training and testing of the model is of importance for the results, as it might influence the manner in which the agent learns, see Subsection 2.7.3.

4.5.1. Scenario Testing

The training is tested by means of the scenarios that have come forward in Chapter 3. The scenarios are designed to maximize rewards by adjusting environmental variables such as demand, supply, and storage capabilities, with a base case scenario serving as a control for performance comparison.

From the scenarios, the impact of changing some parameters and configurations is retrieved. With this information a hybrid scenario is created that aims to represent a configuration of elements for the

case study environment that performs well.

Every scenario results in two graphs: the learning curve for the training and the evaluation. Furthermore, the actions per step are documented for the evaluation as well as the mapping that results from the cluster algorithm.

4.5.2. Data Split

For this created hybrid scenario, the training and reward is optimized by means of testing another data split and extending the training length. Firstly, the seasonal variations are researched by means of running the scenarios with data split in two ways, see Figure 4.6.

1. Split over the year: nine successive training months of data and three successive evaluation months of data.
2. Split over the seasons: four times two and a half successive and a half success evaluation month of data.

The trajectory selection within the data range is done stochastic. For every a trajectory with a first and final step is randomly selected between the extends of the available data (training or evaluation).

The data split leads the training being performed on a specific piece of the data. In the first set up (3:1) the training is done on the months January until October. The evaluation is done on the months October until January. For the other option there is more variety in what is training data and what is evaluation data. The model is evaluated over multiple seasons.

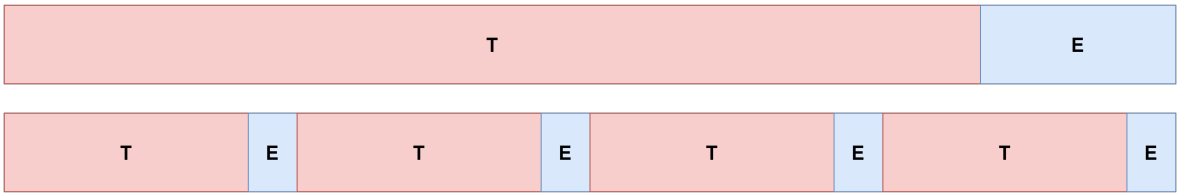


Figure 4.6: Seasonal data split of training data and evaluation data

4.5.3. Training and Evaluation Duration

Furthermore, the impact of the training duration is researched by means of running the most promising scenario for an extensive amount of episodes. The runs performed for the scenarios are performed on a exploratory found number of episodes that shows a stable convergence for scenarios that should converge.

For every scenario the performance is analyzed, by means of validation of the expected behavior and by the convergence and smoothness of the learning curve, see Section 4.6 and Section 4.7.

4.6. Performance of the Model

For the (sub)scenarios a base set of parameters is used. This is determined by means of an iterative search. This base set should be able to converge, but in the end the optimal parameters are searched for the optimal scenario. This is done by means of a grid search with the library Optuna (Akiba et al., 2019). In evaluating the effectiveness of the training process for the microgrid environment machine learning model, two critical features are used to determine the quality of the training: smoothness, and convergence.

4.6.1. Convergence

Convergence is a key indicator that the model has successfully learned the underlying patterns and relationships in the data. It is typically observed when the learning curve levels off or plateaus, indicating that additional training is no longer resulting in significant improvements in performance. This plateau suggests that the model is making accurate predictions and has achieved a stable state. To ensure the model reaches an optimal state, a standard number of episodes are used during training. If convergence is observed within this standard amount of episodes, it indicates that the training process is adequate. For further analysis the amount of episodes can be increased.

4.6.2. Smoothness or Variability

The smoothness of the training curve is an indicator of the stability of the learning process. A smooth curve suggests that the model is learning in a consistent manner, without drastic fluctuations in performance. It can be measured using the variability, measuring the dispersion of the rewards from the mean. To analyze the variability of the rewards during training and evaluation, the mean, variance, and standard deviation of the rewards is calculated. These statistical measures provide insights into the stability and performance of the model. See Equation 4.11, Equation 4.12, and Equation 4.13.

$$\mu = \frac{1}{N} \sum_{i=1}^{N} x_i \tag{4.11}$$

Where:

- μ represents the mean reward.
- N represents the total number of episodes.
- x_i represents the reward for episode i.

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^{N} (x_i - \mu)^2 \tag{4.12}$$

Where:

- σ^2 represents the variance of the rewards.
- μ represents the mean reward.
- N represents the total number of episodes.
- x_i represents the reward for episode i.

$$\sigma = \sqrt{\sigma^2} \tag{4.13}$$

Where:

- σ represents the standard deviation of the rewards.
- σ^2 represents the variance of the rewards.

4.7. Validation

Validation is conducted by examining the results and determining if they align with expectations. Validating RL models can be complex, as discussed in Subsection 2.7.3. In validating the performance of the RL model, multiple critical metrics and methodologies are employed to ensure robustness and effectiveness. Firstly, cumulative rewards are quantified to evaluate the model's proficiency in achieving its designated tasks over time. Secondly, the convergence rate is analyzed to determine the speed and efficiency with which the model approaches near-optimal policies.

Additionally, the model's ability to generalize is tested by introducing data not covered during training, assessing its adaptability and robustness. These are the evaluation iterations. A cyclic training and evaluation strategy is also implemented, enabling the model to continuously adapt to seasonal and dynamic environmental changes. This cyclic approach, along with iterative feedback loops, allows for ongoing adjustments and improvements, ensuring that the model remains effective and relevant under varying conditions.

4.8. Visualization Results

The creation of the visualization involves mapping the physical grid's attributes and behaviors into the digital domain, ensuring that the virtual model is as close a representation of the real system as possible in the time span of the project. The purpose of this visualization is to be both insightful and functional, serving as a reliable tool for strategy development and collaboration. With this the last research question is answered:
Q3How can the location-based optimization of a microgrid be effectively visualized to ensure it creates value for all stakeholders?

4.8.1. Platform Selection

In order for the visualization to create value for the stakeholders, the platform must be accessible and integrated into the environment. The stakeholders include the GIS group of Accenture and its direct clients. Following a cooperative meeting with an expert DT employee from the GIS group, potential tools for use were mapped. After analyzing current projects and searching for similarities and benefits between these projects and the proposed thesis approaches, a collaborative approach is established, see Table 4.4. A currently used DT for real-time water flow management shows significant similarities in performance and aims, and thus, this model is used as a general principle. The information is transformed and hosted in ArcGIS. Furthermore, a front end similar to ArcGIS Experience builder is used. The DT for this thesis is built in a similar way. The transformation, and preparation of all the acquired information is done in ArcGIS. Thereafter, the results are presented utilizing ArcGIS Experience builder. However, it is important that all the information must remain shareable, and for that reason the web map is maintained in AGOL. In this way the information remains fully accessible after the end of this research.

Table 4.4: Digital mapping methods

Phase	Tools and Technologies	Description
Initial Implementation	Map layer in ArcGIS and presentation and hosting in AGOL and ArcGIS Experience Builder	Utilizes ArcGIS for the creation of the layers that can be maintained by the stakeholder, and ArcGIS Experience Builder for spatial data visualization and interaction. The map layer will contain a pre loaded scenario in order to reduce the waiting time for the user and to show the potential.
Evolution	Customized Development	Depending on the specific needs and feedback, the DT may evolve into a more customized solution, like the currently used DT for water flow management. This includes making it possible to change the parameters, run the program, and automatically loading the generated information from the database into the existing layer. Running of the script can in that case, for example, be done with multiple GPU's to speed up the process.

4.8.2. Visualization Design

The information that is acquired from the previous research steps is pre processed in order to be transformed into a temporal format in ArcGIS. The goal of the visualization is to present the case study and influential factors in a web map, and to host this in AGOL. The goal of this is to promote reuse of the concept after this research.

The pre-processing of the data entails transforming the temporal action data, that comes forward from training the agent and evaluating it on unseen data, and showing the changing state over time. Furthermore, the map must show influential factors, like the presence of batteries for specific buildings.

Arcgis Experience Builder offers an intuitive GUI. The DT can be constructed from several modular components, like a map, a legend, but also as advanced as oriented imagery. The DT is created by means of a base map, and stacked map layers. A point layer contains the fixed information of the environment. An additional shape layer, from the original (pre-processed) information is linked. This shape layer functions as the framework for the action information. Dependent of the state of a controllable module individually it is colored. Also, demand per building is colored in a graduate scale. With a temporal slider it is possible for the user to see what the actions are per time step. This may also slide automatically

with a set speed.

4.8.3. Feedback Expert Panel

The proof of concept (POC) is tested by means of a feedback expert panel. The panel exists of a selection of three employees that work with both the clients and have experience in DT employment. During the session the POC is presented to the participants of the panel. Feedback is gathered by means of continuing a structured meeting that assesses various aspects of the DT's functionality, usability, and overall effectiveness. Questions during the meeting aim to elicit the user experience and the technical performance of the system. The feedback session employs interactive discussions, allowing the panel members to express their thoughts and suggestions in real-time as they interact with the DT. This approach ensures that the developers receive immediate and actionable insights, which can be crucial for iterative improvements.

The gathered feedback is analyzed, and common themes and areas requiring enhancement are identified. the forthcoming themes are thereafter prioritized and divided into immediate actions and future work

Immediate actions focus on quick wins and essential fixes that can be implemented swiftly to enhance the performance and usability of the DT. These typically include adjustments to the interface, resolving any bugs identified during the session, and improving data integration processes to ensure smoother operation. The goal is to make the DT more intuitive and efficient for the users, thereby increasing adoption rates and satisfaction.

Future work involves more strategic and comprehensive improvements that require careful planning and possibly significant development resources. This category may include scaling the application, enhancing security features, integrating advanced analytics capabilities, and expanding the DT's functionality to cover additional aspects of the client's operations. It also involves monitoring the impact of implemented changes and continuously gathering user feedback to guide further development.

5

Model Implementation

This chapter describes the technical setup and the implementation details of the methodology of the project. It provides insights into the programming environment, tools used, and the iterative development approach that was employed, highlighting the adaptability and continuous improvement of the project code. The implemented code is provided in Appendix E. The GitHub link and the most important subscripts are presented. The code base is structured in a modular fashion, with various subscripts handling defined functionalities integral to the project's operations.

5.1. Libraries

In order to implement the microgrid, and make it a RL environment with a trained agent several libraries are used. These libraries are explained below, alongside with the choices that are made during the implementation.

Pymgrid

Pymgrid is a Python library specifically designed for microgrid generation and simulation (Henri et al., 2020). It provides a set of tools that enable the rapid deployment and testing of various microgrid configurations. The library includes predefined classes of microgrid modules. This includes Grid, Load, Renewable, Battery, Genset, and UnbalancedEnergy modules. Furthermore, forecasting methods, RL environment wrapping with Gymm and some control algorithms are included. Reinforcement Learning is not built in, but the wrapping makes it suitable to do so.

In Figure 5.1, an overview is offered of the use of pymgrid to from the microgrid environment by means of a UML.

The `RL_microgrid_environment.py` module sets up the overall environment and interactions between various components of the microgrid system. It adjusts load profiles with `adjust_load_profile()`, computes wind power with `calculate_wind_power()`, and initializes microgrid components with `create_microgrid()`. The `LoadModule` manages load profiles, the `BatteryModule` handles energy storage, and the `RenewableModule` simulates renewable energy sources. Both the `LoadModule` and `RenewableModule` are time series modules, reflecting data that changes over time. The `GridModule` manages power exchange with the external grid, also as a time series module. The `Microgrid` integrates these modules to simulate a complete microgrid system.

Gym

Gym, developed by OpenAI, is an open-source toolkit widely used for developing and comparing DRL algorithms (Brockman et al., 2016). In the context of microgrid simulations, Gym offers a standardized way to define and manage environments, which represent different states or scenarios in microgrid operation. The Gym environment for microgrid simulation can f.e. be structured to represent several days of energy management, with each "episode" simulating a single day. This setup allows for the assessment of various energy strategies across different operational scenarios.

`FixedLengthStochasticTrajectory(nr_steps)` is used for the selection of the trajectories. Within the available dataset for the training, every begin step for the next episode is selected stochastic. This

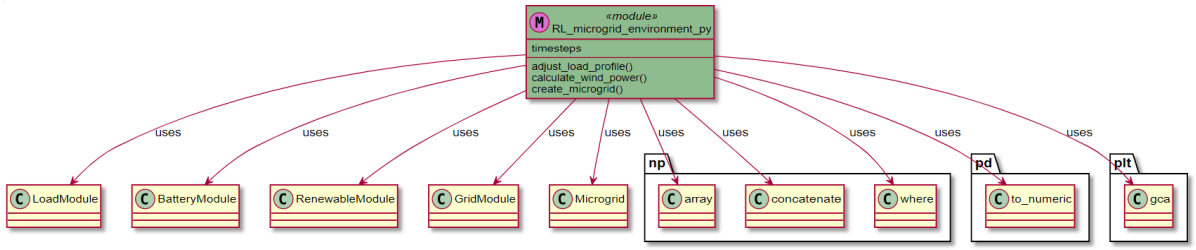


Figure 5.1: UML diagram microgrid environment

selection takes in the trajectory length and thus sets the begin step somewhere between step 0 and step `train\_dataset) - trajectory length`. The trajectory length can be taken as a parameter.

Keras

Keras is used for the implementation of the DRL algorithm. The following imports are used for the DQN agent, training and evaluation functions:

```
...
from collections import deque
from keras.models import Sequential
from keras.layers import Dense
from keras.optimizers import Adam
from keras.layers import Input
...
```

Listing 5.1: Import of Keras

- **Deque from collections**: A deque (double-ended queue) is used to store and manipulate the agent's experiences, enabling efficient memory replay that is essential for learning from past actions.
- **Sequential from keras.models**: This module allows for the linear stacking of layers within the neural network, simplifying the model construction process.
- **Dense from keras.layers**: Dense layers are fully connected layers, commonly used in deep learning networks to connect neurons with activation functions.
- **Adam from keras.optimizers**: The Adam optimizer is employed for its efficient gradient descent optimization, which is faster and more effective than traditional stochastic gradient descent in many cases.
- **Input from keras.layers**: Used to instantiate a Keras tensor, serving as the input layer of the neural network.

Other

More libraries and dependencies are used. These are all provided in the `requirements.txt` file included in the repository, see Appendix E.

5.2. DQN and Reward Function

Pymgrid offers 25 standard microgrids, which can be used to validate the behavior of the DQN implementation with the reward function. From this framework on the further extensions can be build utilizing the simulated data and thus the required amount of components and supply. The implementation of the DQN agent and helper functions are tested on 2 of the 25 pymgrid microgrid environments: 10 and 21. See Table F.1 for the setup of those standard environments.

The reward function is implemented within the `DiscreteMicrogridEnv` class, a custom environment that encapsulates the configured microgrid with all its components, parameters, and potential actions within a gym framework. This class is extended using a super class to enhance functionality, as illustrated in the psuedo code of the step function and the reward shaping mechanism presented in Algorithm 5.1 and Algorithm 5.2.

Algorithm 5.1: Constructor for CustomMicrogridEnv

Input: args, kwargs
Output: CustomMicrogridEnv instance

```
1 begin
2     Call superclass constructor with args and kwargs;
3     Initialize reward_shaper with GridUsagePenaltyShaper instance;
```

Algorithm 5.2: STEP

Input: action
Output: observation, reward, done, info

```
1 begin
2     Call superclass step method with action and store the result in observation, reward, done,
        info;
3     Adjust reward by subtracting reward_shaper(step_info = info) from reward;
4     return observation, reward, done, info;
```

To firmly discourage grid connections, an additional penalty is introduced in the subclass `RL_custom_Env.py`. This subclass adjusts the reward based on grid usage, as detailed in Algorithm 5.2. The `RL\_custom\_reward.py` script encapsulates the logic to apply penalties based on grid usage, as described in Algorithm 5.3.

Algorithm 5.3: GridUsagePenaltyShaper

Input: step_info
Output: penalty

```
1 begin
2     grid_usage ← sum_module_val(step_info, 'grid', 'provided_energy');
3     if grid_usage > 0 then
4         return 50000000 // Halve the reward if the grid was used
5     return 0 // Otherwise, return the full reward
```

In practice, the initiation of the microgrid within the $CustomMicrogridEnv()$ not only calculates the reward but also applies an extra penalty if there is a grid connection during a step. This significant penalty ensures that the agent's strategy prioritizes minimizing grid dependency, encouraging exploration of alternatives. Import and export costs also dynamically influence this penalty, promoting sustainable practices during economically advantageous times, such as negative pricing periods when feeding the grid is beneficial. While the penalty for grid connection is significant to discourage reliance on the central grid, there are occasions when the financial rewards from negative pricing or exporting energy during advantageous times may outweigh this penalty.

The trajectory selection is explained in Algorithm 5.4.

From the training of the agent, with the parameters defined in Table F.1, the input data is put into a graph together in Figure 5.2. The training curves and evaluation curves acquired by the training of the agent in these environments, **with the new reward shaping function penalizing grid connection and costs,** are presented in Figure 5.3.

Algorithm 5.4: FixedLengthStochasticTrajectory

Input: trajectory_length, initial_step, final_step
Output: (initial, initial + trajectory_length)

if *final_step - initial_step < trajectory_length* **then**
    raise ValueError
initial ← random integer in [initial_step, final_step - trajectory_length]
return *(initial, initial + trajectory_length)*

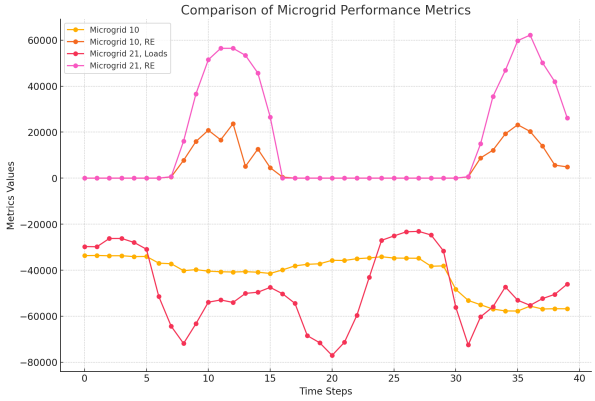
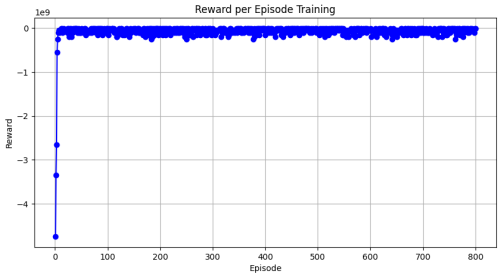
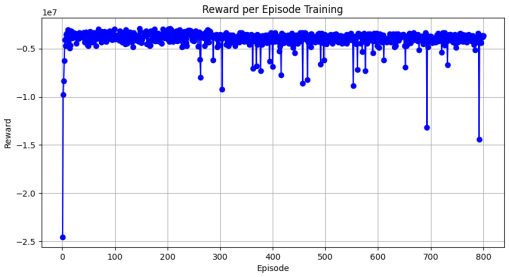


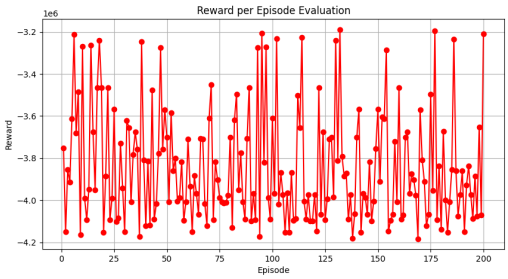
Figure 5.2: Input data Pymgrid microgrid number 10 and 21



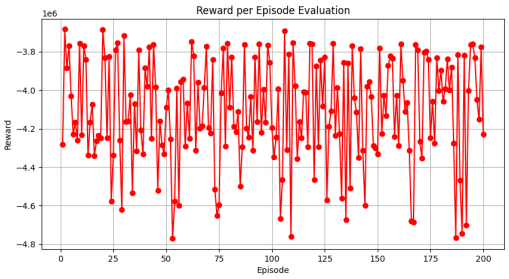
(a) Training rewards for Microgrid 10



(b) Training rewards for Microgrid 21



(c) Evaluation rewards for Microgrid 10



(d) Evaluation rewards for Microgrid 21

Figure 5.3: Comparison of training and evaluation rewards for Microgrids 10 and 21

The agent shows expected training curves for both environments, meaning that the DQN algorithm functions as is expected. The evaluation curves show that the agent has been able to immediately proceed with the trained behavior, while it has not seen the data yet. This confirms the correct training.

Tuning to find a base set of parameters is done on MG10.

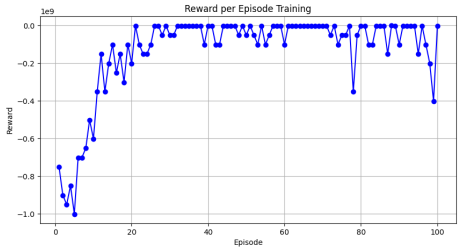
5.3. Evaluation Curves

The evaluation curves and values are analyzed by examining their standard deviation and overall shape. The evaluation curves presented in Figure 5.3 demonstrate the expected behavior. These curves serve to verify the training of the agent, ensuring there is no occurrence of over or under fitting, meaning that the complexity of the system is not captured in the right way.

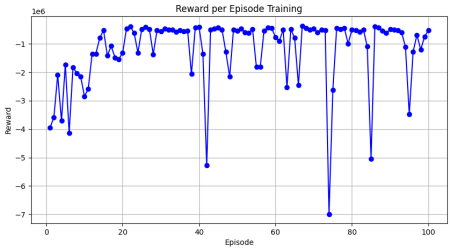
The evaluation curves are crucial for understanding the performance and stability of the agent. The absence of significant deviations and the alignment with expected reward trends indicate robust training.

5.4. Hyper Parameter Tuning

A second run on MG10 with difference parameters reveals the influence of the parameters. In Figure 5.4, the training curves for identification of daily patterns are presented, with an episode length of 24 steps of one hour.



(a) Training curve for MG10 with episodes of 24 hours



(b) Training curve for MG21 with episodes of 24 hours

Figure 5.4: Training curves for MG10 and MG21 with episodes of 24 hours

To evaluate the influence of various parameters on the model's performance, an incremental approach is adopted, adjusting one parameter at a time while monitoring the rate of convergence and the reward function. Although this method is somewhat preliminary, it serves as a foundational step to establish baseline parameters. Once these baseline parameters are identified, further fine-tuning is conducted by testing combinations of different parameter levels. This refined tuning is applied within the context of the optimal scenario determined for the case study, ensuring that the most effective parameter configurations are identified for enhanced performance.

Table 5.1: Summary of training and evaluation runs for MG 10 with episode length of 168 steps

Learning Rate	Mem_Size	Num_Layers	Layer_Size	Eps_Decay	Gamma	Avg. Reward
0.0001	4*steps	4	64	0.995	1	-3.8650E+06
0.00001	4*steps	4	64	0.995	1	-3.8650E+06
0.0001	4*steps	4	64	0.995	0.95	-3.8649E+06
0.0001	4*steps	4	64	0.999	1	-3.8649E+06
0.001	4*steps	4	64	0.995	1	-3.8649E+06
0.0001	8*steps	4	64	0.995	1	-3.8650E+06
0.0001	4*steps	4	128	0.995	1	-3.8650E+06
0.0001	4*steps	8	64	0.995	1	-3.8650E+06
0.001	4*steps	4	64	0.999	0.95	-3.8650E+06
0.001	4*steps	4	64	0.999	1	-3.8650E+06
0.0001	**4*steps**	**4**	**64**	**0.999**	**1**	**-3.8649E+06**

For the initial setup, a base set of parameters that balance performance stability and robustness is chosen from multiple runs, see Table 5.1. The learning rate is set to 0.0001, which provides better performance in reducing the negative average reward compared to lower rates. The memory size is set to 4, a common setting that does not show significant improvement when increased to 8. The configuration includes 4 layers, a common setup across different tests, and a layer size of 64, as increasing the size to 128 did not significantly enhance performance. Epsilon decay is set at 0.999, which stabilizes the

Figure 5.5: Training curve for MG 10 with base set parameters:

learning process better than the more commonly used 0.995. Lastly, a gamma value of 1 is selected, as it is associated with scenarios yielding less negative rewards, suggesting better performance stability. These parameters are anticipated to provide a preliminary adequate foundation for further model development and optimization in the case study.

5.5. Computation: Azure Virtual Machine

Because of the long computation time for some of the scenarios, a connection is made to an Azure Virtual Machine (VM) with more computation power. The VM that is used is the Standard D8s v3 with 2 vCPUs. The VM is started and thereafter, a connection is made via Remote Desktop Connection. From there the code for the hyper parameter optimization and the DRL optimization is run via the Powershell. In order to do this, the code is adjusted and extended with flags. All the parameters, see Table 6.17, and all the paths are passed via the command line.

5.6. Case Study Environment

To acquire robust results for the case study, the DQN agent must be adequately trained in a carefully crafted environment. This environment is constructed using real input data, which is simulated with EnergyPlus among other advanced modeling tools, to mirror realistic scenarios and system dynamics closely. The simulation environment is tailored to incorporate an accurate representation of system components, ensuring each component's functionality is precisely defined.

The configuration of the system components includes specifying the correct type and number of components such as generators, batteries, renewable energy sources, and load profiles. Each component's operational parameters are defined based on realistic data to ensure that the agent encounters scenarios that are as close to real-world conditions as possible.

5.6.1. Scenarios

The scenarios for the case study, conceptually defined in Table 3.2, are further developed and shaped incrementally, reflecting a systematic approach to exploring and enhancing the system's performance. Starting from a base case scenario, which serves as *the benchmark* for the system's current capabilities, each subsequent scenario introduces modifications aimed at testing specific aspects of the system. These modifications focus an adjusting demand patterns, integrating different scales of renewable energy inputs, or implementing varying levels of storage solutions. This incremental and exploratory approach ensures that each scenario builds on the insights gained from the previous ones, leading to a comprehensive understanding of potential improvements and the most influential factors affecting the system's performance. By methodically adjusting and observing the effects of these changes, the study aims to uncover nuanced strategies that can significantly enhance the efficiency and sustainability of energy management systems.

5.6.2. Aim of Optimization

The aim of the optimization is to illustrate the changing behavior across different scenarios and to guide the development of a hybrid scenario that combines the modules in a feasible manner, showcasing a potential future energy system. The scenarios are therefore analyzed after each subsection, and this guides the design and implementation of a hybrid model. By comparing the outcomes of each scenario,

the study identifies the most promising approaches for enhancing the energy system. Key findings are highlighted, such as the impact of increased renewable energy integration on system stability and the benefits of enhanced storage solutions.

6

Results Optimization

The output of the case study is presented in this chapter. The optimization algorithm is run for multiple settings and for various flexible scenarios. First, an overview of the input data is provided, describing the fixed environment. Thereafter, the parameters are defined for each scenario. For every scenario, the learning curve and the evaluation graph are presented. These are analyzed, and these analyses together create a combined scenario. The final combined scenario is then optimized and enhanced further.

6.1. Fixed Environment

The loads, or demand per building, are shown in Figure 6.1. It is clear that the simulation data requires intitialization time. For this reason, the first week is removed from the research in the further research. Thereafter, the generation in the system is showcased for scenario 1, see Figure 6.1d. Comparing the two profiles shows the mismatch in the generation and the load profile. Also it shows the difference in the magnitudes of the two components. For the first scenario the agent is trained in this environment. Expecting to not be able to fully suffice to the demand. The reward function aims to make the system match this as well as possible, by penalizing the fall back to the main grid.

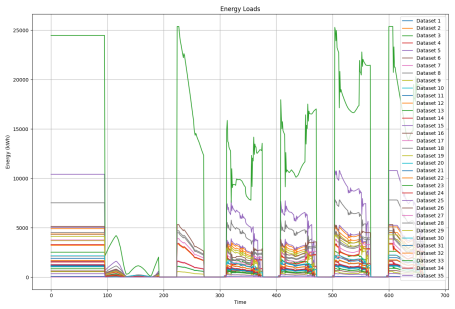
6.2. Variable Environment

For each scenario, specific parameters are defined and executed with the base settings established through iterative testing on the standard Pymgrid environment, as detailed in Chapter 5. These base settings, which have proven to yield steady results, are presented in Table 6.1.

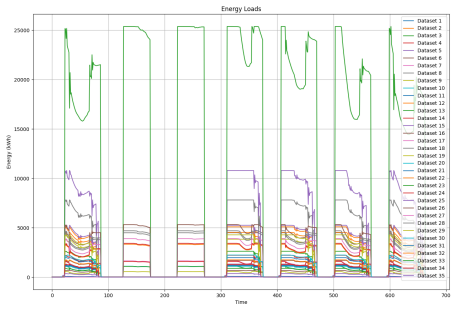
The standard environment has shown that the agent can be effectively trained within 100 episodes, achieving convergence. Therefore, all initial exploratory research across various scenarios is conducted using this 100-episode framework, supplemented by 40 evaluation episodes. To further validate the results and assess long-term performance, the most promising sub-scenarios are retrained for a higher number of episodes to ensure sustained reward levels.

Both daily and weekly patterns were recognized within the standard environment. For the case study, these patterns are tested to determine which is the most favorable and identifiable. The selected pattern is then utilized for further analysis, ensuring the robustness of the approach.

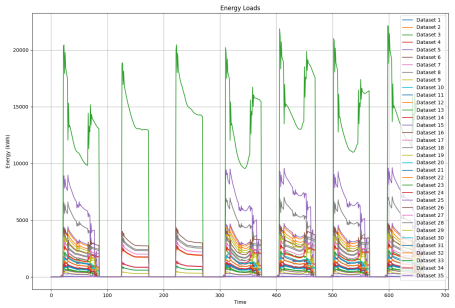
For the most favorable scenario overall, Optuna hyperparameter tuning is implemented, focusing on critical variables such as gamma, learning rate, and epsilon decay. These parameters are selected due to their significant impact on rewards within the standardized training environment, as summarized in Table 5.1.



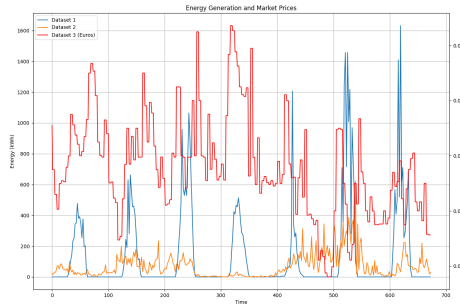
(a) Loads week 1



(b) Loads week 2



(c) Loads week 3



(d) Generation and market prices week 3

Figure 6.1: Energy loads and generation market prices during warm-up time

Table 6.1: Base settings

Setting	Value	Description
Episode length	672 or 96	Duration of one complete episode
Training iterations	100	Number of total training cycles
Evaluation iterations	40	Number of cycles for testing the model
Memory size	4 episodes	Total steps stored for replay
Neural Network	48 by 48	Size of each layer in the neural network
Gamma	0.95	Discount factor for future rewards
Epsilon	1.0	Initial value for exploration
Epsilon decay	0.999	Rate of reduction for epsilon over time
Epsilon min	0.01	Minimum value for exploration
Learning rate	0.0001	Step size for weight updates
Batch size	64	Number of training examples used per iteration
Data split	3:1	Division of data for training and testing
Episode selection	Stochastic	Method for selecting episodes during training
Loss function	Mean Squared Error	Metric for evaluating model predictions

6.2.1. Scenario 1: Base case

The first scenario, as described in Section 3.4, envisions a future with a steady increase in renewable energy sources, such as solar and wind, integrated into the environment. This scenario involves several key components, further explained below.

Every building is equipped with solar panels to the extent that space allows. Additionally, a suitable number of wind turbines are strategically placed based on spatial constraints, determined through geo-spatial exploratory research using ArcGIS. For the micro wind turbines, a free radius of 27 meters is required to ensure optimal performance and spacing. The placement of domestic wind turbines, ensuring a distance of three times the radius between two turbines, is shown in Figure 6.2.

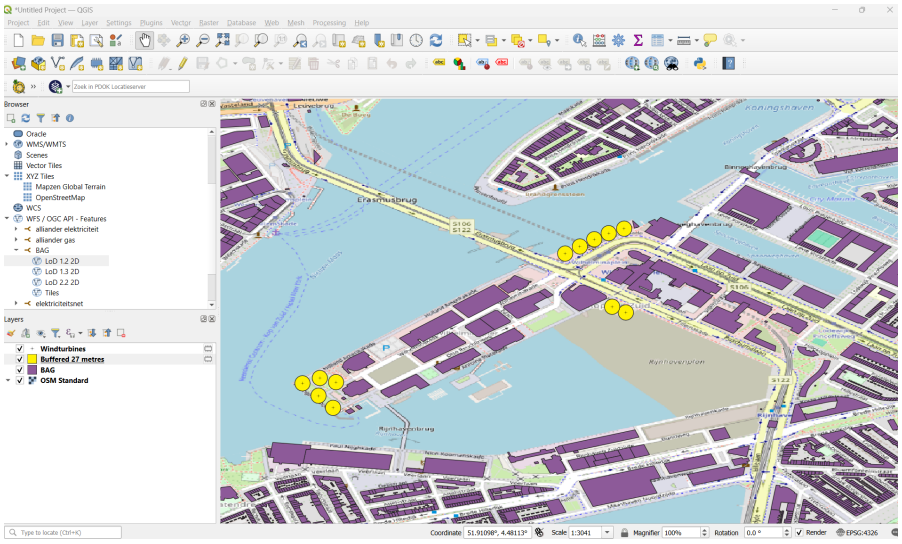


Figure 6.2: Placement of domestic wind turbines, resulting in 12 turbines with a radius of 9 meters

Various energy storage technologies with currently feasible capacities, as outlined in Table 2.2, are incorporated to enhance flexibility in managing intermittent renewable resources and to minimize dependency on the main grid. Buildings with 20 to 100 residential units are assumed to have a small battery, while those with more than 100 residential units are equipped with a large battery. Battery placement is determined through a geo-processing step that generates randomized points within suitable buildings, represented as a point file.

Each building is equipped with a heat pump to provide a more constant energy output, complementing the intermittent energy from solar and wind sources. The heat pumps are assumed to have uniform characteristics, with the area of each building used to calculate the energy output that can be generated from the district's geothermal potential, as detailed in Chapter 4. The efficiency of these heat pumps is enhanced by their COP, set at 4.5, indicating that the energy output is 4.5 times greater than the energy input.

Distances are not considered in this initial setup, to simplify the model, see Section 3.7. Components with identical trajectories are aggregated in the code to reduce the action space as much as possible. This approach reduces the complexity of the action space. With 6 fixed source modules, already a action space of 1440 elements is defined. By aggregating elements with similar characteristics this is reduced. This simplification can be refined when the GNN is integrated with the current algorithm, see Section 3.6.

Table 6.2: Variables, parameters, and values DRL model

Variable	Parameter	Value
Small Batteries		
C_{max}	Capacity	500 kWh
P_{charge}	Charge Power	38 kWh
$P_{discharge}$	Discharge Power	38 kWh
η_{ch}	Charge Efficiency	0.9 (dimensionless)
$Cost_{cycle}$	Cycle Cost	$0.20
$Quantity$	Number of Units	6
Big Batteries		
C_{max}	Capacity	700 kWh
P_{charge}	Charge Power	50 kWh
$P_{discharge}$	Discharge Power	50 kWh

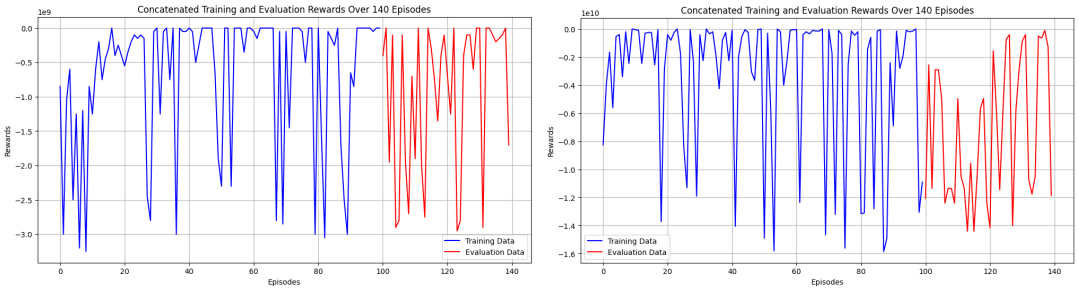
Continued on next page

Table 6.2 – continued from previous page

Variable	Parameter	Value
η_{ch}	Charge Efficiency	0.9 (dimensionless)
$Cost_{cycle}$	Cycle Cost	$0.50
$Quantity$	Number of Units	11
Solar panels		
η_{PV}	Efficiency	0.9 (dimensionless)
$Area_{available}$	Available Area	Solar analysis with Energy atlas, now 0.6 percent
$Quantity$	Coverage	Every available surface
Domestic-scale wind turbines		
$R_{turbine}$	Radius	9 m
C_p	Power Coefficient	0.45 (dimensionless)
ρ_{air}	Air Density	1.225 kg/m^3
$Required\_radius$	Free Radius	27 m
$Quantity$	Number of Units	12
Heat Pumps		
P_{input}	Electrical Input	2 kW
E_{geo}	Potential	0.71
COP	Coefficient of Performance	4.5 (dimensionless)
P_{output}	Heat Output	7 kW
$Cost_{operation}$	Operational Cost	$0.50
$Quantity$	Number of Units	One per residential object

The training curves are presented in Figure 6.3. Because there is no clear convergence the evaluation curves are also analyzed. The learning curves provide several novel insights, see Figure 6.3 . Firstly, it has to be noted that the reward does not seems to converge completely under the current scenario to an optimal value. This can be attributed to several factors, and after extensive testing with standardized environments in Chapter 5 it has been ruled out that the experimental design, the reward function, and the network architecture play a role in this. The lack of convergence and the high variety is explained by the intermittent nature of the input data and the mismatch in quantity, and it is likely that there is no optimum way of performing. This can be declared by the fact that, even though the agent is trained, it must sometimes take the penalty in order to meet the demand of the microgrid.

To compare day and week patterns, the base case is also trained with 672 steps, 15 minute interval for one week, see Figure 6.3b. It can be seen that there is no convergence from the training curve. When concatenating the evaluation curve to the training curve it becomes clear that the learning is done for the daily pattern though and not for the weekly scenario. From now on only daily patterns are therefore trained and tested.



(a) Scenario 1.1: Episode length of a day

(b) Scenario 1.2: Episode length of a week

Figure 6.3: Scenario 1: Concatenated training and evaluation curves

Table 6.3: Summarized version of: Evaluation rewards for sub-scenarios within scenario 1, see Table F.2

Episode	Scenario 1.1	Scenario 1.2
1	-401440312.5	-12060352867
..	..	..
40	-1701428874	-11860354842
Total	-9.31E+08	-7.37E+09

Analysis Scenario 1

In Figure 6.3 the training rewards (blue curve) and evaluation rewards (red curve) over 140 episodes are displayed. For scenario 1.1 the training rewards show high variability with significant drops, indicating instability during the learning process. Despite this, the training rewards generally improve over time, suggesting that the model is learning but with fluctuations. The evaluation rewards, which begin around episode 100, also show variability but tend to stabilize towards the latter episodes.

The training rewards exhibit a high degree of fluctuation, which may suggest over fitting or sensitivity to specific training data points. However, the evaluation rewards, while initially unstable, show signs of improvement and stabilization in the later episodes, indicating potential generalization of the model.

In scenario 1.2 the model is trained on episodes spanning a week, with the corresponding training and evaluation rewards shown in Figure 6.3b. The training rewards exhibit a similar pattern of high variability, but the magnitude of fluctuations appears somewhat reduced compared to Scenario 1.1. The evaluation rewards again start around episode 100 and show substantial variability.

The training rewards still demonstrate fluctuations, but there are more consistent periods of higher rewards compared to Scenario 1.1. However, the evaluation rewards fluctuate significantly, indicating that the model might be struggling to generalize effectively when trained on weekly episodes.

A comparison of the total rewards for each sub-scenario, as summarized in Table 6.3, indicates that Scenario 1.1 (daily episodes) results in a lower cumulative negative reward compared to Scenario 1.2. This suggests that training on daily episodes may be more effective for this particular model and dataset.

Training on daily episodes seems to yield a more stable learning curve, with less severe fluctuations in rewards. The total negative reward is significantly lower, indicating better overall performance. Training on weekly episodes, on the other side, results in higher variability and greater cumulative negative rewards, suggesting that the model may struggle with the longer episode duration, potentially due to more complex patterns that are harder to capture over longer periods.

6.2.2. Scenario 2: Demand Side Decrease

To determine the required reduction in demand for the agent and the algorithm to converge, the loads are decreased. The initial parameters are adopted from Chapter 5. The second scenario evaluates the necessary decrease in demand-side management for the agent to effectively control the environment and achieve optimal performance. The loads are iteratively reduced using both large and small steps to identify the critical turning point, see the concatenated training and evaluation curves in Figure 6.4.

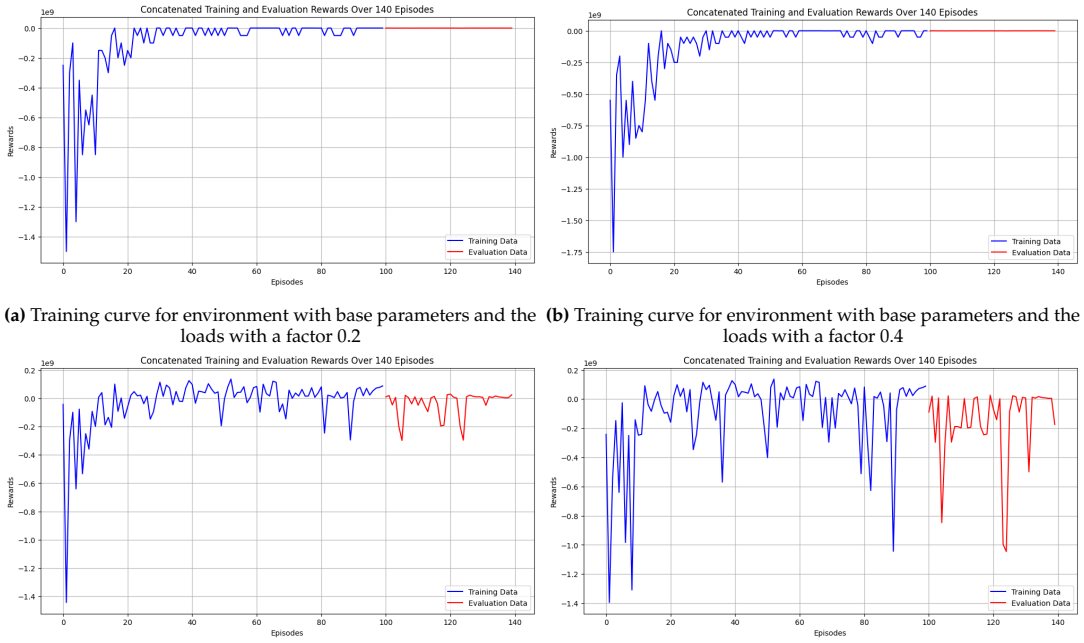
Table 6.4: Summarized version of: Evaluation rewards for sub-scenarios within scenario 2, see Table F.3

Episode	Base case	Scenario 2.1	Scenario 2.2	Scenario 2.3	Scenario 2.4
1	-401440312.5	-263850.225	-562632.52	11243708.78	-88906300
..	..	..	..	..	..
40	-1701428874	-287048.409	-606910.991	25922762.41	-174252802.5
Average	-9.31E+08	-2.46E+05	-5.25E+05	-3.47E+07	-1.59E+08

The evaluation rewards are shown in Table 6.4. A decrease of the loads with a factor 0.6 shows a stable convergence, while with a factor 0.8 also shows potential.

Analysis Scenario 2

The four different scenarios, where the demand loads are decreased by different factors, are analyzed. For all the sub-scenarios for scenario 2 the training rewards (blue curve) and evaluation rewards (red



(a) Training curve for environment with base parameters and the loads with a factor 0.2

(b) Training curve for environment with base parameters and the loads with a factor 0.4

(c) Training curve for environment with base parameters and the loads with a factor 0.6

(d) Training curve for environment with base parameters and the loads with a factor 0.8

Figure 6.4: Scenario 2: Concatenated training and evaluation curves

curve) over 140 episodes are displayed, see Figure 6.4. All scenarios are trained over the same number of episodes, but the demand loads vary in order to analyze the convergence and learning ability of the model with overload of energy. The training and evaluation rewards for each scenario are analyzed in order to understand how varying demand loads impact the learning and generalization of the model.

The training rewards for all the sub-scenarios exhibit high variability, with a general upward trend.It is evident that decreasing the loads enables the agent to learn a more optimal way of acting, thereby increasing the average rewards, see Table 6.4 and Figure 6.4.

The scenarios show variability with substantial drops, potentially indicating instability during the learning process. Despite this, the training rewards generally improve over time, suggesting that the model is learning but with fluctuations. The evaluation rewards, which begin after 100 epsiodes of training but on unseen data, also show variability but tend to stabilize in the latter episodes, demonstrating that the agent has learned and is able to consistently achieve higher rewards.

Although initially unstable, the evaluation rewards show signs of improvement and stabilization in the later episodes, indicating potential generalization of the model. It becomes clear that all scenarios generalize effectively, but as expected, the agent can maximize the reward more effectively when the loads are lower.

Training with lower demand loads seems to yield better overall performance, while higher demand loads introduce more complexity, leading to higher variability and greater cumulative negative rewards. The learning rate may need to be optimized further to potentially reduce variability and enhance overall performance.

In summary, the analysis reveals that the agent's learning and generalization capabilities are significantly influenced by the level of demand loads. Lower demand loads facilitate better learning and higher rewards, while higher demand loads pose challenges that result in greater reward variability and lower performance.

Improvement can be searched in optimizing the learning rate and other hyperparameters to improve stability and performance across different load scenarios.

6.2.3. Scenario 3: Increase in Renewable Energy Supply

In this scenario, it is expected that the renewable energy supply should be increased by a factor of at least 1.2 to align with the turning point on the demand side. However, the intermittent nature of the energy sources can cause this to potentially be higher. The increase in renewable energy generation

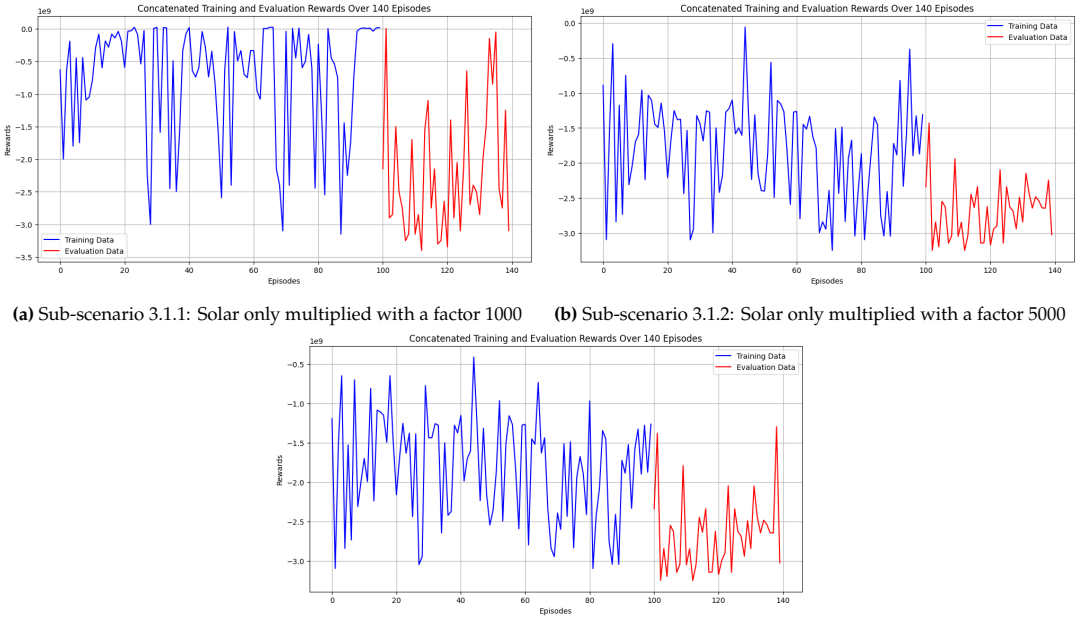
follows a specific pattern, leading to several sub-scenarios:

1. Solar Energy Only
2. Solar Energy Increase
3. Heat pump increase

Each sub-scenario was evaluated to determine its impact on the overall system performance. The focus was on understanding how different forms of renewable energy contribute to the stability and efficiency of the energy system when their supply is increased.

Sub-Scenario 3.1: Solar Energy and Storage Only

In this sub-scenario, the microgrid is configured to rely exclusively on solar energy and storage solutions. The solar energy input is progressively increased until the system reaches convergence. Base parameters are maintained, and the loads are set to normal levels, see Figure 6.5 for the concatenated training and evaluation curves.



(a) Sub-scenario 3.1.1: Solar only multiplied with a factor 1000

(b) Sub-scenario 3.1.2: Solar only multiplied with a factor 5000

(c) Sub-scenario 3.1.3: Solar multiplied with a factor 5000 and storage doubled

Figure 6.5: Scenario 3.1: Concatenated training and evaluation curves

Table 6.5: Summarized version of: Evaluation rewards for sub-scenarios within scenario 3.1, see Table F.4

Episode	Base case	Scenario 3.1.1	Scenario 3.1.2	Scenario 3.1.3
1	-401440312.5	-2148127050	-2338437467	-2338437859
..	..	..	..	..
40	-1701428874	-3095532760	-3023887501	-3023887861
Total	-9.31E+08	-2.22E+09	-2.70E+09	-2.66E+09

Analysis Scenario 3.1

None of the sub-scenarios achieve convergence, as visible in Figure 6.3, indicating that the current model configuration and training parameters are insufficient to handle the increased solar power and storage modifications. The high variability and lack of stable performance suggest that simply increasing solar power and storage without additional adjustments will not yield optimal results.

The absence of constant energy sources contributes to high variability and instability in the microgrid's performance. Renewable sources like solar and wind are inherently intermittent, causing fluctuations

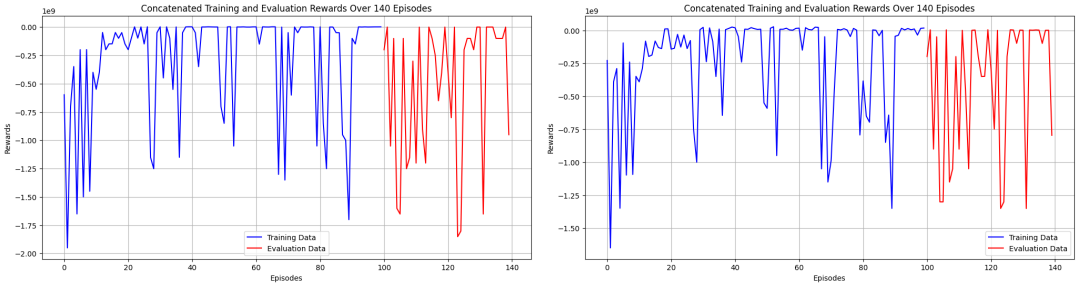
in the energy supply and making it challenging for the microgrid to maintain consistent performance. Consequently, it is logical that the rewards are lower compared to the base case, where there is access to some form of constant energy.

Furthermore, it would not be expected for Scenario 3.1.1 to yield higher rewards than Scenario 3.1.2. In theory, selling excess energy back to the grid should increase the rewards. However, the fact that Scenario 3.1.1 outperformed Scenario 3.1.2 could be due to the inherent variability in the rewards. This variability suggests that the performance observed in Scenario 3.1.1 might have occurred by chance.

The observed results also imply that the process for selling back to the grid may not be correctly implemented, as it does not consistently lead to higher rewards. This needs to be investigated further to ensure that the selling back mechanism is accurately reflected in the model.

Sub-Scenario 3.2: Solar Energy Increase

In this sub-scenario, the generation capacity of solar energy is significantly increased within the microgrid. By multiplying the solar power output by substantial factors, the objective is to assess the effects of heightened solar energy integration on the system's performance, see Figure 6.6 for the concatenated training and evaluation curves.



(a) Sub-scenario 3.2.1: Solar with a factor 100 additional to the base system

(b) Sub-scenario 3.2.2: Solar multiplied with a factor 1000 additional to the base system

Figure 6.6: Scenario 3.2: Concatenated training and evaluation curves

Table 6.6: Summarized version of: Evaluation rewards for sub-scenarios within scenario 3.2, see Table F.5

Episode	Base case	Scenario 3.2.1	Scenario 3.2.2
1	-401440312.5	-200592386.3	-198402177.7
..	..	..	..
40	-1701428874	-950389145.7	-795578282.6
Total	-9.31E+08	-5.12E+08	-3.91E+08

Analysis Scenario 3.2

In Scenario 3.2.1 and Scenario 3.2.2, the impact of multiplying the solar power by factors of 100 and 1000, respectively, is analyzed. The training and evaluation rewards over 140 episodes are examined for both sub-scenarios to understand their impact on the model's performance compared to the base case.

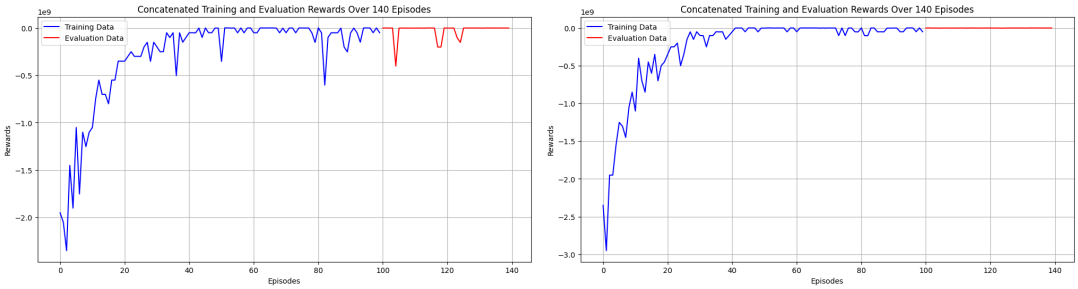
Both scenarios exhibit high variability in training and evaluation rewards, indicating that the model experiences instability during learning and struggles to achieve consistent performance. Increasing the solar power results in greater variability and instability. The model finds it difficult to handle the increased input, leading to frequent and severe drops in rewards. This can help declare the previous scenarios.

Compared to the base case, the rewards in both scenarios are higher, suggesting that the addition of solar power does increase the potential rewards, see Table 6.6. However, the instability indicates that the model requires further optimization to handle these changes effectively, see Figure 6.6.

Sub-Scenario 3.3: Heat Pump Increase

This sub-scenario tests the impact of increasing the generation capacity of heat pumps within the microgrid. By adding higher generation power from heat pumps, the aim is to evaluate how this

additional constant energy source influences the microgrid's performance. See Figure 6.7 for the concatenated training and evaluation curves.



(a) Sub-scenario 3.3.1: Heat pump generation multiplied with a factor 2 additional to the base system

(b) Sub-scenario 3.3.2: Heat pump generation multiplied with a factor 10 additional to the base system

Figure 6.7: Scenario 3.3: Concatenated training and evaluation curves

Table 6.7: Summarized version of: Evaluation rewards for sub-scenarios within scenario 3.3, see Table F.6

Episode	Base case	Scenario 3.3.1	Scenario 3.3.2
1	-401440312.5	-1494583.274	-1494583.274
..	..	..	..
40	-1701428874	-1599882.233	-1599882.233
Total	-9.31E+08	-2.77E+07	-1.40E+06

Analysis Scenario 3.3

The training rewards (blue curve) and evaluation rewards (red curve) over 140 episodes for both sub-scenarios are depicted in Figure 6.7. The training rewards for both sub-scenarios show a consistent upward trend, achieving stability towards the latter episodes. The evaluation rewards exhibit a similar trend, indicating that the model effectively learns and generalizes well with the doubled and tenfold increased of the heat pump generation power.

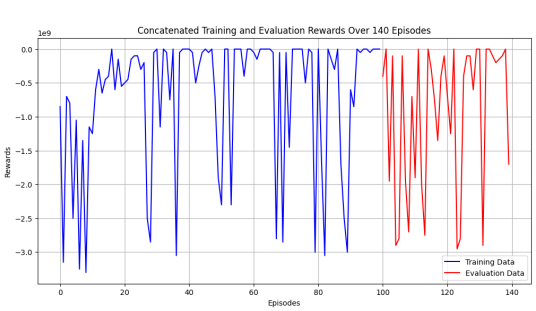
Doubling the generation power of the heat pump results in a substantial improvement in the rewards, as visible in Table 6.7. The model performs much better, with lower negative rewards compared to the base case. This indicates that the microgrid benefits greatly from the additional constant energy, leading to more stable performance. This is expected, as the constant energy directly decreases the needed energy per time step. Increasing the generation power tenfold further enhances the rewards, yielding the highest performance among the sub-scenarios. The stability in both training and evaluation rewards suggests that the model can effectively manage the increased energy input, leading to optimized microgrid operations.

6.2.4. Scenario 4: Increase in (EV) Battery Storage Possibilities

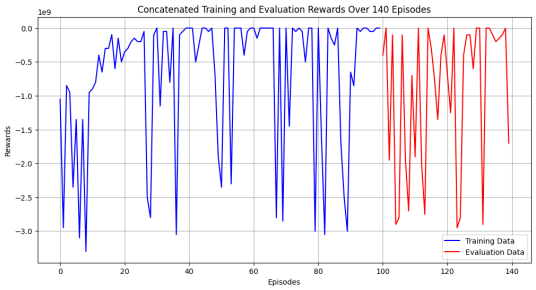
In this scenario, the base case is modified to explore the impact of varying the battery storage capacity and charging speed of (EV) batteries.

Sub-Scenario 4.1: Battery Increase

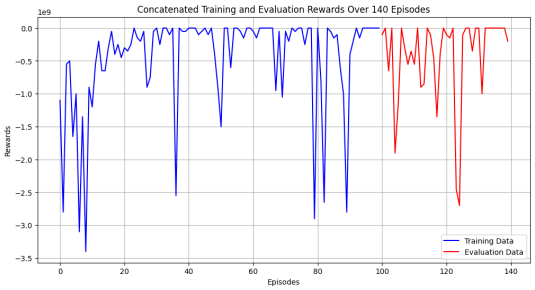
This sub-scenario investigates the effects of varying the battery storage capacity and charging speed within the microgrid. The sub-scenarios involve either increasing the storage capacity, increasing the charging speed, or halving the storage capacity. See Figure 6.8 for the concatenated training and evaluation curves.



(a) Sub-scenario 4.1.1: Doubled amount of battery storage



(b) Sub-scenario 4.1.2: Doubled amount of battery storage and charging speed battery storage



(c) Sub-scenario 4.1.3: Halved amount of battery storage

Figure 6.8: Scenario 4: Concatenated training and evaluation curves

Table 6.8: Summarized version of: Evaluation rewards for sub-scenarios within Scenario 4.1, see Table F.7

Episode	Scenario 1	Scenario 4.1.1	Scenario 4.1.2	Scenario 4.1.3
1	-401440312.5	-401440639.2	-401440428.5	-101166535.1
..	..	..	..	..
40	-1701428874	-1701429021	-1701428959	-201251335
Total	-9.31E+08	-9.31E+08	-9.31E+08	-4.17E+08

Analysis Scenario 4.1

Increasing the battery storage capacity results in improved performance compared to the base case. However, the halved battery storage capacity in sub-scenario 4.1.3 yields the best results, suggesting that under certain conditions, the system might be more efficient with less storage.

When the charging speed is increased along with the storage capacity, the improvements over just increasing the storage capacity alone are marginal. This indicates that while charging speed is a factor, it is not as significant as the storage capacity itself. Additionally, the cost of battery cycles appears to play a significant role in the overall reward. The increased storage capacity leads to more frequent charging and discharging cycles, which can incur higher costs, thereby reducing the net reward. The halved storage scenario results in fewer cycles and lower costs, leading to higher rewards.

The variability in rewards, as indicated by the standard deviation, is highest in Sub-scenario 4.1.1 and 4.1.2, and lowest in Sub-scenario 4.1.3. This suggests that reducing the battery storage capacity results in a more stable performance, potentially due to fewer complexities in managing the storage.

Overall, Scenario 4.1.3, with halved battery storage, achieves the best performance, highlighting the importance of optimizing the storage capacity to achieve stable and efficient microgrid operations. This analysis underscores the need for careful consideration of storage parameters to enhance the system's performance effectively.

Sub-Scenario 4.2: Additional Electric Vehicle Batteries

This sub-scenario investigates the performance of the model when adding batteries that are not always available, like EV batteries. See Figure 6.9 for the concatenated training and evaluation curves for a sub-scenario in which double the amount of batteries is added, but then with availability only during the night.

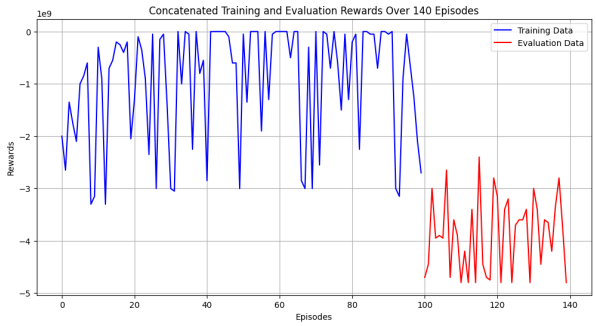


Figure 6.9: Sub-scenario 4.2: 34 Electric Vehicles that can be used as batteries from 18pm to 7am

Table 6.9: Summarized version of: Evaluation rewards for sub-scenarios within Scenario 4.2, see Table F.8

Episode	Scenario 1	Scenario 4.2
1	-401440312.5	-4700146585
..	..	..
40	-1701428874	-4800054279
Total	-9.31E+08	-3.88E+09

Analysis Scenario 4.2

In Sub-Scenario 4.2, where 34 EVs are used as batteries available only from 18pm to 7am, the performance of the model is evaluated. The analysis reveals that the agent struggles to learn an optimal control strategy in this environment. This is evident from the significantly higher negative rewards in Scenario 4.2 compared to Scenario 1, see Figure 6.9 and Table F.8.

The key challenge with EV batteries in the scenario is their intermittent availability and the randomization of their SOC upon re-entering the system. This intermittent availability introduces a high degree of variability and unpredictability, which complicates the agent's task of managing the energy system effectively. Moreover, the variance of the reward in Scenario 4.2 is extremely high, with a standard deviation of 717708463.9252201. This high variance indicates inconsistent performance by the agent, likely due to the added complexity of managing EV batteries that are not always available and whose SOC is randomized.

Additionally, the lack of an intermediate state, such as an idle or saving mode might be a problem. The absence of these states forces the batteries to be either fully engaged or completely inactive, leaving no room for partial engagement that could help balance the system more effectively. This binary state management leads to more abrupt changes in the system's dynamics, further increasing the difficulty for the agent to learn and apply a stable control strategy.

6.2.5. Takeaways

The analyses of various scenarios highlight several critical insights for optimizing microgrid performance. Lower demand loads facilitate better learning and higher rewards, indicating that the agent performs more effectively with reduced complexity. Conversely, higher demand loads introduce more variability and greater cumulative negative rewards, suggesting the need for optimized learning rates and hyperparameters. The integration of renewable energy sources, such as solar power, presents challenges due to their inherent variability. Simply increasing solar power and storage does not yield optimal results without further adjustments. The inclusion of constant energy sources, like heat pumps with increased generation power, significantly enhances performance, demonstrating the importance of integrating reliable energy sources to support intermittent renewables. Battery storage capacity plays a really important role in microgrid efficiency. While increased storage capacity improves performance compared to the base case, the best results are achieved with halved storage, possibly due to reduced operational costs associated with battery cycles and the lack of intermediate states like a saving state or slow charging. The marginal improvements from increased charging speed further emphasize that storage capacity is more critical. These takeaway underscore the need for careful optimization of various

parameters to develop stable, efficient, and cost-effective energy management systems, providing guidance for future research and practical implementations in microgrid optimization.

6.3. Hybrid Scenarios

For the hybrid scenario, four sub-scenarios are developed, each reflecting different levels of technological advancements, societal willingness (private decisions), and demand reductions. These scenarios are derived from the expectations outlined in the report and are designed to illustrate a range of possible futures. Each scenario aims to configure the hybrid microgrid components in a way that aligns with the projected energy scenarios, ensuring both feasibility and operational efficiency. The sub-scenario yielding the highest rewards will undergo extended runs and hyperparameter optimization to refine performance further. It is anticipated that a scenario characterized by neither technological advancements nor societal willingness will fail to provide a system where the agent can identify an optimal solution, similar to the base case. Therefore, this is not considered.

Willingness includes both the public willingness from the government, but also private to reduce demand and to invest in technologies such as heat pumps. Storage advancements are assumed to be driven by government initiatives and are thus related to public, governmental, investments.

The four future scenarios are:

1. **Low Technological Advancement and High Public Organizational Willingness (LT-HPW)**: Technological advancements are limited, but there is strong governmental willingness to reduce demand and invest in available technologies. The government drives the adoption of existing technologies through policies and subsidies.
2. **Low Technological Advancement and High Overall Willingness (LT-HW)**: Technological advancements are limited in this scenario, but societal willingness to reduce demand and invest in available technologies is high. The public compensates for the lack of advanced technologies through strong engagement and investment.
3. **High Technological Advancement and Low Willingness (HT-LW)**: Here, technological advancements are substantial, but both societal and governmental willingness are low. Despite the availability of advanced technologies, engagement in demand reduction and technology adoption is minimal.
4. **High Technological Advancement and High Public Organizational Willingness (HT-HPW)**: Technological advancements are substantial, and there is strong governmental willingness to drive adoption. The government plays a major role in reducing demand and investing in new technologies, despite limited private engagement.
5. **High Technological Advancement and High Overall Willingness (HT-HW)**: In this scenario, both technological advancements and societal willingness are high. There is significant investment in cutting-edge technologies and a strong drive from both the government and the public towards reducing energy demand and adopting new technologies.

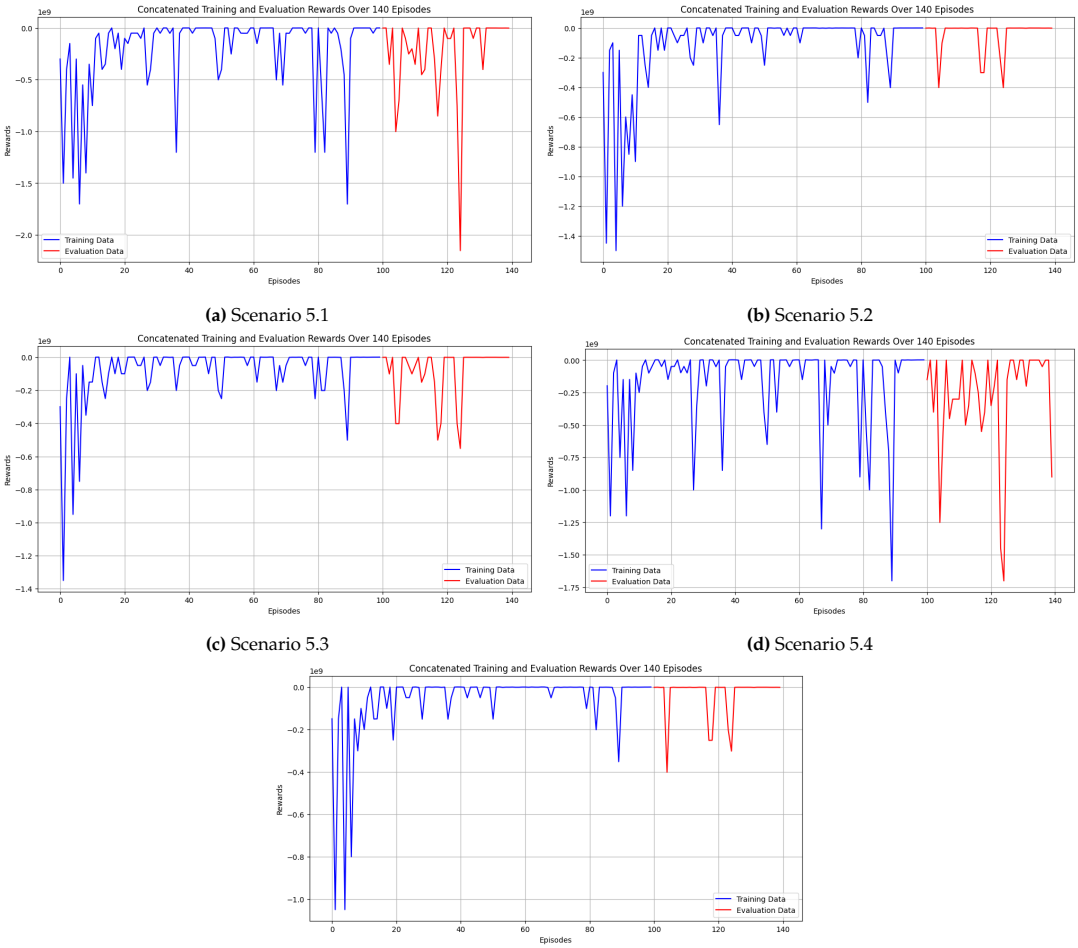
Sub-scenario Set Up

Based on the insights gained from the previous analyses, see Subsection 6.2.5 a set of hybrid scenarios has been developed to explore the combined effects of various DERs and load configurations on microgrid performance. Each sub-scenario varies in terms of load, heat pump generation, solar and wind power, and battery storage to evaluate different combinations and their impacts, see Table 6.10

DER	Scenario 5.1	Scenario 5.2	Scenario 5.3	Scenario 5.4	Scenario 5.5
Load	0.9	0.8	0.8	0.9	0.8
Heat pump	1.25	1.5	1.25	1	1.5
Solar	10	10	50	100	50
Wind	10	10	50	100	50
Batteries	Doubled	Doubled	Base	3-Doubled	Doubled

Table 6.10: Parameters and their values for Sub-scenarios 5.1 to 5.5

The concatenated curves for all the hybrid scenarios are presented in Figure 6.10.



(a) Scenario 5.1

(b) Scenario 5.2

(c) Scenario 5.3

(d) Scenario 5.4

(e) Scenario 5.5

Figure 6.10: Scenario 5: Concatenated training and evaluation curves

6.3.1. Analysis Hybrid Scenarios

The rewards of the scenarios are presented in Table 6.11.

Table 6.11: Summarized version of: Evaluation rewards for sub-scenarios within Scenario 5, see Table F.9

Episode	Scenario 5.1	Scenario 5.2	Scenario 5.3	Scenario 5.4	Scenario 5.5
1	-1078495.003	-928628.5067	-566398.8355	-150477870.2	-567237.9379
..	..	..	..	..	..
40	-1191068.577	-1030657.942	-559543.2194	-900327764	-560689.1198
Total	-2.75E+08	-2.25E+08	-8.43E+07	-4.34E+07	-3.56E+07

Table 6.12: Statistics of evaluation data for various scenarios

File	Mean Reward	Variance of Reward	Standard Deviation of Reward
Scenario5.1	-224752742.35169205	1.6840748210430454e+17	410374806.85868686
Scenario5.2	-43392903.7198508	1.2313632886160768e+16	110966809.84042376
Scenario5.3	-84313661.45816116	2.5542602166707452e+16	159820531.11758655
Scenario5.4	-275403589.82001877	1.6580104160207674e+17	407186740.4546429
Scenario5.5	**-35571728.82936376**	9438798313147836.0	97153478.13201459

Among the evaluated scenarios, Scenario 5.5 provides the most desired balance of mean reward, variance, and standard deviation for both the training and evaluation data, see Table 6.12. This scenarios offer the most consistent and reliable performance during the evaluation phase.

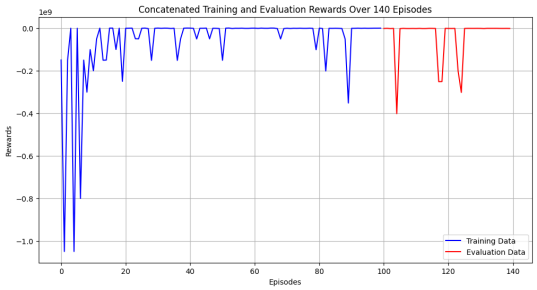
Investment costs and feasibility are, however, also crucial factors in determining the attractiveness of different scenarios. While scenarios with high levels of constant energy, such as 5.2 and 5.5, perform well, they are also more challenging and costly to implement. This highlights the need for a balanced approach that considers both performance and feasibility. The investment costs associated with different setups must be weighed against the potential performance gains to determine the most practical and effective solutions.

It is important to note that while some scenarios may be more desirable in terms of performance, factors such as feasibility and cost must also be considered. The differences in performance across scenarios provide valuable insights into what is achievable with various configurations, showcasing the potential benefits and challenges of each setup. This analysis helps identify the most desirable configurations based on a balance of performance, feasibility, and cost.

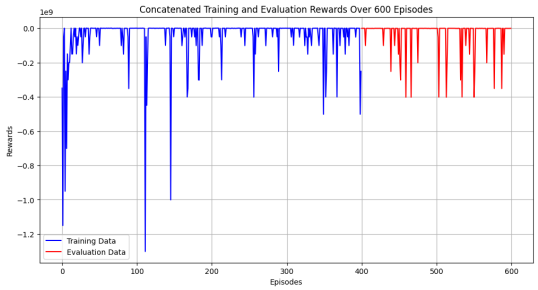
Overall, the analysis emphasizes the need for a holistic approach in optimizing microgrid performance. By integrating multiple DERs, optimizing storage parameters, and considering both technological advancements and societal willingness, it is possible to achieve stable, efficient, and cost-effective energy management systems.

6.3.2. Longer Training and Evaluation

For further investigation into the most desired scenario, Scenario 5.5, the performance is examined. A longer training and evaluation phase is utilized to showcase the behavior, and potential improvements.



(a) Sub-scenario 5.5.1: Training curve for 100 training episodes and 40 evaluation episodes



(b) Sub-scenario 5.5.2: Training curve for 400 training episodes and 200 evaluation episodes

Figure 6.11: Concatenated training and evaluation curves for sub-scenarios 5.5.1 and 5.5.Long

A comparison of the total rewards for Scenario 5.5.1 and Scenario 5.5.2, as summarized in Table 6.14 and the statistics, presented in Table 6.13, indicates that extended training and evaluation lead to a significant improvement in performance for Scenario 5.5.2.

Table 6.13: Comparison of statistics Scenario 5.5 Long

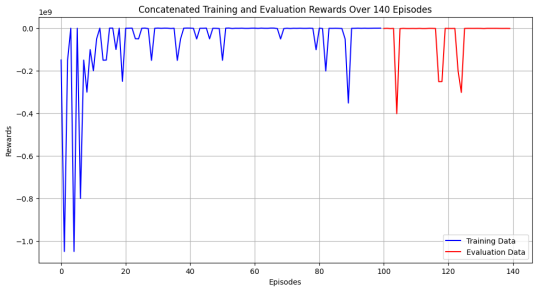
Metric	**Evaluation Scenario 5.5.1**	**Evaluation Scenario 5.5.Long**
Mean Reward	-35,571,728.83	-30,061,503.77
Variance of Reward	9,438,798,313,147,836.0	8,120,566,566,624,243.0
Standard Deviation of Reward	97,153,478.13	90,114,186.27

Table 6.14: Summarized version of: Evaluation rewards for Scenario 5.5 and a longer Training and Evaluation, see Table F.10

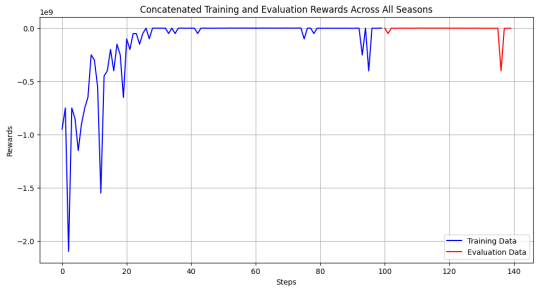
Episode	**Scenario 5.5.1**	Episode	**Scenario 5.5.2**
1	-567237.9379	1	-291967.8563
..	..	..	..
40	-560689.1198	200	-131945.7693
Total	-3.56E+07	Total	-3.01E+07

6.3.3. Seasonal Variations in Training

The most desired scenario is also trained on a seasonal data split, in order to analyze the impact of the data split, see Figure 4.6. The concatenated training and evaluation data is presented in Figure 6.12.



(a) Sub-scenario 5.5.1: Training curve for 100 training episodes and 40 evaluation episodes



(b) Sub-scenario 5.5.2: Training curve for Scenario 5.5 but with data split per season

Figure 6.12: Concatenated training and evaluation curves for sub-scenarios 5.5.1 and 5.5.Seasonal

The evaluation of the two different datasets reveals notable differences in performance metrics, as shown in Table 6.15. The mean reward for the evaluation data only is significantly lower than for the seasonal data split, indicating better performance for the latter. Additionally, the variance and standard deviation of the reward are both lower for the seasonal data split, suggesting more consistent and

reliable results. These observations suggest that incorporating a seasonal data split into the evaluation process enhances both the average performance and the consistency of the rewards.

Table 6.15: Comparison of statistics Scenario 5.5 Seasonal

Metric	Evaluation Scenario 5.5.1	Evaluation Scenario 5.5.1 with Seasonal Data Split
Mean Reward	-35,571,728.83	-11,278,114.11
Variance of Reward	9,438,798,313,147,836.0	4,071,995,653,426,960.5
Standard Deviation of Reward	97,153,478.13	63,812,190.48

Table 6.16: Summarized version of: Evaluation rewards for Scenario 5.5.1 and Scenario 5.5.Seasonal, see Table F.11

Episode	5.5.1 Reward	Episode	Reward
1	-567237.9379	1	47057.77649
..	..	..	..
40	-560689.1198	10	-322036.3121
Total	**-3.56E+07**	**Total**	**-1.13E+07**

The evaluation of the two different datasets reveals notable differences in performance metrics, as shown in Table 6.15. The mean reward for the evaluation data only is significantly lower than for the seasonal data split, indicating better performance for the latter. Additionally, the variance and standard deviation of the reward are both lower for the seasonal data split, suggesting more consistent and reliable results. These observations suggest that incorporating a seasonal data split into the evaluation process enhances both the average performance and the consistency of the rewards.

6.3.4. Hyperparameter Tuning

The optimization of the hyperparameters is crucial to enhance the performance of the DQN model. To achieve this, the Optuna library is employed, which is an open-source Python library designed to automate the search for optimal hyperparameter configurations in machine learning models (Akiba et al., 2019).

Optuna requires the input of the hyperparameters to be optimized, their acceptable ranges, the objective function to be maximized or minimized, and the sampling algorithm used in the optimization process. The ranges and incremental steps for each hyperparameter are defined to guide the optimization process effectively. For the most desired scenario, the Optuna hyperparameter tuning is performed, focusing on the variables gamma, learning rate, and epsilon decay. These parameters have been selected due to their significant impact on the rewards within the standardized training environment, see Table 5.1. The values that are used for the Optuna optimization are presented in Table 6.17.

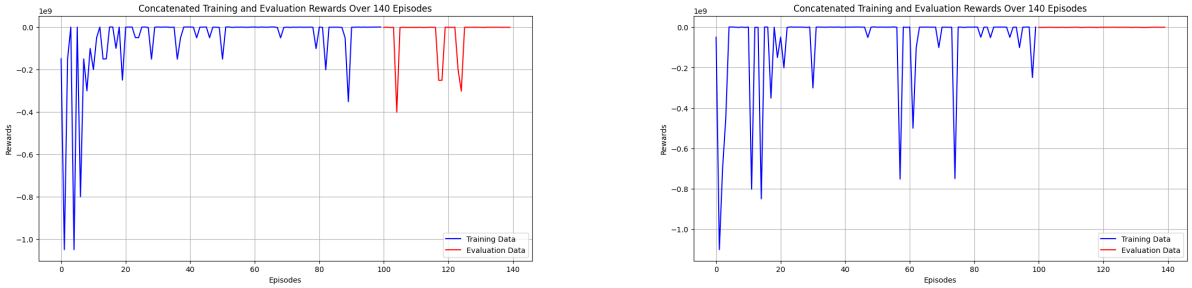
Table 6.17: Hyper parameters used in the DQN model optimization and their default values

Hyperparameter	Optuna Range/Value	Description
learning_rate	[1e-5, 1e-2]	Learning rate for the DQN model, important for controlling the update magnitude of network weights.
gamma	[0.9, 1]	Discount factor, which influences how much future rewards are valued over immediate rewards.
epsilon_decay	[0.9, 0.999]	Epsilon decay rate, used to adjust the exploration-exploitation balance in training.

The best run is trial 26, with a value of -476553.8936589475. See Figure 6.13b for the training curve. The results of an optimization of the parameters for scenario 5.5 with 30 trail runs are as follows:

Table 6.18: Comparison of statistics Scenario 5.5 Optuna

Metric	Evaluation Scenario 5.5.1	Evaluation Scenario 5.5.1 Optuna
Mean Reward	-35,571,728.83	-476,553.89
Variance of Reward	9,438,798,313,147,836.0	181,708,558,680.30
Standard Deviation of Reward	97,153,478.13	426,272.87



(a) Scenario 5.5.1

(b) Scenario 5.5.1 ran with Optuna optimized parameter

Figure 6.13: Concatenated training and evaluation curves for sub-scenarios 5.5.1 and 5.5.Optuna

- 'learning_rate': 2.01605258096069e-05,
- 'gamma': 0.9281892353169647,
- 'epsilon_decay': 0.929251846973606

In Figure 6.13b, the model is optimized with Optuna, and the corresponding training and evaluation rewards are shown. The training rewards still demonstrate fluctuations, but there are more consistent periods of higher rewards compared to Scenario 5.5.1. Furthermore, a comparison of the total rewards for Scenario 5.5.1 and Scenario 5.5.1 with Optuna, as summarized in Table 6.19 indicates that optimization with Optuna leads to a significant improvement in performance.

Table 6.19: Summarized version of: Evaluation rewards for Scenario 5.5 and a Optuna optimization, see Table F.12

Episode	Scenario 5.5.1	Scenario 5.5.Optuna
1	-567237.9379	-254137.8673
..	..	
40	-560689.1198	-55156.87253
Total	-3.56E+07	-4.77E+05

The analysis reveals that optimization with Optuna leads to a significant improvement in performance. The total negative rewards for Scenario 5.5.1 with Optuna are considerably lower than those for Scenario 5.5.1, indicating that the model performs better with optimized hyperparameters. The training rewards for Scenario 5.5.1 with Optuna show reduced variability, suggesting that the optimization process has helped stabilize the learning process.

6.3.5. Takeaways

All of the extensions on Scenario 5.5 have led to improvements in performance. Extended training and evaluation periods, demonstrated significant improvements over Scenario 5.5.1. The total negative rewards were considerably lower, and the training rewards exhibited reduced variability, indicating a more stable learning process. A seasonal data split resulted in improved performance metrics. The mean reward was higher, and both the variance and standard deviation of rewards were lower compared to the non-seasonal split. Lastly, especially the Optuna hyperparameter tuning led to significant improvements. The optimized hyperparameters resulted in a lower total negative reward and reduced variability in training rewards.

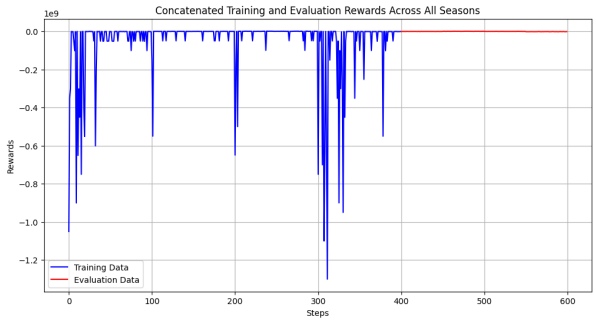


Figure 6.14: Concatenated training and evaluation curves scenario 5.5 with long training and evaluation, optimized hyperparameter, and a seasonal data split.

Combining all three enhancements—extended training and evaluation, seasonal data split, and hyperparameter tuning—is ran to test potential further improvement of the performance of the DQN model, see Figure 6.14. This has led to even further improvement of the model.

Table 6.20: Comparison of statistics sub-scenarios Scenario 5.5

Metric	Scenario 5.5.1	Scenario 5.5.Long	Scenario 5.5.Seasonal	Scenario 5.5.Optuna	Scenario 5.5.All Methods 2
Mean Reward	-35,571,728.83	-30,061,503.77	-11,604,205.55	-476,553.89	-31,693.07
Variance of Reward	9,438,798,313,147,836.0	8,120,566,566,624,243.0	4,070,520,256,847,905.5	181,708,558,680.30	462,848,112,861.20
Standard Deviation of Reward	97,153,478.13	90,114,186.27	63,800,628.97	426,272.87	680,329.41

6.4. Findings

For the hybrid scenarios, multiple well-performing configurations were combined to evaluate their effectiveness. Scenario 5.5, developed within a context of high technological advancements and societal willingness, achieved the highest average evaluation rewards and the lowest variability. These findings underscore the importance of a balanced approach in integrating multiple DERs and optimizing storage parameters to enhance microgrid performance.

While scenario 5.5 is the most preferable due to their high levels of constant energy, they present challenges related to significant technological and infrastructural investments. Their consistent energy sources contribute to superior performance, but the feasibility of implementing these scenarios must be carefully considered against the required resources and effort.

Among the sub-scenarios for scenario 5, the Optuna Scenario stands out as the most desirable, demonstrating the highest mean reward and the lowest standard deviation, indicating superior overall performance and consistency.

Key Findings:

- **Performance of Scenario 5.5:** Scenario 5.5, incorporating a hybrid setup with optimized parameters, achieved the best performance among all tested scenarios. When extendening it with a longer training and evaluation time, a seasonal data split, and tuned hyperparameter the performance increased significantly. It demonstrated the highest mean reward and the lowest standard deviation, highlighting its superior performance and consistency.
- **Variability in Performance:** The variability in rewards for sub-scenarios within Scenario 4 was highest in those with increased storage capacity and charging speed, and lowest in the scenario with halved battery storage. This suggests that reducing battery storage capacity results in more stable performance, potentially due to fewer complexities in managing the storage. The reduced variability in Scenario 5.5.2 indicates that extended training and evaluation helped stabilize the learning process, leading to more reliable outcomes. The lack of an idle option introduces additional complexity, suggesting a need for strategies to manage these complexities more effectively.
- **Feasibility and Costs:** The investment costs and feasibility of implementing different scenarios significantly impact their attractiveness. Scenarios with high levels of constant energy, such as Scenario 5.5, perform well but are also more challenging and costly to achieve. This highlights the

need for a balanced approach that considers both performance and feasibility to ensure practical and cost-effective implementations.

- **Potential of Hybrid Scenarios:** The hybrid scenarios provide valuable insights into the achievable benefits and challenges of different configurations. They illustrate the potential advantages and difficulties of various setups, guiding future research and practical implementations toward more sustainable and efficient energy systems.
- **EV Scenarios:** Analysis of the storage and EV battery integration revealed significant challenges due to their intermittent availability and the randomization of their SOC. The absence of intermediate states such as idle or saving modes might be the cause of the problems, leading to higher variability and inconsistency in performance. The model's representation of EVs and batteries lacks realism due to the absence of intermediate states like saving and idle. This simplification introduces significant variability and unpredictability, complicating the energy management process. Future research should focus on incorporating these intermediate states to create more realistic and manageable scenarios.
- **Performance Model:** Enhancements in Scenario 5.5, such as extending the training duration, incorporating varied training methods, and optimizing hyperparameters, underscore the importance of scenario-dependent model tuning. These improvements demonstrate that substantial progress can be achieved by tailoring the performance model to the specific scenario requirements.

6.5. Solutions: Actions and Configuration

The solutions derived from the optimal scenario involve determining the most efficient combination of components in terms of quantity. This approach not only identifies the best mix of resources but also provides a detailed action list outlining the optimal steps to take at each moment. By analyzing the various sub-scenarios, we can understand how different load factors affect the overall performance and efficiency of the system.

The action list are actions per time step and they are presented in a vector. This vector is the input of the visualization that showcases how a scenaric surrounding with its component would act according to the trained agent in order to maximize the rewards, and thus the independency and costs efficiency.

7

Results Visualization

In this chapter, the visualization process is presented to answer *Q3*. The POC configuration is presented, followed by feedback gathered from an expert panel. The results from this panel are prioritized, and the processed feedback is discussed.

7.1. Operational Results and Feedback

The DT is developed using ArcGIS and is showcased with ArcGIS Experience Builder, with data loaded through locally hosted, pre-processed temporal files. The POC encompasses the most desired scenario that runs for the timespan of a year. This scenario displays the system components and their states over time using a temporal slider. The states represent actions for controllable elements and demand or generation for non-controllable elements.

Controllable and non-controllable elements are defined in the Pymgrid section. Due to the large data volumes, to ensure smooth operation of the demo, the script runs with two sets of parameters, and there is access to the pre-processed documents. For future applications, the DT can be linked to the script via the same pathway, allowing data to be directly loaded into the Web Layer. Parameters are transmitted to the system through a QGIS file, and then the agent can be trained with the script on hosts GPUs.

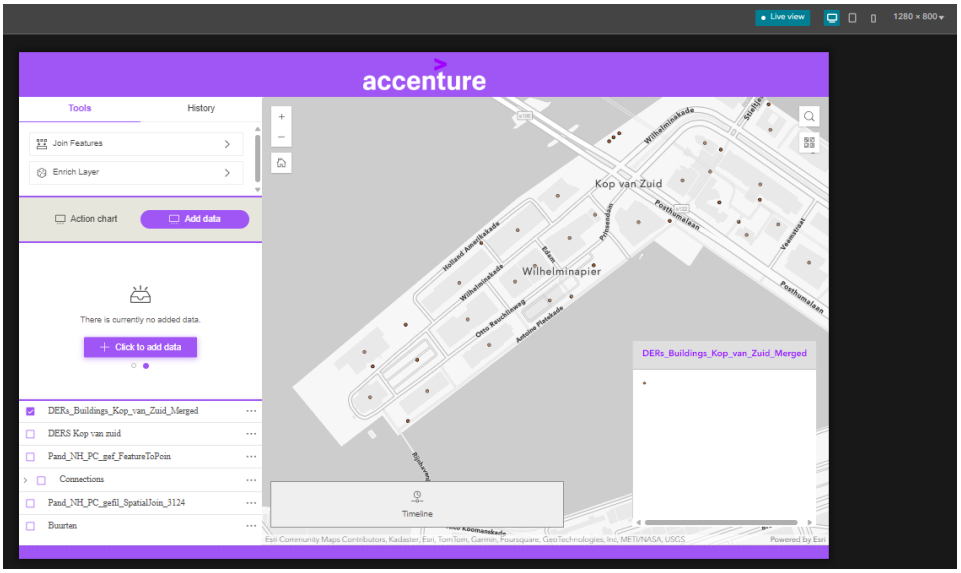


Figure 7.1: ArcGIS Experience Builder showcasing proof of concept digital twinn

7.2. Visualization Feedback Panel

Within the company, experts in the field are identified and selected through internal referrals. Participants in the feedback panel are asked to use the DT, and feedback is gathered through direct verbal feedback and a feedback form. Participants are asked to focus on the operational value and potential enhancements of the DT.

The development and implementation of a visualization tool highlights the elements needed for effective representation of the optimization. By enabling a detailed and dynamic representation of microgrid operations, a DT can provide stakeholders with clear and actionable insights into the effects of various optimization strategies. The feedback loop through the expert panel refines a visualization tools functionality, ensuring it remains aligned with user needs and industry standards, enhancing its reliability and efficacy.

The iterative feedback process showcased the importance of scalability and modularity, directing future development efforts to focus on these aspects to meet the evolving demands of energy management systems.

7.3. Feedback Processing and Value Creation

The remainder of this research emphasizes enhancing the modularity and standardization of a DT, promoting internal dissemination, and outlining potential future steps. The focus areas include:

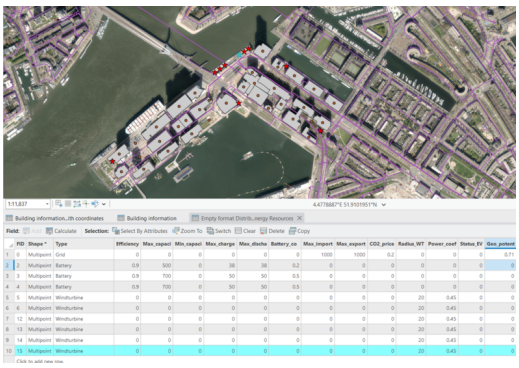
- **Modularity Enhancement:** Improving the modularity of the DT to ensure that individual components can be easily updated or replaced without affecting the entire system. This will allow for more flexible and scalable solutions in future developments. Developing standardized elements and workflows to ensure consistency and reliability across different use cases. This involves creating a uniform framework for data input, processing, and visualization that can be applied to various scenarios and case studies.
- **Internal Dissemination:** Facilitating the sharing of knowledge and results within the organization through comprehensive documentation, training sessions, and presentations. This will help integrate the DT into the company's regular operations and encourage the adoption of the tool by other departments. Identifying and planning potential future steps for the GIS team.

7.4. Improved Workflows

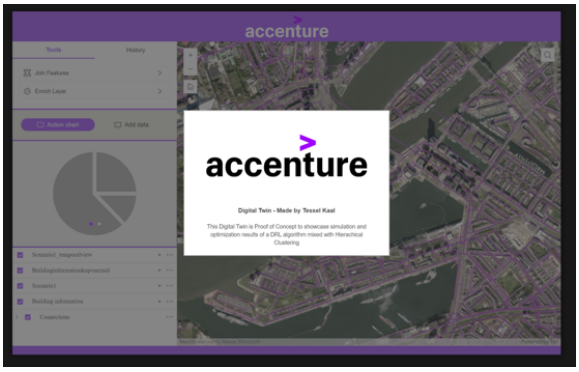
A formatted shapefile is created to serve as the framework for further scenario setups, see Figure 7.2a. This framework functions as a format for the input of the optimization and must provide compatibility to the manner in which a DT is created. The shape file, thus must be used for the optimization as well as for the DT.

The values can be adjusted if geothermal power is not available or if there are changes in import/export capacities. DERs are added to this framework, and the relevant variables are filled in as needed.

The completed shapefile is exported as a CSV file with X and Y coordinates, which serves as the input for running the code. The empty format is available in the GitHub repository, as detailed in Appendix E.



(a) Standardized GIS format for the POC digital twin



(b) Opening screen of the DT and internal presentation

Figure 7.2: Proof of concept digital twin in Arcgis Experience Builder

To create value within the company, a slide deck is created to share the findings and capabilities of the DT, see Figure 7.2b.

7.5. Final Visualization

The application is showcased internally through ArcGIS Experience Builder but hosted on ArcGIS Online (AGOL) as a map layer to ensure it remains openly accessible for showcasing further research.

The input data includes all geographical information and the processing of timely data from the energy simulation and optimization. The input from the optimization results in a state vector that contains a number representing the action the agent chooses to perform per time step. This is a number from 0 to the length of the action space, indicating actions such as a battery charging or discharging and heat pumps generating energy.

While the current visualization serves as a POC, it demonstrates the potential of this approach to create a comprehensive DT tool. This POC lays a strong foundation for stakeholders to further develop and refine the DT, ultimately transforming it into a fully functional tool that can provide real-time data integration and actionable insights. By setting this groundwork, the POC visualization highlights the feasibility and value of such a tool, ensuring stakeholders can effectively leverage it for informed decision-making and enhanced energy management practices.

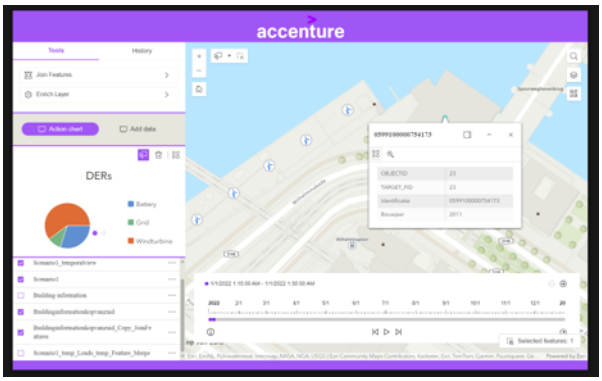


Figure 7.3: Final digital twin with temporal slider

By making the DT accessible and interactive, stakeholders can better understand and analyze the microgrid's performance, thereby facilitating informed decision-making and promoting more effective energy management practices.

The POC, that is hosted on AGOL, can be seen in a demo clip via the following link: DT demo and the AGOL host page.

8

Conclusions

In this chapter, the discussion and limitations of the research are presented first. Thereafter, the conclusion, which also serves as the foundation for key recommendations, is presented. The conclusions drawn from this research directly inform the subsequent recommendations for future work. These recommendations are not only based on the findings but also suggest practical steps for further exploration, effectively bridging the gap between the current research outcomes and potential future inquiries. This overlap ensures that the recommendations extend logically from the detailed conclusions, highlighting their dual role in both summarizing the present study and guiding future research efforts.

8.1. Discussion and Limitations

The purpose of this study was to contribute to the implementation of sustainable energy grids by developing a comprehensive framework for simulating and optimizing microgrid scenarios. Key findings include the identification of critical modeling aspects, the application of reinforcement learning for optimization, and the development of effective visualization tools.

The created pipeline provides a foundation for further exploration and implementation of algorithms, allowing for the testing of various case studies to manage and deploy microgrids effectively. This capability is crucial for advancing the deployment of microgrid solutions and ensuring their efficient and sustainable operation.

8.1.1. Computational Limitations

The extensive computation time required for simulations posed significant restrictions, limiting the scope. The computational limitations necessitated the aggregation of components and prevented the inclusion of distance factors, what was originally aimed to be optimized in within the framework. This issue underscores the need for more sophisticated computational strategies or advanced hardware capable of handling complex simulations and the answer to this is actually that it is thus not possible with the implemented algorithm, and another novel research method is proposed to overcome these limitations. Future research should explore more efficient algorithms and high-performance computing resources to mitigate these limitations. Implementing these advanced techniques requires specialized expertise and further research to develop and optimize their application in the context of microgrid simulations.

8.1.2. Multiple Algorithms

For the case study, DQN is implemented. However, implementing multiple algorithms may allow for a more robust analysis and comparison of different approaches to optimize the microgrid scenarios and may bring more insights. Each algorithm has unique strengths and weaknesses, and employing a variety of methods can provide a more comprehensive understanding of potential solutions.

In the case study, the DQN algorithm was implemented to optimize the microgrid scenarios. However, employing multiple algorithms could offer a more robust analysis and a deeper understanding of the optimization landscape. Implementing a range of algorithms, such as Double Deep Q-Networks,

Proximal Policy Optimization, and Soft Actor-Critic, can facilitate a comprehensive comparison of different approaches. Some characteristics are for example:

- **DDQN** addresses this bias by using separate networks for action selection and evaluation, potentially leading to more accurate value estimates.
- **PPO** offers stability and efficiency in training by maintaining a balance between exploration and exploitation, making it suitable for continuous action spaces.
- **SAC** provides a robust framework for handling complex environments with continuous actions, leveraging entropy regularization to encourage exploration.

Also, combining methods like DQN and DDQN with policy-based methods like PPO and SAC can bring value by helping to address various challenges, such as dealing with uncertainties in demand and supply, or adapting to dynamic changes in the environment.

8.1.3. Implementation and Machine Learning Limitations

Implementing machine learning models in real-world energy systems presents several challenges. A significant limitation is the difficulty in training these models on real data. Acquiring high-quality, real-world data is challenging due to privacy concerns, high data collection costs, and the complexity of the required data. Additionally, deploying these models in real-life scenarios involves considerable costs and risks, which might still be prohibitively high at this stage.

These limitations necessitate a cautious approach to implementation and adjustment of the current grid. The initial risks are substantial due to the shorter training period of the agents, increasing the likelihood that the model may not perform as desired. Failure at this stage could lead to a loss of trust in the technology and foster political opposition. Therefore, the learning process is crucial, and developing comprehensive simulation environments that approximate real-world conditions is essential for the safe and cost-effective testing of the energy model. Future research should focus on creating these environments and exploring alternative data sources to enhance the model's accuracy and reliability.

Biases Introduced by Reinforcement Learning

While effective in navigating complex decision spaces, the RL framework may introduce biases that could skew optimization towards certain configurations or operational modes (Chen et al., 2022). These biases can arise from several sources, including the reward function design, the training data distribution, and the exploration-exploitation balance. A detailed exploration of specific instances where biases were observed during this research could provide valuable insights. For example, certain states or actions might be favored disproportionately due to the structure of the reward function, leading to sub optimal or unfair outcomes. Additionally, the training process might over fit to particular scenarios presented in the training data, thereby limiting the generalizability of the model to unseen situations.

As already mentioned, it can be favorable to test multiple algorithms. One approach is to employ techniques such as DDQN, which have been shown to reduce the overestimation bias inherent in standard Q-learning methods. DDQN achieves this by decoupling the action selection from the Q-value evaluation, thereby providing more accurate value estimates.

Model-Free Approach

In the context of using a model-free method in power systems, integrating it with established model-based approaches offers several promising benefits. Although model-free methods are powerful due to their ability to learn and adapt from data directly, they sometimes lack the ability to utilize existing model knowledge, which can lead to challenges in safety and scalability (Chen et al., 2022). Combining the flexibility of model-free methods with the robustness and theoretical foundation of model-based approaches could enhance overall system effectiveness. Future research should explore hybrid approaches that leverage the strengths of both methodologies.

Forecasts and AI Limitations

Hybrid techniques have been shown to outperform standalone methods in accuracy, particularly in forecasts. However, the effectiveness of machine learning and deep learning methods like Convolutional Neural networks and Recurrent Neural Networks decreases significantly for forecasts over longer time horizons or under variable conditions such as cloudy weather (Tajjour & Singh Chandel, 2023).

8.1.4. Impact of the Reward Function

The reward function, central to minimizing grid dependence and operational costs, is instrumental in steering the system's optimization processes. Further investigation into how this function influenced decision-making in various scenarios could provide deeper insights into its overall efficiency and potential unintended behaviors. Refining the reward function to better capture real-world complexities is essential for improving the model's performance. These suggestions are further explained in the recommendations, see Section 8.4

8.1.5. Achieving More Realistic Behavior

The model can be enhanced to achieve more realistic behavior in several ways. Currently, it utilizes aggregated information for load balancing, whereas the original intention was to employ more localized balancing and energy sharing. Due to constraints, this was not feasible. Incorporating localized balancing and energy sharing in future iterations will significantly improve the model's accuracy. Additionally, more complex models for EV usage and plug-in/out times could be integrated to enhance realism. The current implementation and potential improvements are discussed below.

Heat Pumps

In the current model, heat pumps are defined as a genset, which neglects their energy conversion demands. This simplification should be addressed to accurately reflect their impact on energy management. Future models should incorporate the energy consumption of heat pumps to provide a more precise representation of their role in the microgrid. Extending the pymgrid library to include this feature, as well as adding EVs as a separate module, is currently in progress.

Zonatlas Data Accessibility

The lack of access to Zonatlas data posed a significant limitation. The Zonatlas API provides detailed, time-specific data on roof suitability and solar potential, which would enhance the precision of our estimations. Future research should prioritize securing access to such APIs to improve data accuracy and overall model reliability.

8.1.6. Transmission Losses

Non-technical losses, such as electricity theft and billing inaccuracies, play a significant role in distribution losses (CEER, 2020). These losses are often influenced by socio-economic and regulatory factors rather than just physical distances. Addressing these issues requires comprehensive solutions that consider socio-economic contexts. Future research should explore the socio-economic factors affecting non-technical losses in various regions to provide valuable context and potential solutions.

Hydrogen

The current study did not include hydrogen due to the need for more complex approaches to determine the modeling and the unavailability of ready data and the scope. Hydrogen is critical for energy storage and grid balancing, and its integration into future models can provide additional flexibility and resilience to microgrid operations. Although this study focused on immediate potential, future research should include hydrogen storage systems, supported by comprehensive data, to fully explore its benefits.

Role of V2G in Renewable Energy Penetration

V2G technologies support renewable energy integration but are limited by high costs and accelerated battery degradation due to frequent charging cycles. Discussing potential technological advancements or policy measures that could mitigate these challenges would be beneficial (Richardson, 2013), and (Hannan et al., 2022). Future research should explore these aspects to enhance the feasibility of V2G technologies.

Cost Cycles and Batteries

The current model uses simplified parameters for battery charging cycles and costs. Future studies need to carefully tune and validate these parameters. Consultation with experts and industry stakeholders can provide insights for refining these parameters, leading to more accurate modeling and cost assessments.

Availability of Main Grid
The model assumes that the main power grid provides an unrestricted power supply. This assumption simplifies the energy management system and the microgrid's operational simulations, portraying the grid as a fail-safe mechanism. However, this does not reflect the complex realities of grid operations, which are subject to limitations like transmission constraints (Van et al., 2023). These constraints can lead to curtailment of renewable sources or load shedding during periods of extreme production or low demand. Future models should account for these limitations to provide a more accurate representation of grid operations.

8.1.7. Visualization and Database Connection
Effective visualization and robust database connectivity are critical for the efficient management and utilization of data in our model. Currently, the lack of an integrated database restricts the seamless use and management of the extensive information the model contains. Establishing a standardized database setup can significantly enhance the data pipeline. Additionally, the absence of a live connection to platforms, like the Royal Netherlands Meteorological Institute (2024a) or the EBN (2023), highlights an area for improvement, as such connections can add substantial value and stabilize the data pipeline. Utilizing PostgreSQL for database integration is a suitable approach, as it aligns with the existing model setup and can support real-time data links, such as the one used for water flow in the example DT. The ArcGIS Builder layer, directly connected to the PostgreSQL database, can be managed and updated efficiently using DataGrip for PostgreSQL. By enhancing the integration and functionality of these connections, we can improve overall system performance and reliability.

8.1.8. Visualization and Compatibility
The current model setup involves extensive preprocessing and conversion of various data types, primarily due to the misalignment of these data types and the uncommon combination of geospatial data with such computations. Enhancing compatibility across different data types and geospatial computations is crucial to ensure seamless integration and functionality across diverse platforms and devices. Improving compatibility will make visualization tools more accessible and user-friendly, facilitating better decision-making processes and enhancing the overall efficiency of the system.

8.2. Research Overview
For every sub-research question, the main findings are presented here. These answers are supported by evidence presented in earlier chapters.

Q1: *Which modeling aspects need to be considered when designing a microgrid and it's energy management system, and how are these reflected in the existing literature?*

The design of a microgrid and its energy management system necessitates consideration of various modeling aspects, as highlighted in both the theoretical framework and the existing literature. These aspects include the components of the microgrid, their interactions, and the optimization strategies employed to manage energy resources efficiently.

A microgrid's configuration can include a diverse array of components, each playing a distinct role. Critical components often include renewable energy sources such as solar, wind, and geothermal energy, which provide sustainable energy and reduce dependency on fossil fuels. Additionally, ESSs, including batteries, hydrogen storage, and thermal energy storage, are essential for managing the intermittent nature of renewable energy. Advanced technologies, such as IoT devices, fuel cells, and EVs, can further enhance the microgrid's capabilities for energy management and demand response.

Optimization techniques play a crucial role in managing microgrid operations. Mathematical models, including linear programming, mixed-integer linear programming, and nonlinear programming, are employed to solve optimization problems related to energy distribution, storage management, and demand-response strategies. Machine learning techniques, specifically DRL, can steer adaptive control strategies to manage the variability of renewable energy sources and fluctuating demand.

A comprehensive conceptual model framework is essential for understanding the interactions between these different components. This framework helps simulate various scenarios to determine the optimal configurations for energy management. Given the significant variability in microgrid

configurations, understanding the specific capabilities and interactions of these components is important for effective design.

In conclusion, the critical modeling aspects for designing a microgrid and its energy management system include integrating renewable energy sources, advanced storage technologies, sophisticated control structures, and employing mathematical and machine learning optimization techniques. The comprehensive conceptual model framework and scenario analysis further enhance the understanding and optimization of microgrid configurations. This multifaceted approach ensures efficient and reliable microgrid operations, as evidenced by the extensive research and advancements documented in the existing literature.

Q2: *How can reinforcement learning be utilized to effectively place and manage all identified modeling aspects in a microgrid while maintaining grid stability and reliability?*

The implementation of a DRL model showcases the integration of microgrids, focusing on maintaining grid stability and reliability. Dense urban districts, with their high resource demands, pose complexities in mapping system operations, revealing the limitations of DRL in efficiently handling high-density scenarios. However, the algorithm enables the creation of scenarios that mimic the implementation of a microgrid in situ, providing valuable insights into potential integrations and configurations.

The DRL agent autonomously determines the best actions to take within the energy management system, learning to maximize a reward function that reflects the optimal way of operating the microgrid. Through continuous learning and adaptation, the DRL model improves its decision-making process, effectively managing energy resources to ensure stability and reliability. This capability is of value in the dynamic urban environments where demand patterns and resource availability can be highly variable.

The developed model offers significant value by exploring various microgrid configurations, facilitating the identification of optimal setups. By simulating real-world scenarios, the model contributes to existing research by providing a practical approach to understanding and managing microgrid components and their interactions. This framework not only helps visualize potential integrations but also assists in determining the most efficient and sustainable configurations.

While DRL, particularly DQN, demonstrates suitability for flexibility and adaptation to the dynamic nature of microgrids, the quickly growing action space poses a challenge in managing the complexity and scale of potential actions. To address these limitations, a combination of GNN and Clustering is proposed for future work. This hybrid approach aims to manage the expanding action space and improve the efficiency and scalability of the optimization process.

In hybrid scenarios, multiple well-performing configurations were combined. Among these, Scenario 5.5 stands out as the most desired, combining a hybrid setup with optimized parameters to achieve the best performance. Scenario 5.5 trained with extended methods, such as hyperparameter tuning, also performed significantly better, highlighting the need for tuning of the model. The model significantly contributes to existing research by providing a comprehensive and practical tool for simulating and optimizing real-world microgrid scenarios.

Q3: *How can the location-based optimization of a microgrid be effectively visualized to ensure it creates value for all stakeholders?*

The visualization, developed and showcased using ArcGIS, and ArcGIS Experience Builder, provides a dynamic view of microgrid operations. It displays components and their states over time, offering stakeholders a clear and actionable view of various optimization strategies.

The current application, showcased internally through ArcGIS Experience Builder and hosted on AGOL, serves as a POC. This POC demonstrates the potential of using location-based visualization to create a comprehensive DT tool. The input data includes geographical information and timely data from energy simulations and optimizations, resulting in a state vector that represents the actions taken by the agent at each time step, such as battery charging or discharging and heat pumps generating energy.

Feedback from the expert panel emphasized the importance of making the DT scalable and modular to adapt to different microgrid setups and evolving energy management needs. Establishing real data connections and optimizing code for faster execution are essential next steps to enhance the DT's functionality. These improvements will enable real-time data integration and quicker adjustments to

parameters, making the DT a more efficient and reliable tool for stakeholders.

In conclusion, the POC visualization successfully demonstrates how location-based optimization of a microgrid can be effectively visualized to create value for all stakeholders. It sets a strong foundation for further development into a comprehensive DT tool that can facilitate informed decision-making and promote effective energy management practices. By making the DT accessible and interactive, stakeholders can better understand and analyze microgrid performance, ultimately enhancing grid stability and reliability.

8.3. Conclusion

Main Research Question: *How can Distributed Energy Resources be deployed and managed in a cost-effective manner within an electrified microgrid to achieve a balance between energy consumption and production in order to minimize the burden on the central grid given fluctuating demand?*

The study demonstrates that DERs can be effectively deployed and managed within an electrified microgrid through a combination of simulation techniques with geographical data input, optimization algorithms, and dynamic mappings and visualization. A DRL model, which learns from a reward function aimed at minimizing grid connection and operational costs, ensures a balance between energy consumption and production while reducing the strain on the central grid under fluctuating demand conditions.

A conceptual model provides the fundamentals and understanding for the microgrid design, necessitating consideration of various modeling aspects, as highlighted in both the theoretical framework and the existing literature. The framework facilitates the design and simulation of various scenarios to determine the optimal configurations for energy management.

The DRL model employs a DQN to explore and learn optimal control strategies, encouraging actions that minimize reliance on the central grid and reduce costs. While the DQN presented limitations due to the quickly growing action space in urban environments, implementing Graph Neural Networks within the framework can overcome these issues, enabling the balancing of energy by means of distance.

The model facilitates the simulation and optimization of real-world scenarios, aiding in the identification of optimal configurations for maintaining grid stability and reliability. Additionally, a DT visualization tool can further facilitate the implementation of microgrids by bridging the gap between theoretical models and practical applications for stakeholders. This tool offers a dynamic and actionable representation of microgrid operations. The POC visualization lays a strong foundation for further development into a comprehensive DT tool, demonstrating the feasibility and value of this approach. Feedback from expert panels highlights the importance of scalability and modularity, ensuring that the tools can adapt to diverse microgrid setups and evolving energy management needs.

This research pipeline not only offers a structured approach for simulating and optimizing microgrid scenarios but also provides practical solutions and frameworks for the efficient management and deployment of microgrids. By leveraging these integrated methodologies, the energy grid is closer to being reimagined and transformed, bringing the vision of a sustainable, resilient, and efficient energy system closer to reality.

8.4. Recommendations

The purpose of this study is to contribute to the implementation of a new form of energy grids that are more sustainable, resilient, and efficient. The key findings and conclusions highlight the development of a comprehensive approach to simulate case studies from beginning to end, exploring the implementation of microgrids or energy hubs. These energy hubs should be implemented in the long run for urban environments.

8.4.1. Implementation Guidance

For implementation, the recommendations are as follows:

- **Utilize Geo-Data-based Load Simulation and Extrapolation:** For new case studies, gather geospatial data in the same manner as in the initial study to maintain data integrity and accuracy. The new case studies can vary significantly, showcasing the diversity of possibilities.
- **Run and Train the Model for New Scenarios:** Use the model to explore various microgrid

configurations and optimizations. The comprehensive framework allows for a better understanding of the problem and refinement of energy management strategies.

- **Leverage Visualization Tools:** Utilize current visualization tools to gain a better understanding of microgrid operations and their impact on energy management. This will encourage further research and actions towards implementation. Engaging stakeholders with these visualizations can enhance decision-making processes by illustrating potential outcomes and benefits.

8.4.2. Areas for Further Research

While this study has provided a comprehensive framework for simulating and optimizing microgrid scenarios, there are several aspects that require further investigation. Addressing these areas will enhance the model's accuracy, realism, and applicability, ensuring its effectiveness in diverse and evolving urban environments.

- **Incorporating Real Data:**
  - Address the limitations of using assumptions for certain elements, such as battery cost cycles, by incorporating real-world data. This will improve the accuracy and reliability of the model.
  - Connect to updated information sources, such as the Energie Bank Nederland, to integrate real-time weather data.
- **Extending the Model:**
  - Extend simplified elements in the current model to more closely mimic reality. For example:
    - Incorporate more comprehensive battery charging and discharging models, like Figure 2.10.
    - Use wind turbine models for the wind speed and generation, like used by Alam and Jin (2023).
    - Model EVs and charging stations with greater detail to reflect actual usage patterns and technological advancements, like used by Fachrizal et al. (2020).
  - Investigate different models for EVs to enhance the simulation's realism and applicability.
  - Test multiple algorithms to compare performance.
  - Define the module action space more extensively, including the addition of an idle mode.
- **Enhancing Algorithm Efficiency:**
  - Extend the code with GNNs to address the limitations of the current DQN implementation. This will enable the inclusion of distances and routing optimization, enhancing the efficiency and scalability of the optimization process.
- **Automated Visualization and Geoprocessing:**
  - Develop an automated visualization tool directly connected with the simulation and optimization scripts to streamline the process and make it more user-friendly.
  - Create a geoprocessing tool that can be easily used by stakeholders, facilitating the practical implementation of microgrid solutions in various urban settings.
  - Connect the simulation and optimization scripts to a database, like PostgreSQL.

Additionally, it is possible to support the Pymgrid library, which is currently under development. Integrating additional modules could significantly benefit future research into microgrids. Interaction with the creator of the library has indicated a strong openness to suggestions and, for instance, an interest in the proposal to add a specific EV module that operates based on a time series of availability. Furthermore, it has become evident that expanding the action space within Pymgrid is expected to substantially increase its functionality and applicability in diverse research scenarios.

Implementing these recommendations will significantly advance the deployment and management of Distributed Energy Resources within electrified microgrids. By adopting these strategies, stakeholders can achieve a more sustainable, resilient, and efficient energy system, bringing the vision of a reimagined energy grid closer to reality.

References

AbuElrub, A., Hamed, F., & Saadeh, O. (2020). Microgrid integrated electric vehicle charging algorithm with photovoltaic generation. *Journal of Energy Storage*, *32*, 101858. https://doi.org/10.1016/j.est.2020.101858

ACP. (n.d.). Clean Energy Storage Facts. Retrieved March 8, 2024, from https://cleanpower.org/facts/clean-energy-storage/

Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A Next-generation Hyperparameter Optimization Framework. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2623–2631. https://doi.org/10.1145/3292500.3330701

Alam, F., & Jin, Y. (2023). The Utilisation of Small Wind Turbines in Built-Up Areas: Prospects and Challenges [Number: 4 Publisher: Multidisciplinary Digital Publishing Institute]. *Wind*, *3*(4), 418–438. https://doi.org/10.3390/wind3040024

Ali, Z., Saleem, K., Putrus, G., Marzband, M., & Dudley, S. (2022, January). Chapter 5 - Control of PV and EV connected smart grid. In B. Subudhi & P. K. Ray (Eds.), *Microgrid Cyberphysical Systems* (pp. 115–157). Elsevier. https://doi.org/10.1016/B978-0-323-99910-6.00001-3

Almasan, P., Suárez-Varela, J., Rusek, K., Barlet-Ros, P., & Cabellos-Aparicio, A. (2022). Deep reinforcement learning meets graph neural networks: Exploring a routing optimization use case. *Computer Communications*, *196*, 184–194. https://doi.org/10.1016/j.comcom.2022.09.029

Al-Shetwi, A. Q. (2022). Sustainable development of renewable energy integrated power sector: Trends, environmental impacts, and recent challenges. *Science of The Total Environment*, *822*, 153645. https://doi.org/10.1016/j.scitotenv.2022.153645

Arias, M. B., Kim, M., & Bae, S. (2017). Prediction of electric vehicle charging-power demand in realistic urban traffic networks. *Applied Energy*, *195*, 738–753. https://doi.org/10.1016/j.apenergy.2017.02.021

Bakker, J. (n.d.). Historische dynamische energieprijzen: Graaf in ons archief. Retrieved June 12, 2024, from https://jeroen.nl/dynamische-energie/prijzen/historisch

Bartels, E. A., Pippia, T., & De Schutter, B. (2022). Influence of hydrogen on grid investments for smart microgrids. *International Journal of Electrical Power & Energy Systems*, *141*, 107968. https://doi.org/10.1016/j.ijepes.2022.107968

Basu, A. K., Chowdhury, S. P., Chowdhury, S., & Paul, S. (2011). Microgrids: Energy management by strategic deployment of DERs—A comprehensive survey. *Renewable and Sustainable Energy Reviews*, *15*(9), 4348–4356. https://doi.org/10.1016/j.rser.2011.07.116

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., & Zaremba, W. (2016, June). OpenAI Gym [arXiv:1606.01540 [cs]]. Retrieved April 23, 2024, from http://arxiv.org/abs/1606.01540

CBS Statline. (2023a, November). Renewable energy; consumption by energy source, technology and application. Retrieved January 9, 2024, from https://opendata.cbs.nl/statline/#/CBS/en/dataset/84917ENG/table

CBS Statline. (2023b, December). Kerncijfers wijken en buurten 2022. Retrieved February 8, 2024, from https://opendata.cbs.nl/statline/#/CBS/nl/dataset/85318NED/table?ts=1707396906931

CEER. (2020, February). *2nd Report on Power Losses* (tech. rep.). Retrieved April 25, 2024, from https://www.ceer.eu/documents/104400/-/-/fd4178b4-ed00-6d06-5f4b-8b87d630b060

Chen, X., Qu, G., Tang, Y., Low, S., & Li, N. (2022). Reinforcement Learning for Selective Key Applications in Power Systems: Recent Advances and Future Challenges [arXiv:2102.01168 [cs, eess]]. *IEEE Transactions on Smart Grid*, *13*(4), 2935–2958. https://doi.org/10.1109/TSG.2022.3154718

Chen, X., Qu, G., Tang, Y., Low, S., & Li, N. (2023). Reinforcement Learning for Decision-Making and Control in Power Systems. In J. S. Tietjen, M. D. Ilic, L. Bertling Tjernberg, & N. N. Schulz (Eds.), *Women in Power: Research and Development Advances in Electric Power Systems* (pp. 265–285). Springer International Publishing. https://doi.org/10.1007/978-3-031-29724-3_10

Cheong, D. L. K., Fernando, T., Iu, H., Reynolds, M., & Fletcher, J. (2017). Review of clustering algorithms for microgrid formation. *2017 IEEE Innovative Smart Grid Technologies - Asia (ISGT-Asia)*, 1–6. https://doi.org/10.1109/ISGT-Asia.2017.8378350

EBN. (2023). Energie in cijfers 2023. Retrieved January 5, 2024, from https://www.ebn.nl/wp-content/uploads/2023/01/EBN-Infographic-2023-energie-in-cijfers-A4.pdf

Ediriweera, S., & Widanagama Arachchige, L. (2022). Design and protection of microgrid clusters: A comprehensive review. *AIMS Energy*, *10*, 375–411. https://doi.org/10.3934/energy.2022020

ESRI Nederland. (2022, March). BAG 2.0 - Basisregistratie Adressen en Gebouwen - Overview. Retrieved February 22, 2024, from https://accenturegisnl.maps.arcgis.com/home/item.html?id=5197579f7848401ab9398359d704c3e5

ESRI Nederland. (2024a, February). Energielabels - Overview. Retrieved February 22, 2024, from https://accenturegisnl.maps.arcgis.com/home/item.html?id=54b76186235c4fecafdfb74dcc2767eb

ESRI Nederland. (2024b, February). Woningtypering - Overview. Retrieved February 22, 2024, from https://www.arcgis.com/home/item.html?id=fa01ef63321e482e9b2c55620e554ffc

European Commission. (2023). Renewable energy directive. Retrieved January 5, 2024, from https://energy.ec.europa.eu/topics/renewable-energy/renewable-energy-directive-targets-and-rules/renewable-energy-directive_en

European Council. (2022, December). REPowerEU: Council agrees on accelerated permitting rules for renewables. Retrieved January 8, 2024, from https://www.consilium.europa.eu/en/press/press-releases/2022/12/19/repowereu-council-agrees-on-accelerated-permitting-rules-for-renewables/

European Council. (2023, December). Fit for 55. Retrieved January 11, 2024, from https://www.consilium.europa.eu/en/policies/green-deal/fit-for-55-the-eu-plan-for-a-green-transition/

Fachrizal, R., Shepero, M., van der Meer, D., Munkhammar, J., & Widén, J. (2020). Smart charging of electric vehicles considering photovoltaic power production and electricity consumption: A review. *eTransportation*, *4*, 100056. https://doi.org/10.1016/j.etran.2020.100056

Farzaneh, H. (2019). *Energy Systems Modeling: Principles and Applications*. Springer. https://doi.org/10.1007/978-981-13-6221-7

Gao, K., Wang, T., Han, C., Xie, J., Ma, Y., & Peng, R. (2021). A Review of Optimization of Microgrid Operation. *Energies*, *14*, 2842. https://doi.org/10.3390/en14102842

Gasunie. (2024). Waterstof door gasleiding: Veilig en duurzaam. Retrieved April 22, 2024, from https://www.gasunie.nl/expertise/waterstof/waterstof-door-gasleiding-veilig-en-duurzaam

Gong, X., Dong, F., Mohamed, M., Abdalla, O., & Ali, Z. (2020). A Secured Energy Management Architecture for Smart Hybrid Microgrids Considering PEM-Fuel Cell and Electric Vehicles. *IEEE Access*, *PP*, 1–1. https://doi.org/10.1109/ACCESS.2020.2978789

Hannan, M. A., Mollik, M. S., Al-Shetwi, A. Q., Rahman, S. A., Mansor, M., Begum, R. A., Muttaqi, K. M., & Dong, Z. Y. (2022). Vehicle to grid connected technologies and charging strategies: Operation, control, issues and recommendations. *Journal of Cleaner Production*, *339*, 130587. https://doi.org/10.1016/j.jclepro.2022.130587

Henri, G., Levent, T., Halev, A., Alami, R., & Cordier, P. (2020, November). Pymgrid: An Open-Source Python Microgrid Simulator for Applied Artificial Intelligence Research [arXiv:2011.08004 [cs]]. https://doi.org/10.48550/arXiv.2011.08004

Herbst, A., Toro, F., & Reitze, F. (2012). Introduction to Energy Systems Modelling. *Swiss J Econ Stat*, *148*, 111–135. https://doi.org/10.1007/BF03399363

Hodges, A., Hoang, A. L., Tsekouras, G., Wagner, K., Lee, C.-Y., Swiegers, G. F., & Wallace, G. G. (2022). A high-performance capillary-fed electrolysis cell promises more cost-competitive renewable hydrogen [Number: 1 Publisher: Nature Publishing Group]. *Nature Communications*, *13*(1), 1304. https://doi.org/10.1038/s41467-022-28953-x

Hossain, R., Gautam, M., Thapa, J., Livani, H., & Benidris, M. (2023). Deep reinforcement learning assisted co-optimization of Volt-VAR grid service in distribution networks. *Sustainable Energy, Grids and Networks*, *35*, 101086. https://doi.org/10.1016/j.segan.2023.101086

IEA. (2023, July). Tracking Clean Energy Progress (TCEP). Retrieved February 1, 2024, from https://www.iea.org/energy-system/low-emission-fuels/electrolysers

Ifaei, P., Esfehankalateh, A. T., Ghobadi, F., Mohammadi-Ivatloo, B., & Yoo, C. (2023). Systematic review and cutting-edge applications of prominent heuristic optimizers in sustainable energies. *Journal of Cleaner Production*, *414*, 137632. https://doi.org/10.1016/j.jclepro.2023.137632

International Energy Agency. (2023). Electricity Market Report_update 2023. Retrieved January 8, 2024, from https://iea.blob.core.windows.net/assets/15172a8d-a515-42d7-88a4-edc27c3696d3/ElectricityMarketReport_Update2023.pdf

Israr, A., & Yang, Q. (2021, January). Chapter 9 - Resilient and sustainable microgeneration power supply for 5G mobile networks. In Q. Yang, T. Yang, & W. Li (Eds.), *Renewable Energy Microgeneration Systems* (pp. 213–228). Academic Press. https://doi.org/10.1016/B978-0-12-821726-9.00009-6

Kallenberg, L. (2016). MARKOV DECISION PROCESSES.

Kim, M., Kim, J.-S., Choi, M.-S., & Park, J.-H. (2022). Adaptive Discount Factor for Deep Reinforcement Learning in Continuing Tasks with Uncertainty [Number: 19 Publisher: Multidisciplinary Digital Publishing Institute]. *Sensors*, *22*(19), 7266. https://doi.org/10.3390/s22197266

Klemm, C., & Wiese, F. (2022). Indicators for the optimization of sustainable urban energy systems based on energy system modeling. *Energy, Sustainability and Society*, *12*. https://doi.org/10.1186/s13705-021-00323-3

Ko, W., & Kim, J. (2019). Generation Expansion Planning Model for Integrated Energy System Considering Feasible Operation Region and Generation Efficiency of Combined Heat and Power. *Energies*, *12*, 226. https://doi.org/10.3390/en12020226

LAN. (2024). Probleemanalyse Congestie in het laagspanningsnet.

Lasantha Meegahapola. (2023, March). Grid Integration of Hydrogen Electrolyzers and Fuel-Cells: Opportunities, Challenges and Future Directions - IEEE Smart Grid. Retrieved February 8, 2024, from https://smartgrid.ieee.org/bulletins/march-2023-1/grid-integration-of-hydrogen-electrolyzers-and-fuel-cells-opportunities-challenges-and-future-directions

Li, S., Oshnoei, A., Blaabjerg, F., & Anvari-Moghaddam, A. (2023). Hierarchical Control for Microgrids: A Survey on Classical and Machine Learning-Based Methods. *Sustainability*, *15*(11), 8952. https://doi.org/10.3390/su15118952

Mwasilu, F., Justo, J. J., Kim, E.-K., Do, T. D., & Jung, J.-W. (2014). Electric vehicles and smart grid interaction: A review on vehicle to grid and renewable energy sources integration. *Renewable and Sustainable Energy Reviews*, *34*, 501–516. https://doi.org/10.1016/j.rser.2014.03.031

Nakabi, T. A., & Toivanen, P. (2021). Deep reinforcement learning for energy management in a microgrid with flexible demand. *Sustainable Energy, Grids and Networks*, *25*, 100413. https://doi.org/10.1016/j.segan.2020.100413

NASEO. (2023). Microgrids State Working Group | NASEO. Retrieved January 12, 2024, from https://www.naseo.org/issues/electricity/microgrids

Nationaal Energie Dashboard. (2024). Zonne-energie | Nationaal Energie Dashboard. Retrieved April 21, 2024, from https://ned.nl/index.php/nl/dataportaal/energie-productie/elektriciteit/zonne-energie

Nationaal georegister. (2023, November). Verrijkte BAG ter ondersteuning van lokale energetische vraagstukken. Retrieved February 27, 2024, from https://nationaalgeoregister.nl/geonetwork/srv/dut/catalog.search#/metadata/ffba6e32-3268-46b0-aade-070fcde08583

Netbeheer Nederland. (2023). Het energiesysteem van de toekomst: De II3050-scenario's.

Netbeheer Nederland. (2024, January). Capaciteitskaart invoeding elektriciteitsnet. Retrieved January 17, 2024, from https://capaciteitskaart.netbeheernederland.nl

Netbeheer Nederland, ACM, Rijksoverheid, IPO, Provincie Limburg, Provincie Noordbrabant, VNG, NP RES, VNO-NCW, VEMW, CES-cluster Rotterdam-Moerdijk, CES-cluster Chemelot, CES-cluster 6, Energie Nederland, & NVDE. (2022, December). Landelijk Actieprogramma Netcongestie. Retrieved March 5, 2024, from https://open.overheid.nl/documenten/ronl-4a4a6f1bcb4f30278f4205aeb085c3208f62e8a6/pdf

Nl, A., & Blezer, I. (n.d.). Handleiding voor gemeenten.

Pang, K., Zhou, J., Tsianikas, S., Coit, D. W., & Ma, Y. (2024). Long-term microgrid expansion planning with resilience and environmental benefits using deep reinforcement learning. *Renewable and Sustainable Energy Reviews*, *191*, 114068. https://doi.org/10.1016/j.rser.2023.114068

Parisio, A., Rikos, E., & Glielmo, L. (2014). A Model Predictive Control Approach to Microgrid Operation Optimization [Conference Name: IEEE Transactions on Control Systems Technology]. *IEEE Transactions on Control Systems Technology*, *22*(5), 1813–1827. https://doi.org/10.1109/TCST.2013.2295737

PDOK. (n.d.). PDOK Viewer. Retrieved February 7, 2024, from https://app.pdok.nl/viewer/#x=92832.91&y=435673.02&z=11.3553&background=BRT-A%20standaard&layers=ecd65d6f-0249-4c60-85b9-5fd87fbe60c8;buurten

PDOK. (2024, February). BAG. Retrieved February 13, 2024, from https://service.pdok.nl/lv/bag/atom/bag.xml

Reuters. (2023, December). Dutch power grid operator TenneT warns of shortages by 2030 | Headlines. Retrieved January 16, 2024, from https://www.devdiscourse.com/article/headlines/2317232-dutch-power-grid-operator-tennet-warns-of-shortages-by-2030

Richardson, D. B. (2013). Electric vehicles and the electric grid: A review of modeling approaches, Impacts, and renewable energy integration. *Renewable and Sustainable Energy Reviews*, *19*, 247–254. https://doi.org/10.1016/j.rser.2012.11.042

Rijnstate. (2023). Duurzaamheid in Rijnstate Elst. Retrieved February 5, 2024, from https://www.rijnstate.nl/elst/duurzaamheid

Rostami, Z., Ravadanegh, S. N., Kalantari, N. T., Guerrero, J. M., & Vasquez, J. C. (2020). Dynamic Modeling of Multiple Microgrid Clusters Using Regional Demand Response Programs [Number: 16 Publisher: Multidisciplinary Digital Publishing Institute]. *Energies*, *13*(16), 4050. https://doi.org/10.3390/en13164050

Royal Netherlands Meteorological Institute. (2024a, February). Sunshine and radiation - sunshine and radiation at a 10 minute interval - KNMI Data Platform. Retrieved February 12, 2024, from https://dataplatform.knmi.nl/dataset/zonneschijnduur-en-straling-1-0

Royal Netherlands Meteorological Institute. (2024b, February). Wind - windspeed, direction, standard-deviation at a 10 minute interval - KNMI Data Platform. Retrieved February 12, 2024, from https://dataplatform.knmi.nl/dataset/windgegevens-1-0

Shahzad, S., Abbasi, M., Ali, H., Iqbal, M., Munir, R., & Kiliç, H. (2023). Possibilities, Challenges, and Future Opportunities of Microgrids: A Review. *Sustainability*, *15*, 6366. https://doi.org/10.3390/su15086366

Shepero, M., & Munkhammar, J. (2018). Spatial Markov chain model for electric vehicle charging in cities using geographical information system (GIS) data. *Applied Energy*, *231*, 1089–1099. https://doi.org/10.1016/j.apenergy.2018.09.175

Singh, K. V., Bansal, H. O., & Singh, D. (2019). A comprehensive review on hybrid electric vehicles: Architectures and components. *Journal of Modern Transportation*, *27*(2), 77–107. https://doi.org/10.1007/s40534-019-0184-3

Sinha, R. R., & Kanwar, N. (2024, January). Chapter Four - Hybrid microgrids: Architecture, modeling, limitations, and solutions. In R. C. Bansal, J. J. Justo, & F. A. Mwasilu (Eds.), *Modeling and Control Dynamics in Microgrid Systems with Renewable Energy Resources* (pp. 65–82). Academic Press. https://doi.org/10.1016/B978-0-323-90989-1.00012-9

SolarPower Europe. (n.d.). Annual rooftop and utility scale installations in the EU - SolarPower Europe. Retrieved May 22, 2024, from https://www.solarpowereurope.org/advocacy/solar-saves/fact-figures/annual-rooftop-and-utility-scale-installations-in-the-eu

Spectral. (2018). Smart Integrated Decentralised Energy (SIDE) Systems. Retrieved April 21, 2024, from https://www.metabolic.nl/publication/new-strategies-for-smart-integrated-decentralised-energy-systems/

Stephan Brandligt. (2024). Probleemanalyse Congestie in het laagspanningsnet.

Stultiens, E. (2023, May). Primeur Nederweert: Bedrijvenpark lost netcongestie op met energy hub. Retrieved March 18, 2024, from https://solarmagazine.nl/nieuws-zonne-energie/i34217/primeur-nederweert-bedrijvenpark-lost-netcongestie-op-met-energy-hub

Subramanya, R., Sierla, S., & Vyatkin, V. (2022). Exploiting Battery Storages With Reinforcement Learning: A Review for Energy Professionals. *IEEE Access*, *10*, 1–1. https://doi.org/10.1109/ACCESS.2022.3176446

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction* (Second edition). The MIT Press.

Tajjour, S., & Singh Chandel, S. (2023). A comprehensive review on sustainable energy management systems for optimal operation of future-generation of solar microgrids. *Sustainable Energy Technologies and Assessments*, *58*, 103377. https://doi.org/10.1016/j.seta.2023.103377

TNO. (n.d.). Mapviewer | Thermogis. Retrieved April 27, 2024, from https://www.thermogis.nl/mapviewer

Trivedi, R., & Khadem, S. (2022). Implementation of artificial intelligence techniques in microgrid control environment: Current progress and future scopes. *Energy and AI*, *8*, 100147. https://doi.org/10.1016/j.egyai.2022.100147

tudelft3d & 3DGI. (2023, August). 3DBAG Viewer. Retrieved February 7, 2024, from https://3dbag.nl/en/viewer?rdx=93094.69410911822&rdy=435478.97545096674&ox=240.76812798532774&oy=623.8103628977599&oz=-595.4653418483795

Uddin, M., Mo, H., Dong, D., Elsawah, S., Zhu, J., & Guerrero, J. M. (2023). Microgrids: A review, outstanding issues and future trends. *Energy Strategy Reviews*, *49*, 101127. https://doi.org/10.1016/j.esr.2023.101127

Van, L. P., Chi, K. D., & Duc, T. N. (2023). Review of hydrogen technologies based microgrid: Energy management systems, challenges and future recommendations. *International Journal of Hydrogen Energy*, *48*(38), 14127–14148. https://doi.org/10.1016/j.ijhydene.2022.12.345

Wang, Y., Huang, S., & Infield, D. (2014). Investigation of the potential for electric vehicles to support the domestic peak load. *2014 IEEE International Electric Vehicle Conference (IEVC)*, 1–8. https://doi.org/10.1109/IEVC.2014.7056124

Waterstaat, M. v. I. e. (2020, May). Home - KNMI Dataplatform [Last Modified: 2024-02-05T16:41 Publisher: Koninklijk Nederlands Meteorologisch Instituut]. Retrieved February 12, 2024, from https://www.knmidata.nl/

Wu, P., Partridge, J., & Bucknall, R. (2020). Cost-effective reinforcement learning energy management for plug-in hybrid fuel cell and battery ships. *Applied Energy*, *275*, 115258. https://doi.org/10.1016/j.apenergy.2020.115258

Yamashita, D. Y., Vechiu, I., & Gaubert, J.-P. (2020). A review of hierarchical control for building microgrids. *Renewable and Sustainable Energy Reviews*, *118*, 109523. https://doi.org/10.1016/j.rser.2019.109523

Yu, S., Fan, Y., Shi, Z., Li, J., Zhao, X., Zhang, T., & Chang, Z. (2023). Hydrogen-based combined heat and power systems: A review of technologies and challenges. *International Journal of Hydrogen Energy*, *48*(89), 34906–34929. https://doi.org/10.1016/j.ijhydene.2023.05.187

Zarate-Perez, E., Rosales-Asensio, E., González-Martínez, A., de Simón-Martín, M., & Colmenar-Santos, A. (2022). Battery energy storage performance in microgrids: A scientific mapping perspective. *Energy Reports*, *8*, 259–268. https://doi.org/10.1016/j.egyr.2022.06.116

Zia, M. F., Elbouchikhi, E., & Benbouzid, M. (2018). Microgrids energy management systems: A critical review on methods, solutions, and prospects. *Applied Energy*, *222*, 1033–1055. https://doi.org/10.1016/j.apenergy.2018.04.103

Zonatlas. (n.d.). Data producten. Retrieved June 7, 2024, from https://www.zonatlas.nl/start/toolkits/data-producten/

A

Renewable Energy Sources by the ETP

The following overview is retrieved from the ETP Renewable catalog (ACP, n.d.).

To determine the suitability of the energy sources, described in the Table A.1, for a microgrid in an urban environment, several factors need consideration: scalability, infrastructure compatibility, environmental impact, regulatory constraints, and economic viability.

Considering these factors, options like hydrogen blending in existing natural gas networks, floating solar PV, hydrogen boilers, and hybrid heat pumps appear more immediately feasible for urban microgrids.

Table A.1: Renewable energy sources by the ETP

Name	Phase	Description	Supply Chain	Key Countries
Hydrogen blending in natural gas network	7	Hydrogen blending allows for up to 20% hydrogen by volume in natural gas streams using existing infrastructure with minimal modifications, but blends over 20% require significant upgrades and investments in downstream and end-use equipment, with the possibility of higher thresholds through continued research and development.	H2 transport	China, United Kingdom, United States, Denmark, Italy, Australia, Netherlands, Germany, Portugal
Liquefied hydrogen tanker	7	A liquefied hydrogen tanker, designed to transport LH2, uses advanced thermal insulation to handle hydrogen's low boiling point and recycles boil-off gas as fuel to enhance efficiency and reduce emissions.	H2 transport	Japan, Korea, France, Netherlands
Repurposed natural gas pipelines	8	Repurposing natural gas pipelines for hydrogen transport involves purging, replacing critical components, and regular integrity checks to manage hydrogen embrittlement. Solutions include protective coatings and pressure management, tailored to each pipeline's condition and operational needs.	H2 transport	Netherlands
Floating solar PV	8	Floating PV systems are mounted on a structure that floats on a water surface and can be associated with existing grid connections, for instance in the case of dam vicinity.	Power generation	China, France, India, Japan, Korea, Netherlands, Singapore, United Kingdom, United States
Hydrogen boiler	9	Boilers using hydrogen as a fuel instead of natural gas.	Heat generation, H2 use	United Kingdom, Italy, Netherlands, United States, Japan
Hybrid heat pump	9	Combined electric heat pump with an electrical resistance heater or a gas condensing boiler as back-up systems.	Heat generation	Netherlands, United Kingdom, Canada, United States, Japan
Aquifer thermal energy storage (ATES)	9	An open-loop geothermal network uses an aquifer to store cold or heat on a seasonal basis.	Heat storage	Netherlands

Name	Phase	Description	Supply Chain	Key Countries
Partial capture (steam reforming)	9	Steam methane reforming converts methane and steam into hydrogen and carbon monoxide at high temperatures, followed by further conversion to increase hydrogen yield and capture 60% of the resulting CO2. The technology is prevalent in ammonia and urea production, with the remaining CO2 released during heating with natural gas.	H2 production	Canada, Netherlands, Saudi Arabia, United Kingdom, United States
New hydrogen pipelines	9	Inland hydrogen pipeline construction is regulated by ASME B31.12, with existing pipelines generally smaller and using lower steel grades compared to potential new pipelines that may be larger and use higher steel grades for durability. Future pipelines are also expected to withstand cyclic loading.	H2 transport	Belgium, China, Denmark, Germany, France, Netherlands, Spain, United States
Tidal range	9	Tidal range adopts conventional hydropower principles to harvest energy from the difference in sea level between high and low tides.	Power generation	Canada, China, France, United Kingdom, Netherlands, Russia, Korea
Tidal stream-Ocean current	9	Tidal stream turbines, which convert ocean current energy similar to wind turbines, are advancing towards commercial use with various designs under full-scale sea testing. These include horizontal-axis, vertical-axis, and other innovative models like oscillating hydrofoils and tidal kites. Currently achieving a Technology Readiness Level of 6 to 8, these devices range from 0.1 to 2 MW in capacity, with expectations of commercialization and capacity increases within the next decade.	Power generation	Canada, China, France, Japan, Mexico, Netherlands, Norway, Korea, Sweden, United Kingdom, United States
Seabed fixed offshore wind turbine	9	Seabed-fixed offshore wind turbines, which are similar in technology to onshore turbines but adapted for marine conditions, constitute the majority of existing offshore wind capacity. They feature various foundation types such as monopiles and gravity foundations, supported by structures like tubular towers and jackets to optimize offshore performance.	Power generation	Belgium, China, Chinese Taipei, Denmark, Germany, Greece, France, India, Ireland, Japan, Korea, Norway, Netherlands, Portugal, Spain, Sweden, United Kingdom, United States

Name	Phase	Description	Supply Chain	Key Countries
Flywheel	9	Flywheels store energy as rotating inertia, converting kinetic energy to electricity with high efficiencies of 90-95% through the use of vacuum environments and magnetically levitated bearings. Primarily used for short-term applications like grid frequency regulation due to their low energy output, flywheels offer fast response times, low maintenance, and high cycle life, capable of both up and down regulation but not simultaneously.	Power storage	Germany, Netherlands, Ireland
Heat exchanger	10	Advanced heat exchangers for building-level substations.	Heat transport	United States, United Kingdom, Italy, Norway, Canada, France, Finland, Sweden, Germany, Netherlands
Compressed air energy storage	9-10	Compressed Air Energy Storage (CAES) compresses air into storage spaces like underground caverns or above-ground containers, converting and storing electricity as thermal and mechanical energy. This energy is released by expanding the hot compressed air through a turbine to generate electricity. Traditional CAES systems dissipate heat during compression, requiring external heating for expansion. Adiabatic CAES, under development, aims to retain and reuse this heat to improve efficiency and eliminate the need for external heating sources.	Power storage	Australia, Canada, China, Germany, Netherlands, United States
Salt cavern storage	9-10	Salt caverns are artificial underground cavities ideal for hydrogen storage, offering low cushion gas requirements and high sealing integrity. Their use is limited geographically and involves ongoing research on reusing caverns while addressing contamination risks.	H2 storage	United Kingdom, United States, Netherlands, Germany

Table A.2: Suitability of energy sources from filtered options ETP

Energy Source	Scalability	Infrastructure Compatibility	Environmental Impact	Regulatory Constraints	Economic Viability
Hydrogen Blending in Natural Gas Network	x	x	x	x	x
Liquefied Hydrogen Tanker					
Repurposed Natural Gas Pipelines		x		x	
Floating Solar PV	x		x		x
Hydrogen Boiler		x			x
Hybrid Heat Pump	x	x			x
New Hydrogen Pipelines	x	x		x	x
Tidal Range	x		x		x
Tidal Stream-Ocean Current	x		x		x
Seabed Fixed Offshore Wind Turbine	x		x		x
Flywheel	x				x
Heat Exchanger	x	x			x
Compressed Air Energy Storage	x		x		x
Partial capture (steam reforming)	x		x		x
Salt cavern storage	x		x		x

B

Simulation categories

For the simulation of the energy profiles, categories for the buildings must be made. For every category a belonging building gets a simulated profile which then is extrapolated to all the other buildings in this specific category. The categories are made up out of factors making entities in the set distinct from each other in terms of energy profiles. The factors that influence the profile are among more the building type, the building function, the energy label, the amount of residents (or the amount of square meters), the presence of solar panels, and the presence of connection to district heating.

Data that is converted and geoprocessed in order to present what we need is:

- ESRI Nederland (2024b) offer a dataset that classifies residential units or 'vbo's'.
- ESRI Nederland (2022) offers all the Building information: to decide which version we need https://www.kadaster.nl/-/beslisboom-bag-2.0-producten
- ESRI Nederland (2024a) Energie labels
- Nationaal georegister (2023)

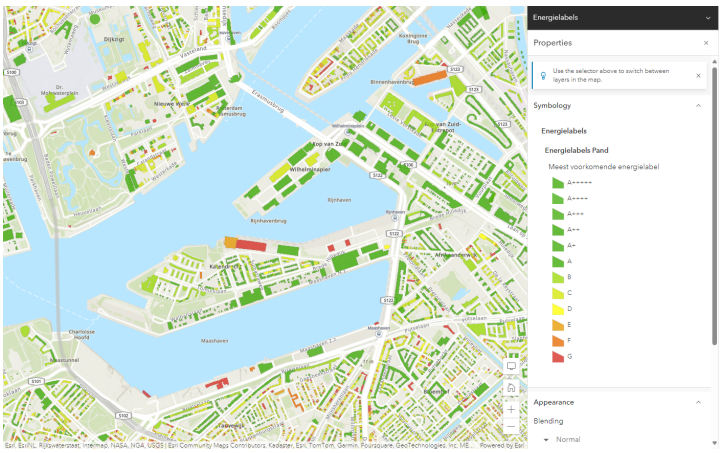


Figure B.2: Energy labels per parcel by ESRI (ESRI Nederland, 2024a)

The Housing Typology dataset contains only buildings that contain at least one residential unit with the use purpose "residential function", hereinafter referred to as "dwellings" (ESRI Nederland, 2024b). These buildings have been divided into the following categories through analysis:

- Apartment. This is a dwelling to which multiple residential units are related, regardless of the use purpose of these residential units. However, as for all buildings in this dataset, at least one of these residential units must have the use purpose 'residential function'.
- Detached house. This is a dwelling that is not connected to another building with a residential unit.

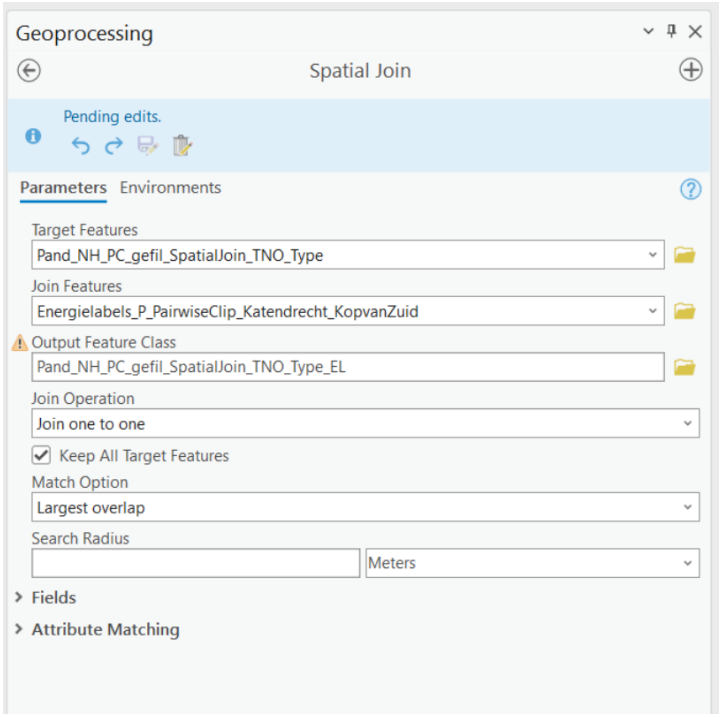


Figure B.1: Geoprocessing data - Spatial join in Arcgis

- Terraced or linked house. This dwelling is connected to multiple buildings with a residential unit. This category also includes linked houses, which are houses connected through their garages instead of their main building.
- Corner house. This is the first or last dwelling in a series of buildings. This dwelling is itself connected to a single building with a residential unit, but this other building is connected to several other buildings.
- Semi-detached house. This dwelling is connected to a single building with a residential unit, and this other building is only connected to the aforementioned dwelling.

Functions grouped by energy profile similarity (ESRI Nederland, 2022):

- Residential and Accommodation Functions:
  - Residential Function: Includes all residential buildings primarily used for living, where energy consumption is characterized by heating, cooling, lighting, and household appliances.
  - Lodging Function: Includes hotels and hostels, which have similar energy needs to residential functions, mainly for heating, cooling, and lighting of rooms.
- Commercial and Institutional Functions:
  - Retail Function: Buildings with retail activities that use energy for lighting, heating, cooling, and sometimes specific cooling needs (for example, for refrigerated storage).
  - Office Function: Offices have a constant energy use, mainly for lighting, computers, and climate control.
  - Assembly Function: Includes churches, conference centers, and other meeting spaces, where energy use can peak during events.
  - Educational Function and Healthcare Function: Both have specific requirements for lighting, heating, cooling, and specialized equipment, resulting in a similar energy profile.
- Industrial and Special Functions:
  - Industrial Function: Characterized by high energy consumption for production processes, machinery, and sometimes 24-hour operations.

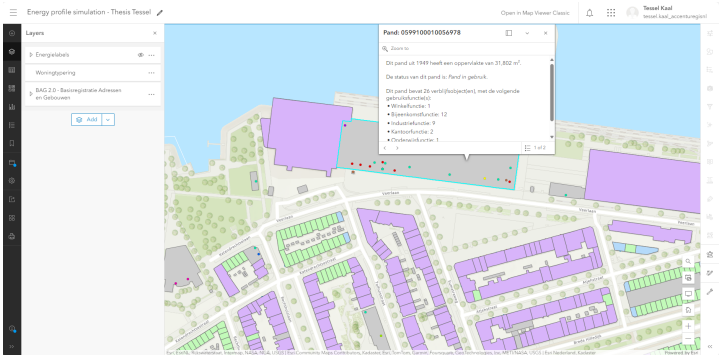


Figure B.4: Non residential buildings defined in BAG2.0

- Sports Function: Large spaces that need to be heated or cooled, in addition to energy use for lighting and sports equipment.
- Cell Function: Can be similar to commercial and institutional functions in terms of lighting and climate control, but with specific security requirements.

```
IF (bag_gebr_woonfunctie OR bag_gebr_logiesfunctie OR tno_n_f1woon OR tno_n_f7logies) > 0:
  Functiongroup = "Woon- en Verblijfsfuncties"

IF (bag_gebr_bijeenkomstfunctie OR bag_gebr_kantoorfunctie OR bag_gebr_onderwijsfunctie OR bag_gebr_gezondheidszorg OR bag_gebr_winkelfunctie OR tno_n_f2kantoor OR tno_n_f3bijeenkomst OR tno_n_f4onderwijs OR tno_n_f5winkel OR tno_n_f8gezondheidszorg OR tno_n_f12bedrpand_zvbo_industri OR tno_n_f13bedrpand_zvbo_groen) > 0:
  Functiongroup = "CommerciÃńle en Institutionele Functies"

IF (bag_gebr_celfunctie OR bag_gebr_industriefunctie OR bag_gebr_sportfunctiewoonfunctie OR logiesfunctie OR tno_n_f6sport
OR tno_n_f9industrie OR tno_n_f10cel) > 0:
  Functiongroup = "IndustriÃńle en Speciale Functies"
```

Listing B.1: Categories

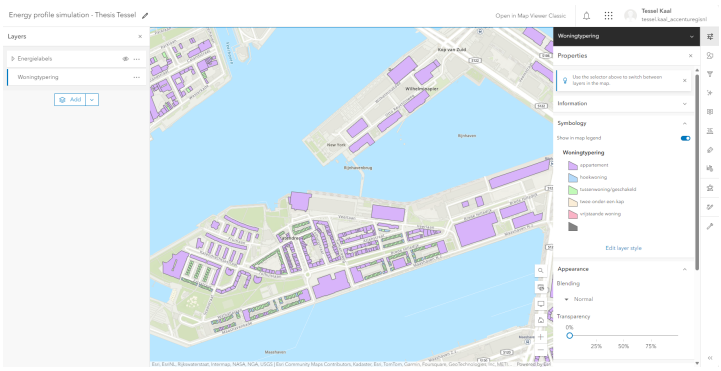


Figure B.3: Buildings types by ESRI (ESRI Nederland, 2024b)

C

Data Cleaning

The data used as the spatial dataset is derived by combining six distinct datasets, each contributing essential information to the analysis. These datasets include the Buildings dataset (BAG) (ESRI Nederland, 2022), energy labels and building types datasets (ESRI Nederland, 2024a, 2024b) provided by ESRI, an augmented dataset by Nationaal georegister (2023), and datasets providing postal code and neighborhood boundaries alongside demographic data (CBS Statline, 2023b).

The BAG dataset contains a complete overview of all buildings, residential and commercial, for the two neighborhoods selected for the case studies. It serves as the foundation of the spatial analysis and contains information such as building area and function. Energy labels and building types datasets (ESRI Nederland, 2024a, 2024b) provide additional details essential for the analysis. Energy labels are assigned to every building in the dataset, while building types are specifically categorized for residential structures. An enriched dataset by Nationaal georegister (2023) focuses on commercial buildings, augmenting understanding of energy-related metrics within this subset.

The variable listings of all these datasets are presented in figures C.1 to C.4 and C.6.

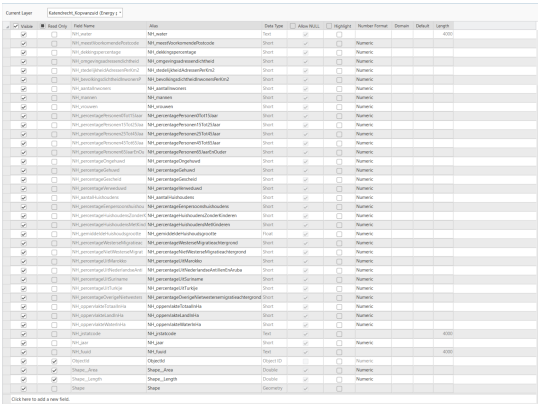


Figure C.1: Variable list: Dataset Neighborhoods (CBS Statline, 2023b)

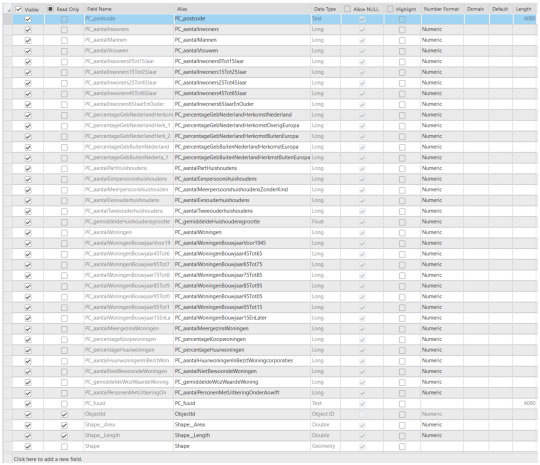


Figure C.2: Variable list: Dataset Postal Codes (CBS Statline, 2023b)

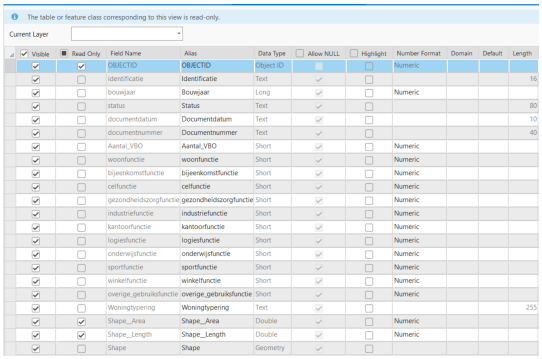


Figure C.3: Variable list: Dataset Building Types (ESRI Nederland, 2024b)

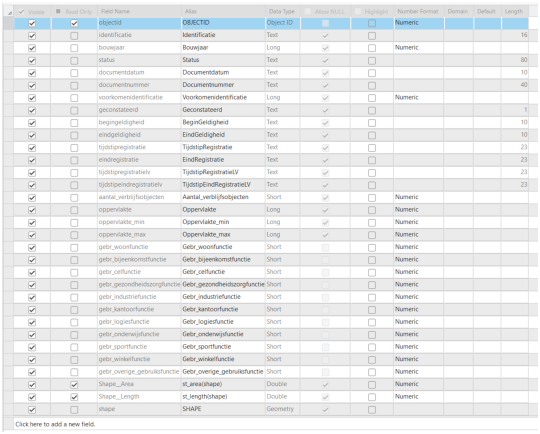
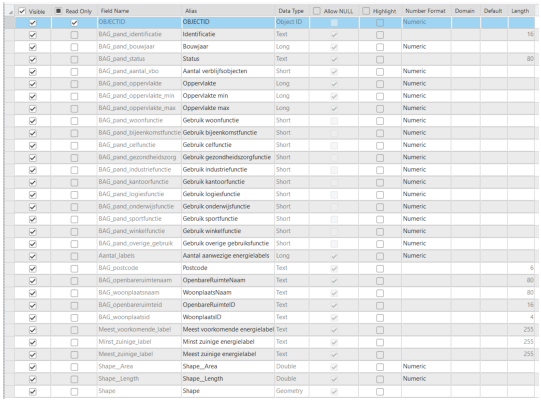
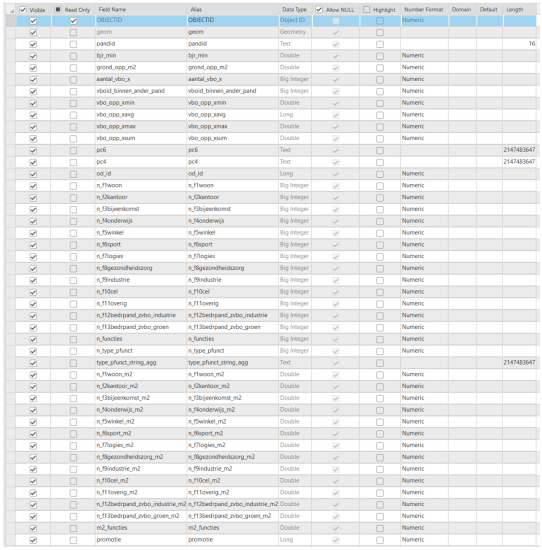


Figure C.4: Variable list: Dataset BAG (ESRI Nederland, 2022)



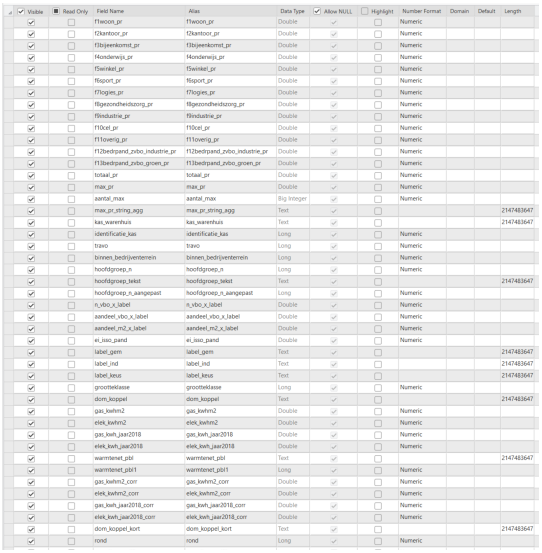
Visible	Read Only	Field Name	Alias	Data Type	Allow NULL	Highlight	Number Format	Domain	Default	Length
✓	✓	OBJECTID	OBJECTID	Object ID			Numeric			
✓		BAG_pand_identificatie	Identificatie	Text	✓					16
✓		BAG_pand_bouwjaar	Bouwjaar	Long	✓		Numeric			
✓		BAG_pand_status	Status	Text	✓					80
✓		BAG_pand_aantal_vbo	Aantal verblijfsobjecten	Short	✓		Numeric			
✓		BAG_pand_oppervlakte	Oppervlakte	Long	✓		Numeric			
✓		BAG_pand_oppervlakte_min	Oppervlakte min	Long	✓		Numeric			
✓		BAG_pand_oppervlakte_max	Oppervlakte max	Long	✓		Numeric			
✓		BAG_pand_woonfunctie	Gebruik woonfunctie	Short			Numeric			
✓		BAG_pand_bijeenkomstfunctie	Gebruik bijeenkomstfunctie	Short			Numeric			
✓		BAG_pand_celfunctie	Gebruik celfunctie	Short			Numeric			
✓		BAG_pand_gezondheidszorg	Gebruik gezondheidszorgfunctie	Short			Numeric			
✓		BAG_pand_industriefunctie	Gebruik industriefunctie	Short			Numeric			
✓		BAG_pand_kantoorfunctie	Gebruik kantoorfunctie	Short			Numeric			
✓		BAG_pand_logiesfunctie	Gebruik logiesfunctie	Short			Numeric			
✓		BAG_pand_onderwijsfunctie	Gebruik onderwijsfunctie	Short			Numeric			
✓		BAG_pand_sportfunctie	Gebruik sportfunctie	Short			Numeric			
✓		BAG_pand_winkelfunctie	Gebruik winkelfunctie	Short			Numeric			
✓		BAG_pand_overige_gebruik	Gebruik overige gebruikfunctie	Short			Numeric			
✓		Aantal_labels	Aantal aanwezige energielabels	Long	✓		Numeric			
✓		BAG_postcode	Postcode	Text	✓					6
✓		BAG_openbareruimtenaam	OpenbareRuimteNaam	Text	✓					80
✓		BAG_woonplaatsnaam	WoonplaatsNaam	Text	✓					80
✓		BAG_openbareruimteid	OpenbareRuimteID	Text	✓					16
✓		BAG_woonplaatsid	WoonplaatsID	Text	✓					4
✓		Meest_voorkomende_label	Meest voorkomende energielabel	Text	✓					255
✓		Minst_zuinige_label	Minst zuinige energielabel	Text	✓					255
✓		Meest_zuinige_label	Meest zuinige energielabel	Text	✓					255
✓		Shape_Area	Shape_Area	Double	✓		Numeric			
✓		Shape_Length	Shape_Length	Double	✓		Numeric			
✓		Shape	Shape	Geometry	✓					

Figure C.5: Variable list: Dataset Energy Labels (ESRI Nederland, 2024a)



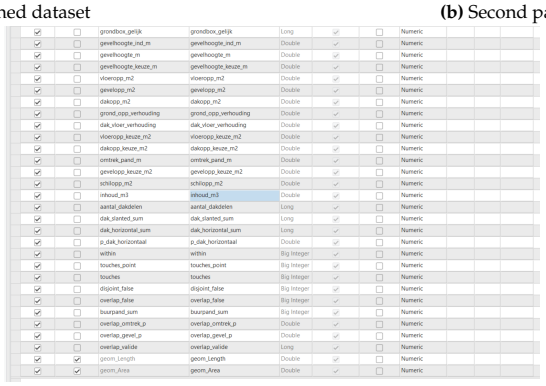
Visible	Read Only	Field Name	Alias	Data Type	Allow NULL	Highlight	Number Format	Domain	Default	Length
✓	✓	OBJECTID	OBJECTID	Object ID			Numeric			
✓		geom	geom	Geometry	✓					
✓		pandid	pandid	Text	✓					16
✓		bjr_min	bjr_min	Double	✓		Numeric			
✓		grond_opp_m2	grond_opp_m2	Double	✓		Numeric			
✓		aantal_vbo_x	aantal_vbo_x	Big Integer	✓		Numeric			
✓		vboid_binnen_ander_pand	vboid_binnen_ander_pand	Big Integer	✓		Numeric			
✓		vbo_opp_xmin	vbo_opp_xmin	Double	✓		Numeric			
✓		vbo_opp_xavg	vbo_opp_xavg	Long	✓		Numeric			
✓		vbo_opp_xmax	vbo_opp_xmax	Double	✓		Numeric			
✓		vbo_opp_xsum	vbo_opp_xsum	Double	✓		Numeric			
✓		pc6	pc6	Text	✓					2147483647
✓		pc4	pc4	Text	✓					2147483647
✓		oid_id	oid_id	Long	✓		Numeric			
✓		n_f1woon	n_f1woon	Big Integer	✓		Numeric			
✓		n_f2kantoor	n_f2kantoor	Big Integer	✓		Numeric			
✓		n_f3bijeenkomst	n_f3bijeenkomst	Big Integer	✓		Numeric			
✓		n_f4onderwijs	n_f4onderwijs	Big Integer	✓		Numeric			
✓		n_f5winkel	n_f5winkel	Big Integer	✓		Numeric			
✓		n_f6sport	n_f6sport	Big Integer	✓		Numeric			
✓		n_f7logies	n_f7logies	Big Integer	✓		Numeric			
✓		n_f8gezondheidszorg	n_f8gezondheidszorg	Big Integer	✓		Numeric			
✓		n_f9industrie	n_f9industrie	Big Integer	✓		Numeric			
✓		n_f10cel	n_f10cel	Big Integer	✓		Numeric			
✓		n_f11overig	n_f11overig	Big Integer	✓		Numeric			
✓		n_f12bedrpand_zvbo_industrie	n_f12bedrpand_zvbo_industrie	Big Integer	✓		Numeric			
✓		n_f13bedrpand_zvbo_groen	n_f13bedrpand_zvbo_groen	Big Integer	✓		Numeric			
✓		n_functies	n_functies	Big Integer	✓		Numeric			
✓		n_type_pfunct	n_type_pfunct	Big Integer	✓		Numeric			
✓		type_pfunct_string_agg	type_pfunct_string_agg	Text	✓					2147483647
✓		n_f1woon_m2	n_f1woon_m2	Double	✓		Numeric			
✓		n_f2kantoor_m2	n_f2kantoor_m2	Double	✓		Numeric			
✓		n_f3bijeenkomst_m2	n_f3bijeenkomst_m2	Double	✓		Numeric			
✓		n_f4onderwijs_m2	n_f4onderwijs_m2	Double	✓		Numeric			
✓		n_f5winkel_m2	n_f5winkel_m2	Double	✓		Numeric			
✓		n_f6sport_m2	n_f6sport_m2	Double	✓		Numeric			
✓		n_f7logies_m2	n_f7logies_m2	Double	✓		Numeric			
✓		n_f8gezondheidszorg_m2	n_f8gezondheidszorg_m2	Double	✓		Numeric			
✓		n_f9industrie_m2	n_f9industrie_m2	Double	✓		Numeric			
✓		n_f10cel_m2	n_f10cel_m2	Double	✓		Numeric			
✓		n_f11overig_m2	n_f11overig_m2	Double	✓		Numeric			
✓		n_f12bedrpand_zvbo_industrie_m2	n_f12bedrpand_zvbo_industrie_m2	Double	✓		Numeric			
✓		n_f13bedrpand_zvbo_groen_m2	n_f13bedrpand_zvbo_groen_m2	Double	✓		Numeric			
✓		m2_functies	m2_functies	Double	✓		Numeric			
✓		promotie	promotie	Long	✓		Numeric			

(a) First part of the enriched dataset



Visible	Read Only	Field Name	Alias	Data Type	Allow NULL	Highlight	Number Format	Domain	Default	Length
✓		f1woon_pr	f1woon_pr	Double	✓		Numeric			
✓		f2kantoor_pr	f2kantoor_pr	Double	✓		Numeric			
✓		f3bijeenkomst_pr	f3bijeenkomst_pr	Double	✓		Numeric			
✓		f4onderwijs_pr	f4onderwijs_pr	Double	✓		Numeric			
✓		f5winkel_pr	f5winkel_pr	Double	✓		Numeric			
✓		f6sport_pr	f6sport_pr	Double	✓		Numeric			
✓		f7logies_pr	f7logies_pr	Double	✓		Numeric			
✓		f8gezondheidszorg_pr	f8gezondheidszorg_pr	Double	✓		Numeric			
✓		f9industrie_pr	f9industrie_pr	Double	✓		Numeric			
✓		f10cel_pr	f10cel_pr	Double	✓		Numeric			
✓		f11overig_pr	f11overig_pr	Double	✓		Numeric			
✓		f12bedrpand_zvbo_industrie_pr	f12bedrpand_zvbo_industrie_pr	Double	✓		Numeric			
✓		f13bedrpand_zvbo_groen_pr	f13bedrpand_zvbo_groen_pr	Double	✓		Numeric			
✓		totaal_pr	totaal_pr	Double	✓		Numeric			
✓		max_pr	max_pr	Double	✓		Numeric			
✓		aantal_max	aantal_max	Big Integer	✓		Numeric			
✓		max_pr_string_agg	max_pr_string_agg	Text	✓					2147483647
✓		kas_warenhuis	kas_warenhuis	Text	✓					2147483647
✓		identificatie_kas	identificatie_kas	Long	✓		Numeric			
✓		bouwjaar_kas	bouwjaar_kas	Long	✓		Numeric			
✓		oppervlakte_kas	oppervlakte_kas	Long	✓		Numeric			
✓		kas_aanwezig	kas_aanwezig	Long	✓		Numeric			
✓		bouwjaar_pand	bouwjaar_pand	Long	✓		Numeric			
✓		label_ind	label_ind	Text	✓					2147483647
✓		label_kas	label_kas	Text	✓					2147483647
✓		label_pand	label_pand	Text	✓					2147483647
✓		label_ind_kas	label_ind_kas	Text	✓					2147483647
✓		label_pand_kas	label_pand_kas	Text	✓					2147483647
✓		label_ind_pand	label_ind_pand	Text	✓					2147483647
✓		label_ind_pand_kas	label_ind_pand_kas	Text	✓					2147483647
✓		bouwjaar_ind	bouwjaar_ind	Long	✓		Numeric			
✓		bouwjaar_ind_kas	bouwjaar_ind_kas	Long	✓		Numeric			
✓		bouwjaar_ind_pand	bouwjaar_ind_pand	Long	✓		Numeric			
✓		bouwjaar_ind_pand_kas	bouwjaar_ind_pand_kas	Long	✓		Numeric			
✓		n_vbo_x_label	n_vbo_x_label	Double	✓		Numeric			
✓		aandeel_vbo_x_label	aandeel_vbo_x_label	Double	✓		Numeric			
✓		aandeel_m2_x_label	aandeel_m2_x_label	Double	✓		Numeric			
✓		ei_isso_pand	ei_isso_pand	Double	✓		Numeric			
✓		label_gem	label_gem	Text	✓					2147483647
✓		label_ind	label_ind	Text	✓					2147483647
✓		label_keus	label_keus	Text	✓					2147483647
✓		grootteklasse	grootteklasse	Long	✓		Numeric			
✓		dom_koppel	dom_koppel	Text	✓					2147483647
✓		gas_kwhm2	gas_kwhm2	Double	✓		Numeric			
✓		elek_kwhm2	elek_kwhm2	Double	✓		Numeric			
✓		gas_kwh_jaar2018	gas_kwh_jaar2018	Double	✓		Numeric			
✓		elek_kwh_jaar2018	elek_kwh_jaar2018	Double	✓		Numeric			
✓		warmtenet_pbl	warmtenet_pbl	Text	✓					2147483647
✓		warmtenet_pbl1	warmtenet_pbl1	Long	✓		Numeric			
✓		gas_kwhm2_corr	gas_kwhm2_corr	Double	✓		Numeric			
✓		elek_kwhm2_corr	elek_kwhm2_corr	Double	✓		Numeric			
✓		gas_kwh_jaar2018_corr	gas_kwh_jaar2018_corr	Double	✓		Numeric			
✓		elek_kwh_jaar2018_corr	elek_kwh_jaar2018_corr	Double	✓		Numeric			
✓		dom_koppel_kort	dom_koppel_kort	Text	✓					2147483647
✓		rond	rond	Long	✓		Numeric			

(b) Second part of the enriched dataset



Visible	Read Only	Field Name	Alias	Data Type	Allow NULL	Highlight	Number Format	Domain	Default	Length
✓		grondbox_gelijk	grondbox_gelijk	Long	✓		Numeric			
✓		gevelhoogte_ind_m	gevelhoogte_ind_m	Double	✓		Numeric			
✓		gevelhoogte_m	gevelhoogte_m	Double	✓		Numeric			
✓		gevelhoogte_keuze_m	gevelhoogte_keuze_m	Double	✓		Numeric			
✓		vloeropp_m2	vloeropp_m2	Double	✓		Numeric			
✓		gevelopp_m2	gevelopp_m2	Double	✓		Numeric			
✓		dakopp_m2	dakopp_m2	Double	✓		Numeric			
✓		grond_opp_verhouding	grond_opp_verhouding	Double	✓		Numeric			
✓		dak_vloer_verhouding	dak_vloer_verhouding	Double	✓		Numeric			
✓		vloeropp_keuze_m2	vloeropp_keuze_m2	Double	✓		Numeric			
✓		dakopp_keuze_m2	dakopp_keuze_m2	Double	✓		Numeric			
✓		omtrek_pand_m	omtrek_pand_m	Double	✓		Numeric			
✓		gevelopp_keuze_m2	gevelopp_keuze_m2	Double	✓		Numeric			
✓		schilopp_m2	schilopp_m2	Double	✓		Numeric			
✓		inhoud_m3	inhoud_m3	Double	✓		Numeric			
✓		aantal_dakdelen	aantal_dakdelen	Long	✓		Numeric			
✓		dak_slanted_sum	dak_slanted_sum	Long	✓		Numeric			
✓		dak_horizontal_sum	dak_horizontal_sum	Long	✓		Numeric			
✓		p_dak_horizontaal	p_dak_horizontaal	Double	✓		Numeric			
✓		within	within	Big Integer	✓		Numeric			
✓		touches_point	touches_point	Big Integer	✓		Numeric			
✓		touches	touches	Big Integer	✓		Numeric			
✓		disjoint_false	disjoint_false	Big Integer	✓		Numeric			
✓		overlap_false	overlap_false	Big Integer	✓		Numeric			
✓		buurpand_sum	buurpand_sum	Big Integer	✓		Numeric			
✓		overlap_omtrek_p	overlap_omtrek_p	Double	✓		Numeric			
✓		overlap_gevel_p	overlap_gevel_p	Double	✓		Numeric			
✓		overlap_valide	overlap_valide	Long	✓		Numeric			
✓	✓	geom_Length	geom_Length	Double	✓		Numeric			
✓	✓	geom_Area	geom_Area	Double	✓		Numeric			

(c) Third part of the enriched dataset

Figure C.6: Variable list: BAG dataset enriched with energy-related metrics (Nationaal georegister, 2023)

The initial subset from the BAG dataset intersected with the two neighborhoods (Katendrecht and Kop van Zuid) contained 1,014 entities, including small electricity houses and sheds. To refine the dataset, an SQL query (shown in Figure C.7) was performed, resulting in a set of 658 rows representing the buildings for optimization. Of these, 35 buildings are located in Kop van Zuid, and the remaining 623 are in Katendrecht.

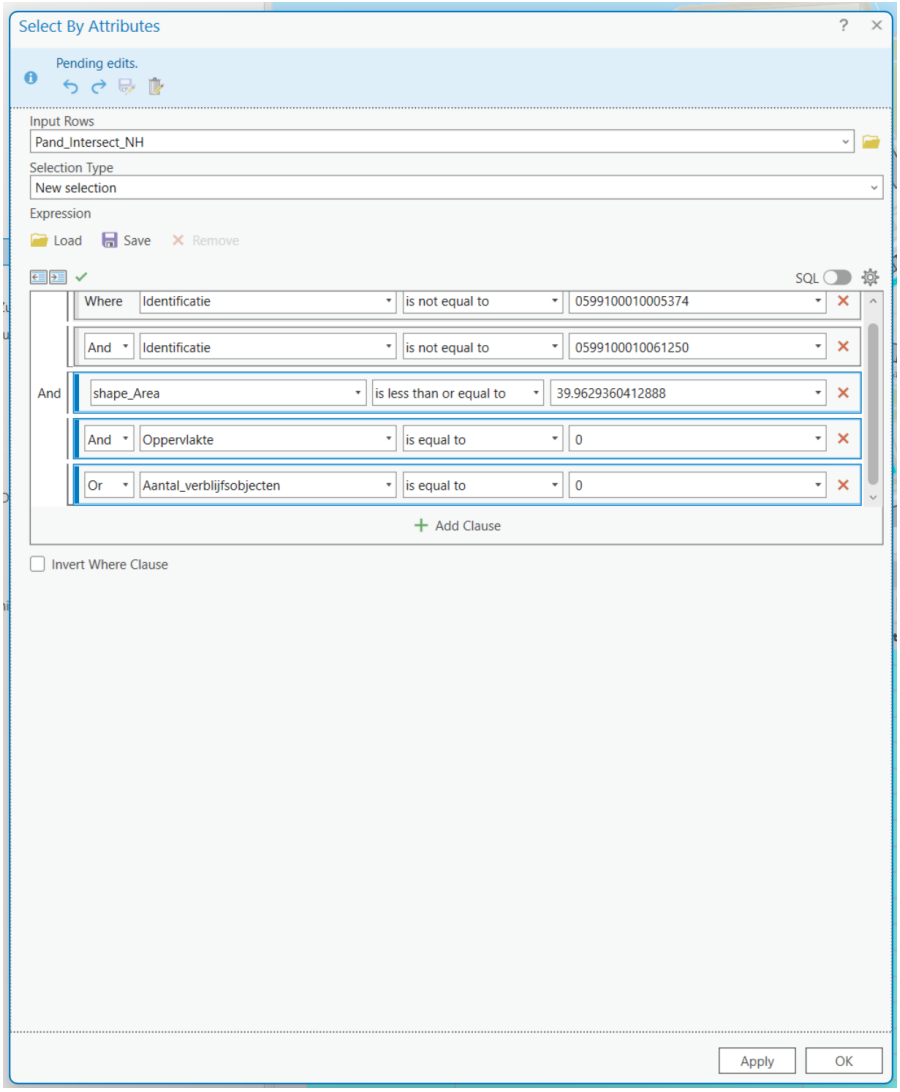


Figure C.7: SQL query in ArcGIS to remove extraneous entities

Further intersection with postal codes, based on the largest overlap, yielded a filtered BAG set enriched with demographic and location information for the analysis.

Next, all building-specific information was joined to the BAG set based on location. Location-based joins, combined with one-to-one joins, ensured that slight errors or deviations in features such as polygon vertices, which can affect other geo-processes like intersection, did not have adverse consequences.

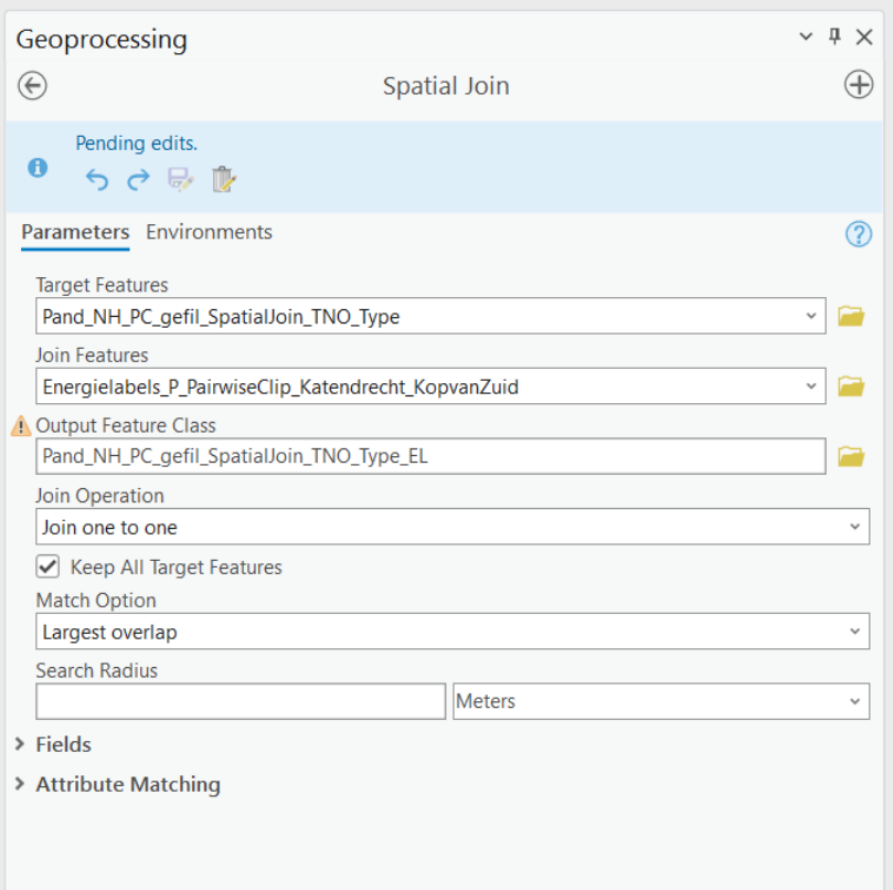


Figure C.8: Spatial join of filtered BAG set with building-specific information

Finally, a comprehensive set was created, containing relevant buildings within the study area and incorporating all available building-specific data. Despite some gaps, as not all supplementary datasets cover the entire input BAG set, this dataset provides a robust overview. It enables the simulation of energy profiles for a subset of buildings, with the potential to extrapolate findings to the entire dataset.

After the framework generation, it appeared that one case study would be sufficient. Kop van Zuid was used. Future work can utilize the framework for a comparative case study.

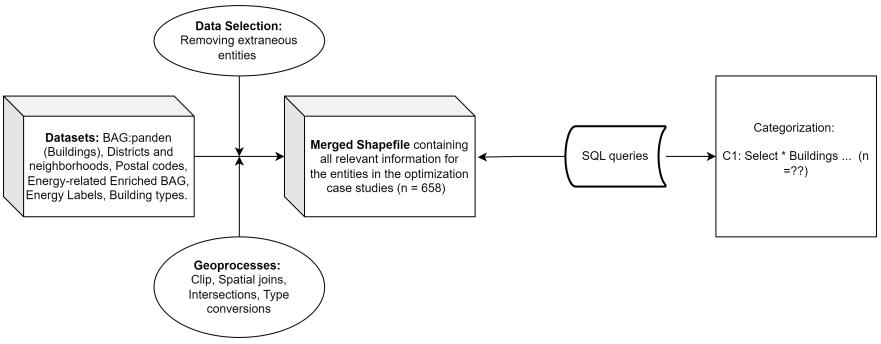


Figure C.9: Data cleaning and data conversion

D

Data acquisition

D.1. KNMI API Data Acquisition

In this section, the process of acquiring weather data from the Royal Netherlands Meteorological Institute (KNMI) API is described. The following Python script illustrates the process of making API calls to download cloud cover data for each month of the year 2022.

```python
import requests
import os

# API key and headers
api_key = '
    ↪ eyJvcmciOiI1ZTU1NGUxOTI3NGE5NjAwMDEyYTNlYjEiLCJpZCI6IjMzOTAyNTJhZTc2NzRlZGZiY2Q5NjA4Y2ZlOTY4YTRhIiwiaC
    ↪ '
headers = {'Authorization': f'Bearer {api_key}'}

# Variables for dataset and version
dataset_name = 'bewolkingsgegevens'
version_id = '1.0'  # Example version ID

# Base URL for file download
base_url = 'https://api.dataplatform.knmi.nl/open-data/v1/datasets/'

# Local directory to save the files
local_directory = r'C:\Users\tessel.kaal\OneDrive - Accenture\Thesis\Thesis P4\Data\Cloud'
os.makedirs(local_directory, exist_ok=True)

for month in range(1, 13):  # Loop through months 1 to 12 for 2022
    formatted_month = f'{month:02d}'
    filename = f'kis_toc_2022{formatted_month}.gz'
    url_file_download = f'{base_url}{dataset_name}/versions/{version_id}/files/{filename}/url
    ↪ '

    # Making the API call to get the download URL
    response_download = requests.get(url_file_download, headers=headers)

    if response_download.status_code == 200:
        download_url_info = response_download.json()
        download_url = download_url_info.get('temporaryDownloadUrl')
        print('Downloading:', filename)

        # Download the file
        if download_url:
            file_response = requests.get(download_url)
            if file_response.status_code == 200:
                file_path = os.path.join(local_directory, filename)
                with open(file_path, 'wb') as file:
                    file.write(file_response.content)
                print(f'Successfully downloaded {filename} to {file_path}')
```

```
            else:
                print(f'Error downloading {filename}')
        else:
            print(f'Download URL not found for {filename}')
    else:
        print(f'Error getting download URL for {filename}:', response_download.status_code,
    ↪ response_download.text)
```

Listing D.1: KNMI API call

The script performs the following tasks:

1. Sets up the API key and headers for authentication.
2. Defines the dataset and version to be accessed.
3. Creates a local directory to store the downloaded files.
4. Iterates over each month of 2022 to download the corresponding cloud cover data files.

D.2. Data Merging and Cleaning

The data from the KNMI API was further processed to integrate with other weather datasets. The following figure shows an example query for merging wind data from KNMI.

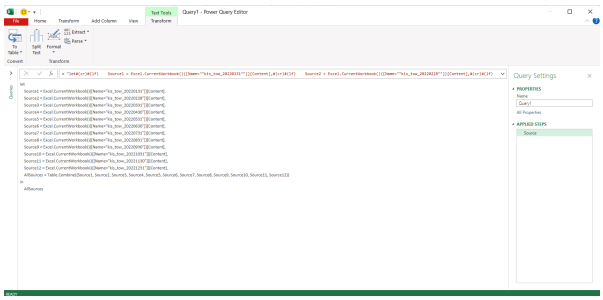


Figure D.1: Query for merging wind data KNMI (Royal Netherlands Meteorological Institute, 2024b)

D.3. Data Resampling and Interpolation

The weather data was initially recorded at 10-minute intervals. For consistency and analysis purposes, the data was interpolated and resampled to 15-minute intervals. For any gaps in the data, the value from one week prior was used to fill in the missing data points.

D.4. Solar Energy Data

The same process was applied to the solar energy data. This ensures uniformity across different datasets, facilitating easier analysis and comparison.

E

Written script

Github repository: https://github.com/TEKaal/Thesis

In this appendix, the most influential scripts are presented with explanations. The entire pipeline consists of additional subscripts, which are all available in the repository.

E.1. Main script

The script outlined below, `RL_main`, Listing E.1, serves as the core for implementing a reinforcement learning framework for microgrid energy management. The main function exists out of:

1. **Imports**: Essential libraries and helper scripts are imported, including those for visualization, environment configuration, and reinforcement learning algorithms. `pymgrid` is used for microgrid simulations.
2. **Main Function**: Defined as `main()`, this function sets up the operational parameters and orchestrates the execution of the simulation.
3. **Setup Parameters**: Specifies the number of steps (`nr_steps`) to simulate a week in 15-minute intervals and sets up paths for load and topology data according to the case study and scenario.
4. **Data Handling**:
    - **Building and DER Configurations**: Loads and processes configurations from CSV files.
    - **Energy Consumption**: Processes energy usage data tailored to the specified buildings and intervals.
5. **Distance Matrix**: Computes a distance matrix to account for spatial relationships within the microgrid, essential for optimization calculations.
6. **Microgrid and Environment Configuration**:
    - **Microgrid Creation**: Constructs a microgrid object with the loaded data.
    - **Environment Initialization**: Sets up a custom environment from the microgrid, defining stochastic trajectories for simulation.
7. **Reinforcement Learning**:
    - **DQN Agent Training**: Trains a DQN agent using specific functions, detailing the number of iterations and steps.
    - **Evaluation**: Assesses the trained agent's performance to validate the effectiveness of the model.
8. **Execution**: Ensures that the main function runs only when the script is executed directly, not when imported.

```python
import argparse
import RL_helpfunctions
from RL_helpfunctionDQN import *
from RL_visualizegrid import *
from RL_microgrid_environment import *
from RL_read_energy_data import *
from RL_connection_matrix import *
from RL_custom_Env import *
import csv
from pymgrid.microgrid.trajectory.stochastic import FixedLengthStochasticTrajectory
from Cluster_algorithm import *
import optuna
import optuna.visualization as vis
import sys
import os
def save_arguments_to_csv(args, outputfolder):
    # Extract necessary arguments from args
    name_giving = input("Please enter the folder name: ")
    dqn_episodes = getattr(args, 'dqn_episodes', 800)  # Using defaults if not specified
    dqn_evaluation_steps = getattr(args, 'dqn_evaluation_steps', 200)

    # Create a directory named with the current date, run number, and other details
    current_date = datetime.now().strftime("%m%d")
    folder_name = f"{current_date}_{dqn_episodes}_{dqn_evaluation_steps}_{name_giving}"
    arguments_folder = os.path.join(outputfolder, folder_name)
    os.makedirs(arguments_folder, exist_ok=True)

    # Create the filename with a timestamp
    current_datetime = datetime.now().strftime("%Y%m%d_%H%M%S")
    csv_filename = f"arguments_{current_datetime}.csv"
    full_path = os.path.join(arguments_folder, csv_filename)

    # Write the arguments to the CSV file
    with open(full_path, 'w', newline='') as csvfile:
        writer = csv.writer(csvfile)
        writer.writerow(['Argument', 'Value'])
        for arg, value in vars(args).items():
            writer.writerow([arg, value])

    return arguments_folder  # Return the new folder name

def main(args):
    # Define the time variables
    run_folder = save_arguments_to_csv(args, args.outputfolder)
    time_interval = args.time_interval

    # Load the case study and scenario files
    df_buildings, coordinates_buildings, horizontal_roof, ids_buildings, type_buildings =
    ↪ load_buildings_from_file(args.case_study_file)
    df_ders, coordinates_ders, ids_ders, type_der = load_DERs_from_file(args.scenario_file,
    ↪ ids_buildings)
    combined_df = concatenate_and_combine_columns(df_buildings, df_ders)

    # Load all the energy data
    Energy_consumption = process_energy_consumption_files(args.folder_path_loads, list(
    ↪ ids_buildings), time_interval)
    microgrid = create_microgrid(Energy_consumption, combined_df, df_buildings)
    print("Microgrid is created, now wrap in gym environment")
    print(microgrid)

    #Wrap microgrid
    microgrid_env = CustomMicrogridEnv.from_microgrid(microgrid)

    # microgrid_env = CustomMicrogridEnv.from_scenario(microgrid_number=10)
    microgrid_env.trajectory_func = FixedLengthStochasticTrajectory(args.nr_steps)

    print("Initialising trajectory")

    # steps used for 3 : 1 training
    microgrid_env.initial_step = 0
```

```
    microgrid_env.final_step = 26280

    trained_agent_DQN = train_dqn_agent(microgrid_env, run_folder, args.dqn_episodes, args.
    ↪ nr_steps, args.dqn_batch_size, args.learning_rate, args.memory_size,  args.num_layers,
    ↪  args.layers_size, args.epsilon_d, args.gamma)

    # print('TRAINING DQL DONE')
    microgrid_env.initial_step = 26280
    microgrid_env.final_step = 35040

    evaluate_dqn_agent(microgrid_env, run_folder, trained_agent_DQN, args.
    ↪ dqn_evaluation_steps, args.nr_steps)

    return microgrid_env

if __name__ == "__main__":
    parser = argparse.ArgumentParser(description='Run microgrid simulation.')

    # File name containing the loads
    folder_path_loads = "Final loads"
    case_study_file = "Buildings and scenarios/CS1.csv"
    scenario_file = "Buildings and scenarios/Scenario1.csv"
    output_file = r"C:\Users\tessel.kaal\OneDrive - Accenture\Thesis\Output training model\
    ↪ VERSION 5\\VERSION 5.1"

    parser.add_argument('--outputfolder', type=str, default=output_file, help='Folder to save
    ↪  output files.')
    parser.add_argument('--folder_path_loads', type=str, default=folder_path_loads, help='
    ↪ Path to the folder containing load files.')
    parser.add_argument('--case_study_file', type=str, default=case_study_file, help='Path to
    ↪  the case study file.')
    parser.add_argument('--scenario_file', type=str, default=scenario_file, help='Path to the
    ↪  scenario file.')
    parser.add_argument('--nr_steps', type=int, default=96, help='Number of steps for the
    ↪ simulation.')
    parser.add_argument('--time_interval', type=int, default=15, help='Time interval in
    ↪ minutes.')
    parser.add_argument('--dqn_episodes', type=int, default=400, help='Number of episodes for
    ↪  DQN training.')
    parser.add_argument('--dqn_batch_size', type=int, default=64, help='Batch size for DQN
    ↪ training.')
    parser.add_argument('--dqn_evaluation_steps', type=int, default=200, help='Number of
    ↪ evaluation steps for DQN.')
    parser.add_argument('--learning_rate', type=float, default=0.0000201605258096069, help='
    ↪ Learning rate for DQN.')
    parser.add_argument('--memory_size', type=int, default=96*4, help='Memory allocation.')
    parser.add_argument('--num_layers', type=int, default=4, help='Neural network')
    parser.add_argument('--layers_size', type=int, default=64, help='Neural layer size')
    parser.add_argument('--epsilon_d', type=float, default=0.929251846973606, help='Memory
    ↪ allocation.')
    parser.add_argument('--gamma', type=float, default=0.9281892353169647, help='Memory
    ↪ allocation.')
    parser.add_argument('--n_trials', type=int, default=100, help='Number of trials for
    ↪ Optuna optimization.')

    args = parser.parse_args()

    # Initialize the environment in the main function
    microgrid_env = main(args)
```

Listing E.1: Main script

E.2. Microgrid environment

The subscript below, `RL_microgrid_environment`, Listing E.2, integrates several modules for energy generation, consumption, and storage. Here is an overview of its functionalities:

1. Imports and Seed Setting: Initially, the script imports necessary libraries for data handling and visualization, such as `matplotlib`, `numpy`, and `pandas`. It sets a random seed to ensure

reproducibility of results.

2. Utility Functions:

   - `calculate_wind_power()`: Calculates power generation from wind speeds using physical parameters of the wind turbine like radius, air density, and power coefficient. The result is adjusted for 15-minute intervals.
   - `adjust_load_profile()`: Adjusts energy consumption profiles to match the simulation's time steps, either by truncating or padding the data.

3. Microgrid Setup:

   - Load Modules: For each house in the energy consumption dataset, a `LoadModule` is created with adjusted load profiles.
   - Battery Modules: Configures small and large battery modules based on counts derived from the combined data frame. Batteries are set up with specified capacities, charge/discharge rates, efficiencies, and initial states of charge.
   - Solar and Wind Modules: Solar and wind energy generation is simulated using imported data functions and calculated using predefined formulas. The solar module considers the roof area available for solar panels from the buildings data frame.

4. Energy Forecasting: Each module (load, solar, wind, and grid) is equipped with an `OracleForecaster` to predict future energy production or consumption. This forecasting component is critical for planning and decision-making in the microgrid management system.
5. Grid Module Setup: The grid module simulates energy import and export, integrating dynamic pricing and CO2 impact, using interpolated cost arrays for energy import and export.
6. Microgrid Assembly: Combines all individual energy modules (grid, solar, wind, load, and battery) into a single `Microgrid` object. This object represents the complete microgrid system, ready for simulation and optimization via reinforcement learning algorithm

```
    import matplotlib.pyplot as plt
from RL_read_energy_data import *
from RL_read_solar_data import *
from RL_read_wind_data import *
from RL_read_grid_costs import *
from RL_evbattery_module import EVBatteryModule

file_path = r"C:\Users\tessel.kaal\OneDrive - Accenture\Thesis_v18may\RL_availability.csv"

# Load the CSV file into a pandas DataFrame
availability = pd.read_csv(file_path)

# Convert the DataFrame column to a list (array)
availability_array = availability.iloc[:, 0].tolist()

steps = 35040

import random
import numpy as np
import pandas as pd

np.random.seed(0)

from pymgrid import Microgrid
from pymgrid.modules import (
    BatteryModule,
    LoadModule,
    RenewableModule,
    GridModule,
    GensetModule)
from pymgrid.forecast import OracleForecaster


```

```
def calculate_wind_power(wind_speeds, radius, power_coefficient, air_density=1.225):
    # Area of the rotor swept by the blades (\pi r^{2})
    area = np.pi * (radius ** 2)

    # Calculate power in watts using the wind power formula: P = 0.5 * \rho * A * v^{3} * C_{
    ↪ p}
    power_watts = 0.5 * air_density * area * (wind_speeds ** 3) * power_coefficient

    # Convert power from watts to kilowatts
    power_kilowatts = power_watts / 1000
    power_kilowattshour = power_kilowatts * 0.25

    return power_kilowattshour

def adjust_load_profile(load_profile, steps, factor=0.8):
    # Convert load profile to a NumPy array if it's not already one
    if not isinstance(load_profile, np.ndarray):
        load_profile = np.array(load_profile)

    # Apply the factor to each element in the load profile
    scaled_profile = load_profile * factor

    length = len(scaled_profile)

    if length > steps:
        # Keep only the last 'timesteps' values (considering warm up period)
        adjusted_profile = scaled_profile[-steps:]
    elif length < steps:
        # Extend the load profile if it's shorter than required
        # This example pads with the last value, but other strategies can be used
        padding = [scaled_profile[-1]] * (steps - length)
        adjusted_profile = np.concatenate((scaled_profile, padding))
    else:
        # If the load profile already matches the timesteps, use it as is
        adjusted_profile = scaled_profile

    return adjusted_profile

def create_microgrid(Energy_consumption, combined_df, df_buildings):
    load_modules = {}
    plot = []
    RES = []

    # Access the first row assuming it contains the grid information
    grid_info = combined_df.iloc[len(df_buildings)]  # Use iloc to access by position
    horizon = int(grid_info['Horizon'])

    for house_id, load_profile in Energy_consumption.items():
        # Create the load module
        processed_profile = adjust_load_profile(load_profile, steps)
        load_module = LoadModule(time_series=adjust_load_profile(load_profile, steps))
        plot.append(processed_profile[(steps // 52 )* 2 :(steps // 52 )* 3])
        # Set the forecaster separately if the LoadModule class requires it
        load_module.set_forecaster(forecaster="oracle",
                                   forecast_horizon=horizon,
                                   forecaster_increase_uncertainty=False,
                                   forecaster_relative_noise=False)

        load_modules[f'load_{house_id}'] = load_module
        print("this is the length of the load module", len(load_module))

    # Filter out the rows where the Type is "Battery"
    battery_df = combined_df[combined_df["Type"] == "Battery"]

    # Group by the characteristics that define 'sameness' and sum relevant columns
    grouped_battery_df = battery_df.groupby(['Min_capaci', 'Max_capaci', 'Max_charge', '
    ↪ Max_discha', 'Efficiency', 'Battery_co']).agg(
        {
            'Min_capaci': 'sum',
            'Max_capaci': 'sum',
            'Max_charge': 'sum',
```

```
            'Max_discha': 'sum',
            'Efficiency': 'first',
            'Battery_co': 'sum'
        }).reset_index(drop=True)

    # Initialize a dictionary to store battery modules
    battery_modules = {}

    # Iterate over the grouped DataFrame
    for index, row in grouped_battery_df.iterrows():
        # Create a BatteryModule instance for each aggregated group
        battery_module = BatteryModule(
            min_capacity=row["Min_capaci"],
            max_capacity=row["Max_capaci"],
            max_charge=row["Max_charge"],
            max_discharge=row["Max_discha"],
            efficiency=row["Efficiency"],
            init_soc=random.uniform(0, 1),  # Initialize SOC randomly between 0 and 1
            battery_cost_cycle=row["Battery_co"]
        )

        # Store the battery module in the dictionary with its index as the key
        battery_modules[index] = battery_module

    solar_data_array = solar_data(steps)

    # Convert the column to numeric, coercing errors to NaN
    df_buildings["TNO_dakopp_m2"] = pd.to_numeric(df_buildings["TNO_dakopp_m2"], errors='
    ↪ coerce')
    roof_partition = df_buildings["TNO_dakopp_m2"].sum() * 0.6
    # Print the calculated sum
    print("Horizontal roof partition calculated:", roof_partition)
    # problem is not all buildings have this

    efficiency_pv = 0.9
    solar_energy = roof_partition * solar_data_array * efficiency_pv

    RES.append((solar_energy[(steps // 52 )* 2 :(steps // 52 )* 3]))

    solar_energy = RenewableModule(time_series=(50*solar_energy))
    solar_energy.set_forecaster(forecaster="oracle",
                                            forecast_horizon=horizon,
                                            forecaster_increase_uncertainty=False,
                                            forecaster_relative_noise=False)

    # Filter out the rows where the Type is "Windturbine"
    windturbine_df = combined_df[combined_df["Type"] == "Windturbine"]
    # Dictionary to store wind modules
    wind_modules = {}

    # Iterate through the filtered DataFrame
    for index, row in windturbine_df.iterrows():
        # If wind data varies for each turbine, call here; otherwise, call outside the loop
    ↪ as shown
        wind_speed_array = wind_data(steps)
        wind_data_array = calculate_wind_power(wind_speed_array, row["Radius_WT"], row["
    ↪ Power_coef"])

        # Create a RenewableModule instance for each turbine
        wind_module = RenewableModule(time_series=(50*wind_data_array))

        # Configure the forecaster settings for the module
        wind_module.set_forecaster(
            forecaster="oracle",
            forecast_horizon=horizon,  # Assuming the horizon is 4 weeks, 24 hours per day
            forecaster_increase_uncertainty=False,
            forecaster_relative_noise=False
        )

        # Store the module in the dictionary with a unique key
        wind_modules[f'windmod_{index}'] = ("windmodule", wind_module)
```

```
171
172     RES.append(((wind_data_array)*len(windturbine_df))[(steps // 52 )* 2 :(steps // 52 )* 3])
173
174     # Ensure that Epot and heat_pump_cop are defined
175     Epot = grid_info['Geo_potent']
176     heat_pump_cop = 4.5
177
178     # Initialize total variables
179     total_base_electrical_load = 0
180     total_energy_output = 0
181     df_buildings['TNO_grond_opp_m2'].fillna(0, inplace=True)
182
183     for index, row in df_buildings.iterrows():
184         # Directly access the column value in the row, which is now guaranteed not to be NaN
185         amount_squaremeters = row['TNO_grond_opp_m2']
186         print(amount_squaremeters)
187         residential_homes = row['BAG_aantal_verblijfsobjecten']
188
189         # Calculate base electrical load for current row
190         base_electrical_load_gjoule = Epot * amount_squaremeters
191
192         # Constant for conversion from GJ to kWh
193         gj_to_kWh = 277.778
194
195         # Calculate the kWh for the given joules
196         base_electrical_load = base_electrical_load_gjoule * gj_to_kWh * 0.25
197
198         # Calculate total energy output for current row
199         energy_output = base_electrical_load * heat_pump_cop
200         print(energy_output)
201
202         # limit energy output with amount of residential homes * 20kW
203         max_energy_output = residential_homes * 20 * heat_pump_cop * 0.25# Convert 20 kW to
    ↪ watts
204         print("residential homes", residential_homes)
205         print("max energy output", max_energy_output)
206
207         if energy_output > max_energy_output:
208             energy_output = max_energy_output
209
210         # Accumulate the totals
211         total_base_electrical_load += base_electrical_load
212         total_energy_output += energy_output
213
214     print("The total energy output is", total_energy_output)
215
216     # Create a single GensetModule instance with the total calculated values
217     total_heat_pump = GensetModule(
218         running_min_production=0,  # Assuming no minimum production specified
219         running_max_production=(total_energy_output*0.8)*1.5, #effiency,
220         genset_cost=0.5,  #standard is 0.4 divided by 4
221         co2_per_unit=0.0,
222         cost_per_unit_co2=0.0,
223         start_up_time=2,
224         wind_down_time=2,
225         provided_energy_name='Heatpump'
226     )
227
228     max_import = grid_info['Max_import']
229     max_export = grid_info['Max_export']
230     co2_price = grid_info['CO2_price']
231
232     # Now you can use these values in your application
233     print("Max Import:", max_import)
234     print("Max Export:", max_export)
235     print("CO2 Price:", co2_price)
236
237     import_array = interpolate_import_costs(steps)
238     RES.append(import_array[(steps // 52 )* 2 :(steps // 52 )* 3])
239
240     # import, export, Co2
```

```
    grid_ts = [0.2,0.1,co2_price] * np.ones((steps, 3))
    grid_ts[:, 0] = import_array
    grid_ts[:, 1] = import_array # just
    grid_ts[:, 0] = np.where(grid_ts[:, 0] < 0, 0, grid_ts[:, 0])
    grid_ts[:, 1] = np.where(grid_ts[:, 1] < 0, 0, grid_ts[:, 1])

    grid = GridModule(max_import=max_import,
                      max_export=max_export,
                      time_series=grid_ts)

    grid.set_forecaster(forecaster="oracle",
                                forecast_horizon=horizon,
                                forecaster_increase_uncertainty=False,
                                forecaster_relative_noise=False)

    modules = [grid,
               ('Heatpump', total_heat_pump),
               ('solar_energy', solar_energy)
               ]

    combined_modules = (modules + list(load_modules.values()) + list(battery_modules.values())
    ↪ )
                        + list(wind_modules.values()))
    microgrid = Microgrid(combined_modules)

    # print all input data
    plt.figure(figsize=(15, 10))

    # Create the first axes
    ax1 = plt.gca()  # Gets the current axis (creates if necessary)

    # Plot all datasets except the last on the first y-axis
    for i, sublist in enumerate(RES[:-1]):
        ax1.plot(sublist, label=f'Dataset {i + 1}')

    # Set labels for the first y-axis
    ax1.set_xlabel('Time')
    ax1.set_ylabel('Energy (kWh)')
    ax1.set_title('Energy Generation and Market Prices')

    # Create a second y-axis
    ax2 = ax1.twinx()  # Creates a second y-axis sharing the same x-axis

    # Plot the last dataset on the second y-axis
    ax2.plot(RES[-1], label=f'Dataset {len(RES)} (Euros)', color='r')  # 'r' is for red, or
    ↪ choose any color

    # Set labels for the second y-axis
    ax2.set_ylabel('Price (Euros)')

    # Set the legend
    # Getting labels and handles for both axes and combining them for one legend
    handles1, labels1 = ax1.get_legend_handles_labels()
    handles2, labels2 = ax2.get_legend_handles_labels()
    ax1.legend(handles1 + handles2, labels1 + labels2)

    # Add grid
    ax1.grid(True)

    # Show the plot
    # plt.show()
    return microgrid
```

Listing E.2: Script Microgridenv

E.3. DQNAgent Class and Initialization

Below is an explanation of the `DQNAgent` script, Listing E.3, which is essential for training and evaluating a Deep Q-Network (DQN) agent in a reinforcement learning environment. The script is divided into

several key sections, each of which is described in detail.

1. **Imports and Class Definition**:
    - Essential libraries for numerical operations, machine learning, and data plotting are imported, including `numpy`, `random`, `keras`, and `matplotlib`.
    - A `DQNAgent` class is defined, equipped with methods for building neural network models, managing experience replay, action selection through epsilon-greedy policy, and learning through replay of past experiences.
2. **Agent Initialization**:
    - The `__init__` method initializes parameters such as state and action sizes, memory for experience replay, learning rate, and discount factors.
    - The exploration rate (`epsilon`), its decay factor (`epsilon_decay`), and the minimum exploration rate (`epsilon_min`) are set up.
    - Two neural network models are initialized: the primary model and the target model. The target model is updated at fixed intervals to stabilize learning.
3. **Neural Network Model**:
    - The `_build_model` method constructs a Sequential neural network model with an input layer matching the state size, multiple hidden layers specified by `num_layers` and `layer_size`, and an output layer that maps to the action space.
    - The model uses the mean squared error loss function and Adam optimizer with a specified learning rate.
4. **Experience Replay**:
    - **Memory Management**: The `remember` method saves experiences (state, action, reward, next_state, done) into a deque object, which acts as a buffer for training the agent.
    - **Replay Method**: The `replay` method samples from this memory to perform batch updates on the model. It calculates target Q-values using the target network and minimizes loss by updating the primary network.
5. **Action Selection and Training**:
    - **Action Selection**: The `act` method returns actions using an epsilon-greedy policy. It selects random actions with probability `epsilon` to encourage exploration, and the action with the highest Q-value otherwise.
    - **Training the Agent**: The `train_dqn_agent` function initializes the environment and agent, then runs through a number of episodes, collecting rewards and updating the agentâĂŹs knowledge. At each step, the agent decides actions based on the current state, updates its memory, and learns from a batch of past experiences. The target model is updated at fixed intervals.
6. **Evaluation**:
    - **Agent Evaluation**: The `evaluate_dqn_agent` function tests the trained agent on the environment without further training, assessing its performance over several episodes. It records the rewards and actions for each episode, saves them to CSV files, and generates plots to visualize the reward dynamics.
7. **Visual Output**:
    - Plots are generated to visually represent the reward dynamics across training and evaluation phases, helping to assess the agent's learning progress and effectiveness.

```python
import random
import numpy as np
from collections import deque
from keras.models import Sequential
from keras.layers import Dense
from keras.optimizers import Adam
from keras.layers import Input
import csv
import matplotlib.pyplot as plt
from datetime import datetime
import os
import tensorflow as tf
import gym

class DQNAgent:
    def __init__(self, state_size, action_size, learning_rate, memory_size, num_layers, layers_size, epsilon_d, gamma):
        self.state_size = state_size
        self.action_size = action_size
        self.memory = deque(maxlen=memory_size)  # Increased! Memory for experience replay
        self.gamma = gamma  # Discount factor
        self.epsilon = 1.0  # Exploration rate
        self.epsilon_decay = epsilon_d
        self.epsilon_min = 0.01
        self.learning_rate = learning_rate
        self.num_layers = num_layers
        self.layer_size = layers_size
        self.model = self._build_model()
        self.target_model = self._build_model()
        self.update_interval = 1000
        self.update_target_model()
        self.total_steps = 0


    def _build_model(self):
        model = Sequential()
        model.add(Input(shape=(self.state_size,)))
        for _ in range(self.num_layers):
            model.add(Dense(self.layer_size, activation='relu'))
        model.add(Dense(self.action_size, activation='linear'))
        model.compile(loss='mse', optimizer=Adam(learning_rate=self.learning_rate))
        return model

    def save(self, filepath, rewards_data):
        with open(filepath, 'w', newline='') as file:
            writer = csv.writer(file)
            writer.writerow(['Episode', 'Total Reward'])
            writer.writerows(rewards_data)

    def update_target_model(self):
        """Updates the target model to be the same as the model."""
        self.target_model.set_weights(self.model.get_weights())

    def remember(self, state, action, reward, next_state, done):
        """Stores experiences in the memory."""
        self.memory.append((state, action, reward, next_state, done))

    def act(self, state):
        """Returns actions using epsilon-greedy policy."""
        if np.random.rand() <= self.epsilon:
            return random.randrange(self.action_size)
        act_values = self.model.predict(state)
        return np.argmax(act_values[0])

    def replay(self, batch_size):
        if (len(self.memory) < batch_size):
            return  # Not enough memory to sample from

        minibatch = random.sample(self.memory, batch_size)
        states = np.array([t[0].reshape(self.state_size) for t in minibatch])
```

```
        actions = np.array([t[1] for t in minibatch])
        rewards = np.array([t[2] for t in minibatch])
        next_states = np.array([t[3].reshape(self.state_size) for t in minibatch])
        dones = np.array([t[4] for t in minibatch])

        # Predict the Q values of next states
        next_Q_values = self.target_model.predict(next_states)
        # Update the Q values for each state based on whether episode is done
        target_Q_values = rewards + self.gamma * np.max(next_Q_values, axis=1) * (1 - dones)

        # Predict the Q values of initial states
        target_f = self.model.predict(states)
        for i, action in enumerate(actions):
            target_f[i][action] = target_Q_values[i]

        # Train the model with states and their new Q-values
        self.model.fit(states, target_f, epochs=1, verbose=0)

        # Decay epsilon
        if self.epsilon > self.epsilon_min:
            self.epsilon *= self.epsilon_decay


def train_dqn_agent(env, outputfolder, episodes, nr_steps, batch_size, learning_rate,
    ↪ memory_size, num_layers,
                    layer_size, epsilon_decay, gamma, trial='1', pre_trained_agent=None):
    print("Start to train the agent")

    state_size = env.observation_space.shape[0]
    action_size = env.action_space.n

    # Use pre-trained agent if provided, otherwise create a new one
    if pre_trained_agent:
        agent = pre_trained_agent
        print("Using pre-trained agent")
    else:
        agent = DQNAgent(state_size, action_size, learning_rate, memory_size, num_layers,
    ↪ layer_size, epsilon_decay,
                         gamma)
        print("Creating new agent")

    rewards_data = []
    total_rewards = 0
    total_steps = 0  # Total number of steps taken for updating target model

    for e in range(episodes):
        state = env.reset()
        state = np.reshape(state, [1, state_size])
        total_reward = 0
        counter = 0

        while counter < nr_steps:
            action = agent.act(state)
            next_state, reward, done, _ = env.step(action)
            next_state = np.reshape(next_state, [1, state_size])

            agent.remember(state, action, reward, next_state, done)
            state = next_state
            total_reward += reward
            total_steps += 1

            if len(agent.memory) > batch_size:
                agent.replay(batch_size)

            # Update target model at fixed intervals
            if total_steps % agent.update_interval == 0:
                agent.update_target_model()

            counter += 1

            if done:
```

```
                break

        print(f'Episode: {e + 1}/{episodes}, Total Reward: {total_reward}, Epsilon: {agent.epsilon:.2f}')
        rewards_data.append([e + 1, total_reward])  # Append episode number and total reward
        total_rewards += total_reward

    average_reward = total_rewards / episodes
    print(f'Average Reward: {average_reward}')

    plt.figure(figsize=(10, 5))
    plt.plot([x[0] for x in rewards_data], [y[1] for y in rewards_data], marker='o', linestyle='-', color='b')
    plt.xlabel('Episode')
    plt.ylabel('Reward')
    plt.title('Reward per Episode Training')
    plt.grid(True)
    current_datetime = datetime.now().strftime("%Y%m%d_%H%M%S")
    plt.savefig(os.path.join(outputfolder, f"plot_training_rewards_season{trial}_{current_datetime}.png"))
    plt.close()

    # Save rewards to CSV for training
    with open(os.path.join(outputfolder, f"training_rewards_season{trial}_{current_datetime}.csv"), 'w', newline='') as file:
        writer = csv.writer(file)
        writer.writerow(['Episode', 'Reward'])
        writer.writerows(rewards_data)

    return agent


def evaluate_dqn_agent(env, outputfolder, agent, episodes, nr_steps, trial='1'):
    episode_rewards_list = []
    total_rewards = 0
    current_datetime = datetime.now().strftime("%Y%m%d_%H%M%S")

    for e in range(episodes):
        state = env.reset()
        state = np.reshape(state, [1, agent.state_size])
        done = False
        episode_rewards = 0
        counter = 0
        actions = []  # Reset actions list for each new episode

        while (counter < nr_steps):
            action = np.argmax(agent.model.predict(state)[0])
            actions.append(action)
            next_state, reward, done, _ = env.step(action)
            next_state = np.reshape(next_state, [1, agent.state_size])
            state = next_state
            episode_rewards += reward
            counter += 1

        episode_rewards_list.append(episode_rewards)
        print(f'Episode: {e + 1}/{episodes}, Reward: {episode_rewards}')
        total_rewards += episode_rewards

        # Save actions to CSV for each episode
        with open(os.path.join(outputfolder, f"actions_season{trial}_{current_datetime}_episode_{e+1}.csv"), 'w', newline='') as file:
            writer = csv.writer(file)
            writer.writerow(['Step', 'Action'])
            writer.writerows([[i + 1, act] for i, act in enumerate(actions)])

    with open((outputfolder + f"\\output_rewards_evaluation_season{trial}_{current_datetime}.csv"), 'w', newline='') as file:
        writer = csv.writer(file)
        writer.writerow(['Episode', 'Reward'])
        writer.writerows([[i + 1, r] for i, r in enumerate(episode_rewards_list)])

```

```
    average_reward = total_rewards / episodes
    print(f'Average Reward Evaluation: {average_reward}')

    plt.figure(figsize=(10, 5))
    plt.plot(range(1, episodes + 1), episode_rewards_list, marker='o', linestyle='-', color='
    ↪ r')
    plt.xlabel('Episode')
    plt.ylabel('Reward')
    plt.title('Reward per Episode Evaluation')
    plt.grid(True)
    plt.savefig(os.path.join(outputfolder, f"plot_evaluation_rewards_season{trial}_{
    ↪ current_datetime}.png"))
    plt.close()

    return average_reward, episode_rewards_list
```

Listing E.3: DQN helper script

E.4. Seasonal data split

F

Results Optimization

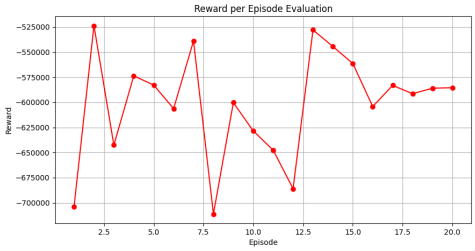
F.1. Pymgrid standard: Microgrid 10 and 21

Table F.1: Comparison of Microgrid 10 and Microgrid 21 configurations.

Parameter	Microgrid 10	Microgrid 21
Final Step	8759	8759
Initial Step	0	0
Load Module		
Forecast Horizon	23	23
Forecaster	oracle	oracle
Renewable Module (PV)		
Provided Energy Name	renewable_used	renewable_used
Unbalanced Energy Module		
Loss Load Cost	10	10
Overgeneration Cost	1	1
Genset		
CO2 Per Unit	2	2
Genset Cost	0.4	0.4
Running Max Production	60144.3	95435.1
Running Min Production	3341.35	5301.95
Battery Module		
Max Capacity	219201	170694
Max Charge	54801	42674
Max Discharge	54801	42674
Min Capacity	43840.2	34138.8
Current Charge	43840.2	34138.8
SOC	0.2	0.2
Final Simulation Parameters		
NR Steps	168	
Time Interval (min)	60	
DQN Episodes	800	
DQN Batch Size	64	
DQN Evaluation Steps	200	
Learning Rate	0.0001	
Memory Size	672	
Num Layers	4	
Layers Size	64	
Epsilon Decay	0.995	

Parameter	Microgrid 10	Microgrid 21
Gamma	1	

(a) Evaluation curve for MG10 with episodes of 24 hours



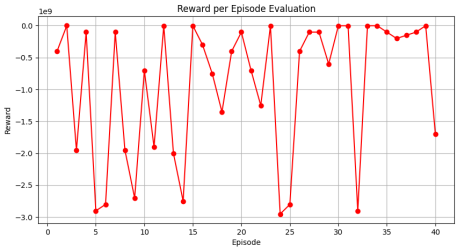
(b) Evaluation curve for MG21 with episodes of 24 hours

F.2. Case study: CustomEnv

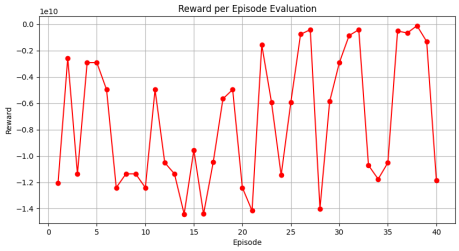
In this section the evaluation curves and the evaluation rewards for the research scenarios are presented.

F.2.1. Scenario 1: Base case

Scenario 1 is the base case scenario that is tested for daily and weekly patterns respectively. The training with day-long episodes yield the highest reward.



(a) Evaluation curve for scenario 1.1



(b) Evaluation curve for scenario 1.2

Figure F.2: Evaluation curves for sub-scenarios within scenario 1

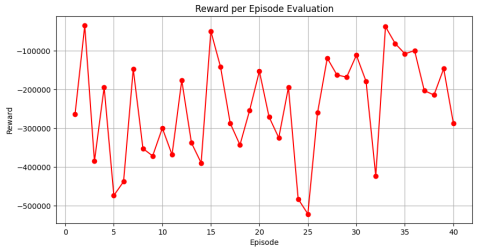
Table F.2: Comparison of rewards for sub-scenarios Scenario 1

Episode	Rewards 1.1	Rewards 1.2
1	-401440312.5	-12060352867
2	-290749.0805	-2557614982
3	-1951769635	-11360413463
4	-101134646.8	-2908055173
5	-2901737619	-2908055104
6	-2801790086	-4958608699
7	-101045346.8	-12410805206
8	-1951708074	-11360408853
9	-2701759190	-11360449605
10	-701521511.4	-12410630492
11	-1901713121	-4958973412
12	-969805.8387	-10510417695
13	-2001428316	-11360538980
14	-2751766084	-14411637488
15	-359051.6801	-9560755822
16	-300965735.2	-14411338733

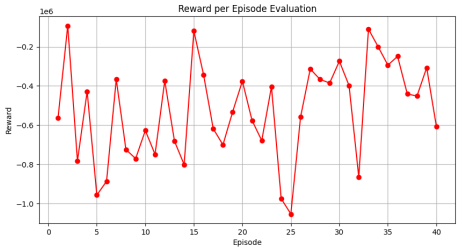
Episode	Rewards 1.1	Rewards 1.2
17	-751487621.1	-10460559093
18	-1351321902	-5659390611
19	-401168110.6	-4958921778
20	-101056708.4	-12410805267
21	-701448132.4	-14161135890
22	-1251670566	-1556939908
23	-1037309.929	-5909593990
24	-2951860735	-11460606298
25	-2801879624	-5909594345
26	-401436519.4	-756239366.9
27	-100919952.6	-405145386.9
28	-100983285.9	-14011080376
29	-600967269.4	-5859542282
30	-781312.6711	-2907999624
31	-1067598.981	-855579351.3
32	-2901777753	-405229635.5
33	-417388.2715	-10710258434
34	-569120.3922	-11760243201
35	-100901456.9	-10510195686
36	-200701576.1	-505387825.9
37	-151159518.1	-655847428.8
38	-101162941.1	-103155895.8
39	-793543.2523	-1306408304
40	-1701428874	-11860354842
Total	-9.31E+08	-7.37E+09

F.2.2. Scenario 2: Demand side decrease

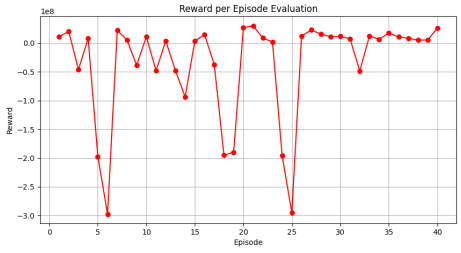
Below the evaluation graphs for all the sub-scenarios of scenario 2 are presented. The exact rewards plus deviation are shown in Table F.3.



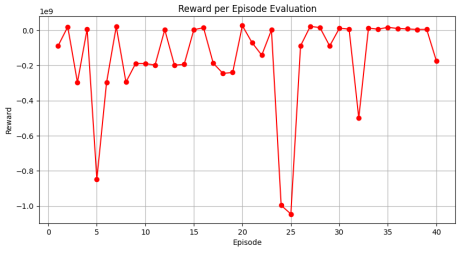
(a) Evaluation curve for scenario 2.1



(b) Evaluation curve for scenario 2.2



(c) Evaluation curve for scenario 2.3



(d) Evaluation curve for scenario 2.4

Figure F.3: Evaluation curves for sub-scenarios within scenario 2

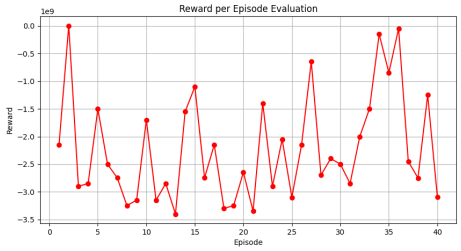
Table F.3: Comparison of rewards for sub-scenarios Scenario 2

Episode	**Scenario 1**	**Scenario 2.1**	**Scenario 2.2**	**Scenario 2.3**	**Scenario 2.4**
1	-401440312.5	-263850.225	-562632.52	11243708.78	-88906300
2	-290749.0805	-34968.999	-92935.0974	20378926.98	20356687
3	-1951769635	-384920.472	-782249.678	-46042368.52	-296224418.3
4	-101134646.8	-193800.091	-428016.955	8170001.99	8081568.372
5	-2901737619	-473546.473	-956693.207	-197268172.2	-847443382.4
6	-2801790086	-437868.006	-886458.897	-297281555.9	-297471342.3
7	-101045346.8	-146809.048	-366125.136	22030294.83	21952417.04
8	-1951708074	-352285.516	-724626.324	5596034.553	-294586171.7
9	-2701759190	-371416.105	-770544.004	-38415474.62	-188583110.3
10	-701521511.4	-299474.677	-627095.943	11228261.92	-188915336.9
11	-1901713121	-368421.834	-749255.82	-47781401.78	-197931585.7
12	-969805.8387	-176259.086	-373671.195	3842559.113	3761055.307
13	-2001428316	-337618.052	-680932.05	-47996450.07	-198115006.9
14	-2751766084	-389471.463	-801630.265	-93509379.5	-193672097.9
15	-359051.6801	-49115.9305	-120294.556	3912893.134	3884350.758
16	-300965735.2	-141752.684	-342871.809	14698386.45	14627943.95
17	-751487621.1	-287361.481	-618399.035	-37215337.55	-187350224
18	-1351321902	-342925.449	-700292.652	-194918454.2	-245024130.1
19	-401168110.6	-253397.649	-531932.648	-189779398.8	-239880575
20	-101056708.4	-152473.466	-375631.543	26634277.5	26554284.28
21	-701448132.4	-270634.917	-576107.551	30322274.97	-69847279.25
22	-1251670566	-324649.061	-678785.414	8799921.766	-141360118.2
23	-1037309.929	-194001.327	-404160.243	2202013.086	2121915.077
24	-2951860735	-482558.635	-974312.962	-196040737.7	-996239320.6
25	-2801879624	-521425.306	-1053096.82	-295331461	-1045515685
26	-401436519.4	-259669.276	-558281.566	12245522.37	-87903699.62
27	-100919952.6	-118818.703	-313561.719	22858774.29	22798864
28	-100983285.9	-162260.152	-366401.414	15705776.2	15636738.34
29	-600967269.4	-167752.306	-385247.579	11195124.45	-88862202.69
30	-781312.6711	-111276.838	-273811.606	11849764.22	11794647.85
31	-1067598.981	-178843.067	-398632.738	7539739.265	7445525.149
32	-2901777753	-422981.679	-863831.567	-48402364.16	-498581637.7
33	-417388.2715	-37398.6933	-110468.475	12474408.14	12450404.17
34	-569120.3922	-80970.1103	-200714.726	6771840.549	6733757.988
35	-100901456.9	-107510.916	-293961.406	17382208.54	17330734.78
36	-200701576.1	-99257.1608	-249133.363	10760545.22	10738528.37
37	-151159518.1	-202654.167	-440806.192	8492855.758	8393038.004
38	-101162941.1	-214026.966	-451708.699	5109410.92	5031447.912
39	-793543.2523	-146044.388	-307072.45	5590508.191	5525677.703
40	-1701428874	-287048.409	-606910.991	25922762.41	-174252802.5
Average	-9.31E+08	-2.46E+05	-5.25E+05	-3.47E+07	-1.59E+08
St. Dev	1067533166	128878.9704	246395.9152	87375974.8	261354482.4
Percentage	-1.15	-0.52	-0.47	-2.52	-1.65

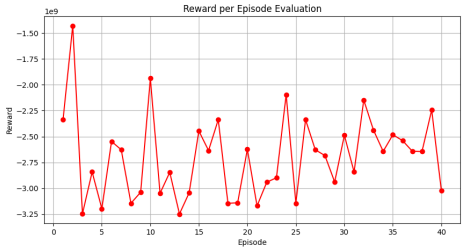
F.2.3. Scenario 3: Increase in Renewable Energy supply

Sub-Scenario 3.1: Solar energy and storage only

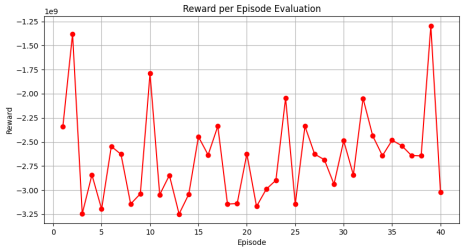
Only solar energy is put into the system. The solar energy in the system is increased until convergence is reached. Base parameters are used and **loads are set to normal again.**



(a) Evaluation curve for scenario 3.1.1: Solar with a factor 1000



(b) Evaluation curve for scenario 3.1.2: Solar with a factor 5000



(c) Evaluation curve for scenario 3.1.3: Solar with a factor 5000 and double storage

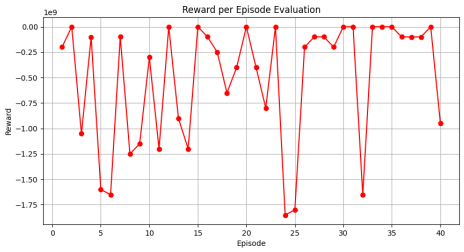
Figure F.4: Evaluation curves for sub-scenarios within scenario 3.1

Table F.4: Comparison of rewards for sub-scenarios Scenario 3.1

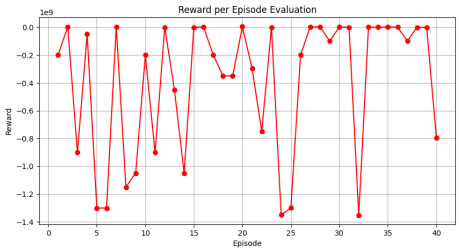
Episode	**Scenario 1.1**	**Scenario 3.1.1**	**Scenario 3.1.2**	**Scenario 3.1.3**
1	-401440312.5	-2148127050	-2338437467	-2338437859
2	-290749.0805	3983204.278	-1429605336	-1379604936
3	-1951769635	-2899677748	-3245623880	-3245624088
4	-101134646.8	-2848684736	-2841645640	-2841645840
5	-2901737619	-1499881406	-3196868952	-3196869152
6	-2801790086	-2499904495	-2546834621	-2546834982
7	-101045346.8	-2746039142	-2627827651	-2627827964
8	-1951708074	-3249211964	-3143959795	-3143960009
9	-2701759190	-3148084315	-3038043644	-3038043860
10	-701521511.4	-1698128432	-1938464833	-1788465124
11	-1901713121	-3149858197	-3047423448	-3047423652
12	-969805.8387	-2849556610	-2845971054	-2845971245
13	-2001428316	-3400026906	-3247831135	-3247831341
14	-2751766084	-1549162881	-3043127653	-3043127864
15	-359051.6801	-1099515622	-2446029162	-2446029351
16	-300965735.2	-2747490557	-2635151404	-2635151724
17	-751487621.1	-2147795032	-2336879010	-2336879419
18	-1351321902	-3299524795	-3144687460	-3144687648
19	-401168110.6	-3248151247	-3139539393	-3139539581
20	-101056708.4	-2644981661	-2623267178	-2623267446
21	-701448132.4	-3344547619	-3169510065	-3169510430
22	-1251670566	-1398588345	-2940863649	-2990863940
23	-1037309.929	-2900019451	-2897608366	-2897608671
24	-2951860735	-2049709334	-2095632876	-2045633278
25	-2801879624	-3099680920	-3144925985	-3144926258
26	-401436519.4	-2148086582	-2337434943	-2337435455
27	-100919952.6	-645684323.4	-2627041222	-2627041504
28	-100983285.9	-2697405262	-2684209107	-2684209428

Episode	**Scenario 1.1**	**Scenario 3.1.1**	**Scenario 3.1.2**	**Scenario 3.1.3**
29	-600967269.4	-2398093904	-2938678240	-2938678437
30	-781312.6711	-2497963296	-2488051719	-2488051918
31	-1067598.981	-2848795512	-2842245312	-2842245502
32	-2901777753	-2000071039	-2147835924	-2047836218
33	-417388.2715	-1497775755	-2437482483	-2437482680
34	-569120.3922	-148905715.5	-2643142330	-2643142532
35	-100901456.9	-846858979.1	-2482532613	-2482532812
36	-200701576.1	-48254918.45	-2539210529	-2539210844
37	-151159518.1	-2448526912	-2641291358	-2641291646
38	-101162941.1	-2749367903	-2644748281	-2644748597
39	-793543.2523	-1249180454	-2244302128	-1294302417
40	-1701428874	-3095532760	-3023887501	-3023887861
Total	-9.31E+08	-2.22E+09	-2.70E+09	-2.66E+09

Sub-Scenario 3.2: Solar Energy Increase



(a) Evaluation curve for scenario 3.2.1: Solar with a factor 100 additional to the base system



(b) Evaluation curve for scenario 3.2.2: Solar with a factor 1000 additional to the base system

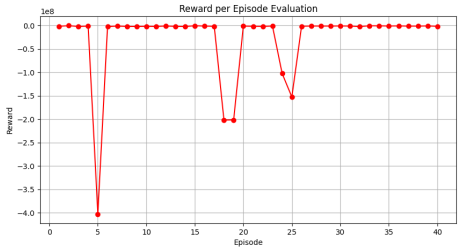
Figure F.5: Evaluation results for sub-scenarios within scenario 3.2

Table F.5: Comparison of rewards for sub-scenarios Scenario 3.2

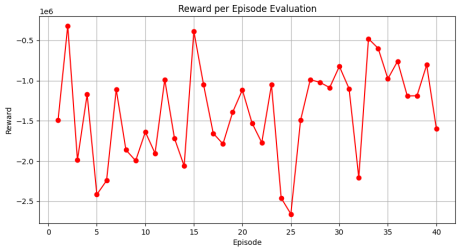
Episode	**Scenario 1**	**Scenario 3.2.1**	**Scenario 3.2.2**
1	-401440312.5	-200592386.3	-198402177.7
2	-290749.0805	290745.2365	3976572.782
3	-1951769635	-1051161635	-900115458.3
4	-101134646.8	-100350925.4	-48781272.56
5	-2901737619	-1601398497	-1300313001
6	-2801790086	-1651378594	-1300409237
7	-101045346.8	-100050172.9	4043965.488
8	-1951708074	-1250955415	-1149675797
9	-2701759190	-1150978132	-1048503222
10	-701521511.4	-300706710.3	-198471712.5
11	-1901713121	-1201179094	-900325462.6
12	-969805.8387	-427360.9375	387298.9733
13	-2001428316	-901231234.1	-450258845.2
14	-2751766084	-1201111989	-1049506349
15	-359051.6801	-92710.62233	652024.1602
16	-300965735.2	-100211590.5	2586814.451
17	-751487621.1	-250572514.3	-198098215.4
18	-1351321902	-650866500.6	-349564607.9
19	-401168110.6	-400464281.4	-348508825.1
20	-101056708.4	68092.65606	4972963.033

Episode	**Scenario 1**	**Scenario 3.2.1**	**Scenario 3.2.2**
21	-701448132.4	-400359317.6	-294716991.6
22	-1251670566	-800765685.9	-748963367
23	-1037309.929	-515794.4422	54004.65052
24	-2951860735	-1851476599	-1350193492
25	-2801879624	-1801512276	-1300114887
26	-401436519.4	-200571434	-198200961.6
27	-100919952.6	-99925148.28	4291488.26
28	-100983285.9	-100162808	2790930.866
29	-600967269.4	-200170108.3	-98038357.97
30	-781312.6711	-66979.21546	2109056.835
31	-1067598.981	-380440.7393	1091592.541
32	-2901777753	-1651262814	-1350546955
33	-417388.2715	116157.6365	2387854.059
34	-569120.3922	-84004.88314	1181514.543
35	-100901456.9	37372.63772	3231480.561
36	-200701576.1	-99999561.38	2015141.078
37	-151159518.1	-100409037.1	-98740343.95
38	-101162941.1	-100488732.8	661877.5628
39	-793543.2523	-265577.842	832002.9158
40	-1701428874	-950389145.7	-795578282.6
Total	-9.31E+08	-5.12E+08	-3.91E+08

Sub-Scenario 3.3: Heat pump Increase



(a) Evaluation curve for scenario 3.3.1: Heat pump with a factor 2 additional to the base system



(b) Evaluation curve for scenario 3.3.2: Heat pump with a factor 10 additional to the base system

Figure F.6: Evaluation results for sub-scenarios within scenario 3.3

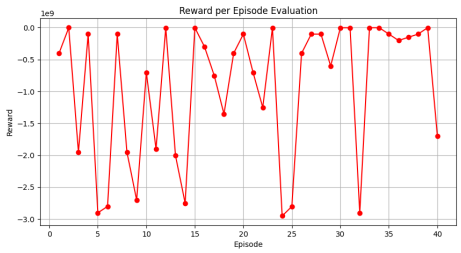
Table F.6: Comparison of rewards for sub-scenarios Scenario 3.3

Episode	**Scenario 1**	**Scenario 3.3.1**	**Scenario 3.3.2**
1	-401440312.5	-1494583.274	-1494583.274
2	-290749.0805	-317656.6706	-317656.6706
3	-1951769635	-1987036.991	-1987036.991
4	-101134646.8	-1171205.949	-1171205.949
5	-2901737619	-402401383.8	-2415996.618
6	-2801790086	-2243646.411	-2243646.411
7	-101045346.8	-1111711.204	-1111711.204
8	-1951708074	-1862477.485	-1862477.485
9	-2701759190	-1996165.931	-1996165.931
10	-701521511.4	-1638481.176	-1638481.176
11	-1901713121	-1904728.54	-1904728.54
12	-969805.8387	-987593.7651	-987593.7651

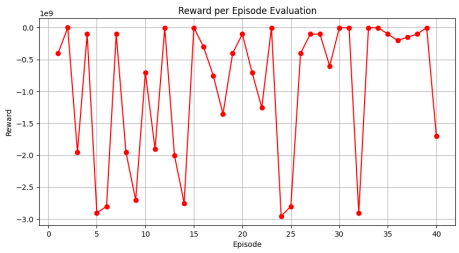
Episode	**Scenario 1**	**Scenario 3.3.1**	**Scenario 3.3.2**
13	-2001428316	-1717114.139	-1717114.139
14	-2751766084	-2061299.753	-2061299.753
15	-359051.6801	-387235.057	-387235.057
16	-300965735.2	-1050961.435	-1050961.435
17	-751487621.1	-1655818.994	-1655818.994
18	-1351321902	-201783628.2	-1787440.909
19	-401168110.6	-201389455.4	-1393268.136
20	-101056708.4	-1115907.722	-1115907.722
21	-701448132.4	-1528301.083	-1528301.083
22	-1251670566	-1771368.425	-1771368.425
23	-1037309.929	-1051327.796	-1051327.796
24	-2951860735	-102456986.9	-2459441.073
25	-2801879624	-152656022.8	-2658404.173
26	-401436519.4	-1493733.249	-1493733.249
27	-100919952.6	-988830.9527	-988830.9527
28	-100983285.9	-1021648.685	-1021648.685
29	-600967269.4	-1087476.365	-1087476.365
30	-781312.6711	-823151.2348	-823151.2348
31	-1067598.981	-1100488.828	-1100488.828
32	-2901777753	-2204966.351	-2204966.351
33	-417388.2715	-479898.2987	-479898.2987
34	-569120.3922	-599032.8434	-599032.8434
35	-100901456.9	-977573.2026	-977573.2026
36	-200701576.1	-759856.6389	-759856.6389
37	-151159518.1	-1191100.431	-1191100.431
38	-101162941.1	-1188408.669	-1188408.669
39	-793543.2523	-805661.9072	-805661.9072
40	-1701428874	-1599882.233	-1599882.233
Total	-9.31E+08	-2.77E+07	-1.40E+06

F.2.4. Scenario 4: Increase in (EV) battery storage possibilities

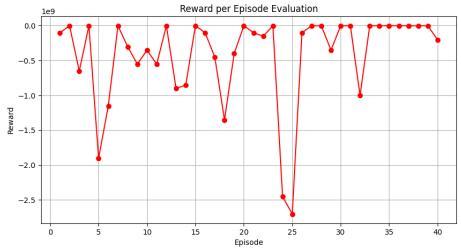
Sub-scenario 4.1: Battery storage and speed increase



(a) Sub-scenario 4.1.1: Double battery storage



(b) Sub-scenario 4.1.2: Double battery speed



(c) Sub-scenario 4.1.3: Halved battery speed

Figure F.7: Evaluation results for sub-scenarios within scenario 4.1

Table F.7: Comparison of rewards for sub-scenarios Scenario 4.1

Episode	Scenario 1	Scenario 4.1.1	Scenario 4.1.2	Scenario 4.1.3
1	-401440312.5	-401440639.2	-401440428.5	-101166535.1
2	-290749.0805	-290749.0805	-290749.0805	-227217.7464
3	-1951769635	-1951769635	-1951769635	-651529183
4	-101134646.8	-101134646.8	-101134646.8	-906434.9657
5	-2901737619	-2901737619	-2901737619	-1901648872
6	-2801790086	-2801790086	-2801790363	-1151636673
7	-101045346.8	-101045346.8	-101045346.8	-830365.8136
8	-1951708074	-1951708074	-1951708187	-301462136.8
9	-2701759190	-2701759880	-2701759799	-551537360.1
10	-701521511.4	-701521511.4	-701521523.6	-351263483.4
11	-1901713121	-1901713818	-1901713732	-551465154.6
12	-969805.8387	-970158.4377	-970318.4377	-772807.1798
13	-2001428316	-2001428627	-2001428596	-901303507.1
14	-2751766084	-2751766084	-2751766288	-851575275.7
15	-359051.6801	-359421.4327	-359793.7974	-282360.2476
16	-300965735.2	-300966071	-300966312.5	-100778840
17	-751487621.1	-751487621.1	-751487626.9	-451247800.6
18	-1351321902	-1351321902	-1351322103	-1351195845
19	-401168110.6	-401168110.6	-401168311.4	-400973168.6
20	-101056708.4	-101056708.4	-101056708.4	-839616.0199
21	-701448132.4	-701448132.4	-701448159.7	-101190002
22	-1251670566	-1251670566	-1251670600	-151383889.8
23	-1037309.929	-1037662.528	-1037822.528	-827564.4643
24	-2951860735	-2951861050	-2951860838	-2451767460
25	-2801879624	-2801879624	-2801879624	-2701768491
26	-401436519.4	-401436519.4	-401436552.1	-101162912

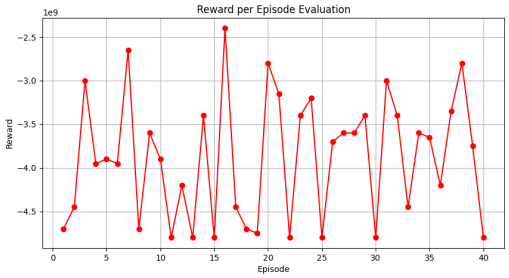


Figure F.8: Sub-scenario 4.2: Double the amount of EV batteries

Episode	Scenario 1	Scenario 4.1.1	Scenario 4.1.2	Scenario 4.1.3
27	-100919952.6	-100919952.6	-100919952.6	-729547.6235
28	-100983285.9	-100983285.9	-100983381	-785299.9936
29	-600967269.4	-600967269.4	-600967269.4	-350816084.9
30	-781312.6711	-781312.6711	-781312.6711	-616682.4242
31	-1067598.981	-1067956.858	-1068116.858	-847990.5289
32	-2901777753	-2901778103	-2901778313	-1001676037
33	-417388.2715	-417760.1076	-417908.0288	-321841.9419
34	-569120.3922	-569120.3922	-569536.77	-449749.2762
35	-100901456.9	-100901785.2	-100901541.4	-709067.3656
36	-200701576.1	-200702281.1	-200702203.8	-566056.1337
37	-151159518.1	-151159875.9	-151159875.9	-925918.0639
38	-101162941.1	-101163269.8	-101163185.9	-932780.5788
39	-793543.2523	-793742.2585	-794009.7691	-632910.1524
40	-1701428874	-1701429021	-1701428959	-201251335
Total	-9.31E+08	-9.31E+08	-9.31E+08	-4.17E+08

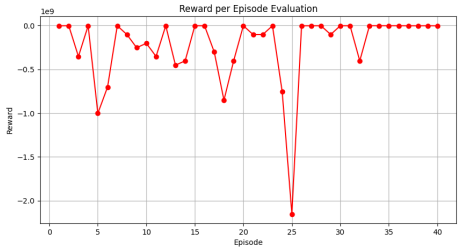
Sub-scenario 4.2: Electric Vehicles as batteries

Table F.8: Comparison of rewards for Sub-scenario 4.2

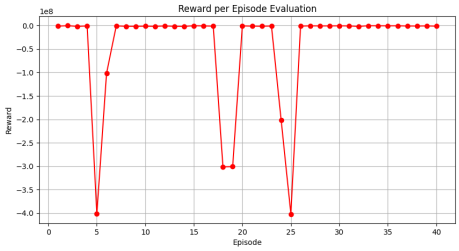
Episode	Scenario 1	Scenario 4.2
1	-401440312.5	-4700146585
2	-290749.0805	-4450054731
3	-1951769635	-3000043011
4	-101134646.8	-3950055382
5	-2901737619	-3900074514
6	-2801790086	-3950051911
7	-101045346.8	-2650046434
8	-1951708074	-4700293974
9	-2701759190	-3600101828
10	-701521511.4	-3900317779
11	-1901713121	-4800126023
12	-969805.8387	-4200321985
13	-2001428316	-4800081329
14	-2751766084	-3400044251
15	-359051.6801	-4800136678
16	-300965735.2	-2400088577
17	-751487621.1	-4450377212
18	-1351321902	-4700112698
19	-401168110.6	-4750074924

Episode	Scenario 1	Scenario 4.2
20	-101056708.4	-2800068907
21	-701448132.4	-3150036269
22	-1251670566	-4800092111
23	-1037309.929	-3400049411
24	-2951860735	-3200053106
25	-2801879624	-4800450481
26	-401436519.4	-3700090543
27	-100919952.6	-3600102904
28	-100983285.9	-3600042695
29	-600967269.4	-3400038801
30	-781312.6711	-4800083398
31	-1067598.981	-3000042530
32	-2901777753	-3400097567
33	-417388.2715	-4450081067
34	-569120.3922	-3600236894
35	-100901456.9	-3650025957
36	-200701576.1	-4200046246
37	-151159518.1	-3350047064
38	-101162941.1	-2800058771
39	-793543.2523	-3750195867
40	-1701428874	-4800054279
Total	-9.31E+08	-3.88E+09

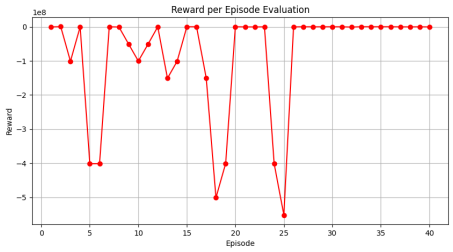
F.2.5. Scenario 5: Hybrid configurations



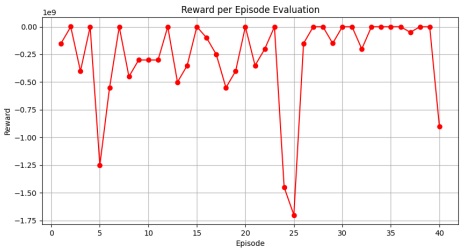
(a) Evaluation for Scenario 5.1



(b) Evaluation for Scenario 5.2



(c) Evaluation for Scenario 5.3



(d) Evaluation for Scenario 5.4

(e) Evaluation for Scenario 5.5

Figure F.9: Evaluations for different scenarios in Scenario 5

Table F.9: Comparison of rewards for sub-scenarios Scenario 5

Episode	**Scenario 5.1**	**Scenario 5.2**	**Scenario 5.3**	**Scenario 5.4**	**Scenario 5.5**
1	-1078495.003	-928628.5067	-566398.8355	-150477870.2	-567237.9379
2	-111751.3387	-88872.9702	109755.754	309460.0558	108558.3276
3	-351673009.5	-1492272.312	-101119598.4	-400957522.4	-1124777.637
4	-751934.6783	-637355.0833	-338358.8584	-269435.1205	-339927.5407
5	-1001912549	-401827378.1	-401484592.2	-1251223813	-401536094.4
6	-701833833.9	-101700655.1	-401320853.9	-551068456.7	-1341692.264
7	-566576.513	-486630.0637	-188118.8065	15239.94315	-188955.112
8	-101515689	-1332863.448	-905315.8937	-450818514.6	-906314.1948
9	-251564625.3	-1380596.322	-50917041.16	-300753813.5	-920370.1854
10	-201247766	-1095208.584	-100661210	-300499742.8	-664488.6147
11	-351588184.6	-1416086.253	-50991089.38	-300782230.7	-992462.9343
12	-717814.8727	-618895.3713	-261764.1573	-144231.562	-262827.9528
13	-451464430.6	-1323482.586	-151121615.6	-500981086.5	-1125909.348
14	-401653131	-1471064.67	-101010142.4	-350895345.4	-1016188.557
15	-183074.4062	-156368.1222	-20647.3009	61635.93921	-22370.22285
16	-578852.1214	-504359.1955	-218257.0263	-100002683.9	-219160.8654
17	-301139131.9	-994428.8853	-150604189.3	-250466818.3	-611013.1191
18	-851344687.3	-301299859	-500954995.7	-550709576.2	-251007206.9

Episode	**Scenario 5.1**	**Scenario 5.2**	**Scenario 5.3**	**Scenario 5.4**	**Scenario 5.5**
19	-400950799.6	-300904665.8	-400544382.6	-400372144.5	-250594184.8
20	-544021.6023	-461032.3277	-137454.1758	128227.3333	-138013.0058
21	-101109410.8	-956946.4752	-506037.4739	-350224286.6	-506741.0795
22	-101364311.6	-1189786.181	-708266.8054	-200578612.1	-709264.8048
23	-812876.0973	-707583.1928	-317812.2987	-189031.7836	-319308.7977
24	-752032657.8	-201885889.1	-401558822.6	-1451323510	-201598029.4
25	-2152126646	-402034309.4	-551675635.2	-1701379515	-301726935
26	-1048409.048	-898531.4061	-555966.3577	-150458011.1	-556805.4602
27	-432051.2391	-366440.6072	-91524.49234	120474.4607	-92520.19534
28	-626926.5034	-535333.6601	-246222.6483	-76064.6596	-247329.8989
29	-100617496.3	-527606.9543	-255295.7557	-150128374.4	-256179.6735
30	-397733.4125	-330690.3061	-111084.5486	11534.14712	-112197.2046
31	-682967.3093	-581528.2116	-293746.8967	-194990.1318	-294758.6891
32	-401810788.1	-1612418.925	-1098213.616	-200942929.8	-1099043.32
33	-132324.7008	-111222.7865	20876.88942	141445.5915	20471.58978
34	-273081.1821	-230539.8026	-94744.50719	-24111.65993	-95175.43052
35	-390539.3673	-330926.358	-69518.84322	113889.7442	-70515.74927
36	-332340.2572	-270390.4022	-42170.83112	101614.5398	-43261.47001
37	-809489.0859	-696237.186	-391521.6532	-50293090.72	-392537.8568
38	-889708.0367	-770813.8176	-455746.0079	-383794.5503	-457893.8688
39	-608510.9607	-527593.3131	-279191.4712	-199742.03	-279800.5361
40	-1191068.577	-1030657.942	-559543.2194	-900327764	-560689.1198
Total	-2.75E+08	-2.25E+08	-8.43E+07	-4.34E+07	-3.56E+07

Longer training and evaluation

Table F.10: Comparison of rewards for Scenario 5.5 and a Longer Training and Evaluation

Episode	**Scenario 5.5.1**	Episode	**Scenario 5.5.2**
1	-567237.9379	1	-291967.8563
2	108558.3276	2	-292589.785
3	-1124777.637	3	-17219.77371
4	-339927.5407	4	66413.10608
5	-401536094.4	5	-370306.3678
6	-1341692.264	6	-100604959.9
7	-188955.112	7	-869396.3575
8	-906314.1948	8	-536399.332
9	-920370.1854	9	-67895.14364
10	-664488.6147	10	-696450.2214
11	-992462.9343	11	-1014476.503
12	-262827.9528	12	-534855.3968
13	-1125909.348	13	-991690.4549
14	-1016188.557	14	-995314.3516
15	-22370.22285	15	86052.24309
16	-219160.8654	16	-1423791.515
17	-611013.1191	17	-1105206.264
18	-251007206.9	18	-629167.2629
19	-250594184.8	19	7559.488676
20	-138013.0058	20	-311937.031
21	-506741.0795	21	-319057.0514
22	-709264.8048	22	-281537.2464
23	-319308.7977	23	-136573.2419
24	-201598029.4	24	-67562.8921

Episode	**Scenario 5.5.1**	Episode	**Scenario 5.5.2**
25	-301726935	25	-172954.2319
26	-556805.4602	26	-130913.4892
27	-92520.19534	27	-1259628.007
28	-247329.8989	28	-331487.8037
29	-256179.6735	29	-377997.5517
30	-112197.2046	30	-100604993.1
31	-294758.6891	31	-1099457.679
32	-1099043.32	32	-25597.03818
33	20471.58978	33	-1474674.823
34	-95175.43052	34	-206817.2142
35	-70515.74927	35	-147854.5055
36	-43261.47001	36	-25032.11746
37	-392537.8568	37	-570350.3646
38	-457893.8688	38	-892988.657
39	-279800.5361	..	..
40	-560689.1198	200	-131945.7693
Total	-3.56E+07	Total	-3.01E+07

Seasonal variations in training

Table F.11: Comparison of rewards for Scenario 5.5 and a Seasonal Training and Evaluation

Episode	**5.5.1 Reward**	**Episode**	**Reward**
1	-567237.9379	1.00	47057.77649
2	108558.3276	2.00	-50601875.65
3	-1124777.637	3.00	102266.2758
4	-339927.5407	4	-537987.4309
5	-401536094.4	5.00	-24484.77286
6	-1341692.264	6	342412.2377
7	-188955.112	7.00	24222.39662
8	-906314.1948	8.00	403520.6712
9	-920370.1854	9.00	-207681.3869
10	-664488.6147	10	2327.481963
11	-992462.9343	1	899639.0371
12	-262827.9528	2	702453.0734
13	-1125909.348	3	897431.6407
14	-1016188.557	4	304138.6972
15	-22370.22285	5	644763.5401
16	-219160.8654	6	980869.7747
17	-611013.1191	7	550744.8261
18	-251007206.9	8	775644.6729
19	-250594184.8	9	822110.851
20	-138013.0058	10	834296.5077
21	-506741.0795	1	38401.20361
22	-709264.8048	2	-3221.094048
23	-319308.7977	3	245262.3568
24	-201598029.4	4	231031.4448
25	-301726935	5	275741.7715
26	-556805.4602	6	-64793.19887
27	-92520.19534	7	-12700.79107
28	-247329.8989	8	208201.597
29	-256179.6735	9	411444.3498
30	-112197.2046	10	206786.3238

Episode	5.5.1 Reward	Episode	Reward
31	-294758.6891	1	-681862.5217
32	-1099043.32	2	-1327228.969
33	20471.58978	3	-296353.0804
34	-95175.43052	4	-1116734.712
35	-70515.74927	5	-842364.3588
36	-43261.47001	6	-915217.7786
37	-392537.8568	7	-401646490.5
38	-457893.8688	8	-1289910.899
39	-279800.5361	9	-1184389.257
40	-560689.1198	10	-322036.3121
Total	**-3.56E+07**		**-1.13E+07**

Hyperparameter optimization

Table F.12: Comparison of rewards for Scenario 5.5 and an Optuna optimization

Episode	**Scenario 5.5.1**	**Scenario 5.5.Optuna**
1	-567237.9379	-254137.8673
2	108558.3276	-1131906.638
3	-1124777.637	-607663.9305
4	-339927.5407	-272011.2323
5	-401536094.4	-264744.755
6	-1341692.264	-905927.3377
7	-188955.112	-968894.1816
8	-906314.1948	-108428.2982
9	-920370.1854	-1125063.985
10	-664488.6147	-694632.8068
11	-992462.9343	-587946.921
12	-262827.9528	-45397.5381
13	-1125909.348	71880.98331
14	-1016188.557	-1467184.141
15	-22370.22285	-607644.7826
16	-219160.8654	-517108.8133
17	-611013.1191	-283236.7551
18	-251007206.9	-558229.5919
19	-250594184.8	-1198103.848
20	-138013.0058	-31581.79296
21	-506741.0795	-362292.7886
22	-709264.8048	-110447.6915
23	-319308.7977	-312444.963
24	-201598029.4	71164.22374
25	-301726935	-113750.8208
26	-556805.4602	-498902.6695
27	-92520.19534	-210194.052
28	-247329.8989	-120546.8667
29	-256179.6735	-262508.3857
30	-112197.2046	-1106813.435
31	-294758.6891	-98784.54515
32	-1099043.32	-722088.6514
33	20471.58978	-1100438.252
34	-95175.43052	-1301842.077
35	-70515.74927	-618727.8332
36	-43261.47001	-293177.8893

Episode	**Scenario 5.5.1**	**Scenario 5.5.Optuna**
37	-392537.8568	-86728.84825
38	-457893.8688	10792.18422
39	-279800.5361	-211301.2794
40	-560689.1198	-55156.87253
Total	-3.56E+07	-4.77E+05

F.3. Smoothness

.

Table F.13: Summary statistics for reward training and evaluation in different Scenarios

Scenario	Mean Reward	Variance of Reward	Standard Deviation of Reward
scenario1.1	-748697815.402244	1.052440048560872e+18	1025885007.4744595
Scenario1.2	-4532227544.86295	2.7855488537145594e+19	5277829907.939966
Scenario2.1	-73362648.52440174	4.519595324935479e+16	212593398.88471323
Scenario2.2	-94623850.503038	5.5649439107242104e+16	235901333.41556615
Scenario2.3	**-31197748.552268416**	2.8719702203172268e+16	169468882.69877827
Scenario2.4	-112691202.06678012	7.752077663618224e+16	278425531.58103555
Scenario3.1.1	-1167125251.2581403	1.2602067468858775e+18	1122589304.6372201
Scenario3.1.2	-2070371666.6779401	5.677878485346621e+17	753516986.2283545
Scenario3.1.3	-2053586091.6102486	5.2452755251454835e+17	724242744.1918547
Scenario3.2.1	-358594202.92848307	2.812106520176486e+17	530292986.95876473
Scenario3.2.2	-277911749.6061633	1.788888283429618e+17	422952513.1063318
Scenario3.3.1	-219927958.97262383	1.964102862489977e+17	443182001.2692276
Scenario3.3.2	**-195286482.0662653**	2.247983408362737e+17	474129033.95201784
Scenario4.1.1	-757266163.9321418	1.0608097436471012e+18	1029956185.3045503
Scenario4.1.2	-754412785.5835935	1.0468486775730555e+18	1023156233.2181022
Scenario4.1.3	-444638023.4074139	5.5216750410762925e+17	743079742.7649534
Scenario5.1	**-219838080.38950327**	1.608182264393648e+17	401021478.7755948
Scenario5.2	**-101552195.55696352**	6.178312911500302e+16	248562123.25091493
Scenario5.3	**-85710107.4798487**	3.424910092532794e+16	185065126.17272857
Scenario5.4	-202782510.06089294	1.3277985565757835e+17	364389703.0070668
Scenario5.5.2	**-38819745.92687987**	1.5521467381513952e+16	124585181.22760007
Scenario5.5.Optuna	**-55353541.77566619**	3.310309039416286e+16	181942546.95964563
Scenario5.5	-53571501.41521569	2.482516622132702e+16	157560040.05244166

Still to be changed for only the evaluation data.

Most Promising Scenarios

The top five most promising scenarios based on a combination of mean reward, variance, and standard deviation are highlighted in the table above. These scenarios are:

1. Scenario 5.3: With the highest mean reward of -85,710,107.4798487, a low variance of 3.424910092532794e+16, and a standard deviation of 185,065,126.17272857, this scenario shows excellent performance with minimal variability.

2. Scenario 5.5.2: This scenario has a mean reward of -38,819,745.92687987, the lowest variance of 1.5521467381513952e+16, and a standard deviation of 124,585,181.22760007, making it highly stable and reliable.

3. Scenario 5.2: With a mean reward of -101,552,195.55696352, variance of 6.178312911500302e+16, and a standard deviation of 248,562,123.25091493, this scenario balances good performance with moderate variability.

4. Scenario 5.1: Featuring a mean reward of -219,838,080.38950327, variance of 1.608182264393648e+17, and a standard deviation of 401,021,478.7755948, this scenario offers a reliable performance with reasonable variability.

5. Scenario 5.5.Optuna: With a mean reward of -55,353,541.77566619, variance of 3.310309039416286e+16, and a standard deviation of 181,942,546.95964563, this scenario demonstrates strong potential with good performance and low variability.

These scenarios were selected based on their optimal combination of high mean rewards, low variances, and low standard deviations, indicating they are both effective and stable.

G

Reproducibility self-assessment

G.1. Marks for each of the criteria

Table G.1: Reproducibility ratings

Aspect	Rating	Description
Input Data	2	One data set accessed with ArcGIS.
Preprocessing	2	One data set is accessed with ArcGIS and must be preprocessed there as well.
Methods	3	All methods are reproducible.
Computational Environment	3	Open source and repository public available.
Results	3	Results can be reproduced with set up computational environment and same settings.

G.2. Reproducibility

The reproducibility of this thesis is of high importance and has been designed to be as high as possible. Here, the aspects that enhance reproducibility are detailed and limitations for the reproducibility are addressed.

G.2.1. Data and Simulation Setup

The overall setup and methodology for this study can be replicated given the same tools and environment:

1. Simulation Environment: The simulation environment, built using EnergyPlus and other advanced modeling tools, can be replicated with the same configurations and input parameters. Detailed instructions and configurations are documented in the repository.
2. Code Availability: The complete code base, including scripts for data processing, model training, and scenario analysis, is available on GitHub. The repository includes a README file, and a requirements file to ensure that others can set up the environment and run the simulations with minimal effort.

G.2.2. Template and Scenario Creation

To facilitate reproducibility:

1. Scenario Templates: A basic template for scenario creation has been developed and is included in the GitHub repository. This template guides users through the process of setting up new scenarios or replicating existing ones, ensuring consistency and ease of use.
2. Documentation: Detailed documentation is provided within the repository, explaining the process of running simulations, generating results, and interpreting outputs. This documentation includes example scripts, configuration files, and illustrative results.

G.2.3. Input Data

A potential challenge to full reproducibility is the access to certain input data:

1. Load Simulation Data: The input data used for load simulations includes proprietary data sourced from ArcGIS, some of which is behind a paywall. This restricts immediate access for all researchers. However, ArcGIS has plans to open source more of their data, which could alleviate this issue in the future. In the meantime, similar datasets might be obtained from alternative sources or through institutional access.
2. Map Layers and GIS Data: The map layers for the digital twin are hosted on ArcGIS Online (AGOL) and were set up using ArcGIS software. While this software was chosen for its capabilities and ease of integration for company purposes, equivalent setups can be created using QGIS, which is an open-source alternative. This provides flexibility for researchers who do not have access to ArcGIS.